CURSO PYTHON DESDE 0



Pere Manel Verdugo Zamora Sobre los vídeos de PILDORASINFORMATICAS

TEMARIO (VIDEO 1)

- Introducción
- Requisitos previos.
- Instalación software
- Fundamentos y sintaxis básica del lenguaje
- POO con Python
- Algoritmos, listas y tramos.
- BBDD
- Trabajo con gráficos y contenidos.
- Procesos y tareas.
- Programación de red.
- Sockets
- Ejercicios prácticos.





INTRODUCCIÓN (VIDEO 2)

- Historia de Python y características de Python.
- Requisitos hardware e instalación de Python.
- IDE a utilizar.

Características.



- Creado por Guido Van Rossum a comienzos de los 90.
- Lenguaje de alto nivel. Gramática sencilla, clara y muy legible.
- Tipado dinámico y fuente.
- Orientado a objetos.
 - Sobrecarga de constructores. Herencia múltiple. Encapsulación. Interfaces. Polimorfismo.
- Open Source.
- Fácil de aprender.
- Librerías estándar muy amplia.
- Interpretado.
- Versátil.
 - Aplicaciones de escritorio, aplicaciones de servidor, aplicaciones web.

Instalación Python

Tendremos que ir a la siguiente página web: <u>http://python.org/downloads/</u>

C Download Python Pyth × + ✓ D O P Python Software Fo	undation ILISI https://www.mith	mm/manloads/					 - 0	
	Python	PSF	Docs	РуРІ	Jobs	Community		
	🄁 pytł	าดก		Donate	Search	60 Socialize		
	Down Lookin Limux// Want t Docker Lookin	nload Python 3.8.5 g for Python with a different O JINX, <u>Mac OS X, Other</u> o help test development versic <u>images</u> g for Python 2.7? See below fo	S? Python for <u>Windows</u> , ins of Python? <u>Prereleases</u> , r specific releases					
	Active Python	Releases	Guida					
	Python version	Maintenance status	First released	End of support	Release scheo	fule		
	3.8	bugfix	2019-10-14	2024-10	PEP 509			
	3.7	security	2018-06-27	2023-06-27	PEP 537			
	3.6	security	2016-12-23	2021-12-23	PEP 494			
	3.5	security	2015-09-13	2020-09-13	PEP 478			
	2.7	end-of-life	2010-07-03	2020-01-01	PEP 373			
	Looking for a s	necific release?						



Selecciona el botón y este descargará la versión para tu sistema operativo.

En mi caso como estoy trabajando con Windows, en la carpeta Descargas encontraremos el siguiente archivo.



Activaremos la opción Add Python 3.8 to PATH, para que desde la línea de comando tengamos acceso a dicha aplicación.

A continuación seleccionaremos Install Now.



Empieza la instalación.

b Python 3.8.5 (32-bit) Setup	- 🗆 X
	Setup was successful
_	Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.
	New to Python? Start with the <u>online tutorial</u> and <u>documentation</u> . See what's new in this release.
	 Disable path length limit Changes your machine configuration to allow programs, including Python, to bypass the 260 character "MAX_PATH" limitation.
python	<u> </u>
windows	<u>C</u> lose

Seleccionaremos path length limit, seguido del botón Closes.

Observaremos en nuestro ordenador que se han instalado las siguientes aplicaciones:



Una de las aplicaciones es IDLE (Python 3.8 32-bit), si lo ejecutamos observaremos una especie de consola:



Para programa Python podemos utilizar varios editores como Eclipse, Notepadd++ o Sublimetext 3 entre otros, en este curso vamos a utilizar Sublimetext 3.





SINTAXIS BÁSICA I (VIDEO3)

En este capítulo veremos:

- IDLE Python
- Primeras instrucciones en Python
- Instalación Sublime Text 3

Vamos a ejecutar IDLE de Python

```
      Python 3.8.5 Shell
      -
      X

      Eile Edit Shell Debug Options Window Help

      Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (In tel)] on win32

      Type "help", "copyright", "credits" or "license()" for more information.

      >>>

      Image: Shell Debug Options Window Help

      Eile Edit Shell Debug Options Window Help

      Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (In tel)] on win32

      Type "help", "copyright", "credits" or "license()" for more information.

      >>>>

      Image: Copyright Debug Options Debug Options Options Debug O
```

Observamos el siguiente símbolo >>> este es el prompt de la consola donde introduciremos nuestras instrucciones Python.

Vamos a escribir la siguiente instrucción:

>>> print ("Hola Alumnos")

Al ejecutar la tecla Enter se ejecutará el resultado.



Si queremos incluir varias instrucciones en una misma línea esta irá separadas por ;.

Podremos agregar comentarios que nos pueden ayudar en la realización del programa, este empezará por el símbolo de almuadilla (#).

También se utilizar para invalidar linear de código que no queremos borrar.



Observas que al pulsar Enter no da error e ignora la línea.

Utilizaremos la barra \ si queremos que la instrucción tanga más de una línea.

```
🍓 Python 3.8.5 Shell
                                                                               ×
<u>File Edit Shell Debug Options Window Help</u>
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (In \land
tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> mi nombre="Mi nombre es Juan!"
>>> mi nombre
'Mi nombre es Juan!'
>>> mi nombre="Mi nombre es \
Juan"
>>> mi_nombre
'Mi nombre es Juan'
>>>
                                                                               Ln: 10 Col: 4
```

No se utiliza con frecuencia pero era interesante comentarlo.

La identación consiste en programar de una manera que a la hora de programas vemos las líneas que pertenecen a una parte del programa.

```
🌛 Python 3.8.5 Shell
                                                                              \times
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (In ٨
tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> for i in range(5):
     print (i)
0
1
2
3
4
>>>
                                                                              Ln: 12 Col: 4
```

Cuando el código se muestra más hacia la derecha es la dentación, para ejecutar este ejemplo tenemos que dar dos Enter.

Pero para poder programar con más facilidad vamos a usar el editor Sublime Text 3, este lo encontraremos en la siguiente página web: <u>https://www.sublimetext.com/3</u>

Download Sublime Text 3 is the current version of Sublime Text. For bleeding-edge releases, see the dev builds. Version: Build 3211 • OS X (10.7 or later is required) • Windows - also available as a portable version • Windows 64 bit - also available as a portable version • Linux repos - also available as a <u>64 bit or 32 bit tarball</u>

Seleccionaremos Windows -also as a portable versión si trabajas con Windows y si trabajas con otro sistema operativo descargas el tuyo.

Vamos a descargas y veremos el siguiente archivo.



Vamos a copiarlos en una carpeta en el Escritorio para descomprimirlo.

Ar Evo Inicio Compartir	Vista			- 0	i × ^ ॡ
Anclar al Acceso rápido	✓ ✓ </th <th>Nueva carpeta</th> <th>Seleccionar too</th> <th>do ⁻ nada ón</th> <th></th>	Nueva carpeta	Seleccionar too	do ⁻ nada ón	
Portapapeles	Organizar	Nuevo Abrir	Seleccionar		
← → · ↑ 🔒 > Sublime	Text >		5 V	D Buscar en	Sublime
A N	lombre ^	Fecha de modificación	Тіро	Tamaño	
	Data	01/08/2020 16:55	Carpeta de archivos		
Escritorio 🖈	Packages	01/08/2020 16:55	Carpeta de archivos		
🔶 Descargas 🖈	changelog	01/10/2019 9:20	Documento de te	56 KB	
🖆 Documentos 🖈	crash reporter	01/10/2019 10:35	Aplicación	227 KB	
📰 Imágenes 🖈 🛛	msvcr100.dll	11/10/2018 19:11	' Extensión de la ap	756 KB	
Disco local (D:)	plugin host	01/10/2019 10:35	Aplicación	6.489 KB	
	python3.3	11/10/2018 19:11	Archivo WinRAR Z	2.567 KB	
	vthon33.dll	11/10/2018 19:11	Extensión de la ap	5.293 KB	
Informenat	subl	01/10/2019 10:35	Aplicación	441 KB	
Kit	sublime	17/09/2019 7:44	Python File	38 KB	
Creative Cloud File	sublime plugin	17/09/2019 7:44	Python File	37 KB	
	sublime text	01/10/2019 10:35	Aplicación	6.735 KB	
o Creative Cloud File	= ipdate_installer	01/10/2019 10:35	Aplicación	157 KB	
13 elementos					

Ejecutamos el siguiente archivo:

🗾 sublime_text

5	untitle	d - Sublime	Text (U	NREGIS	TERED)						_		×
<u>F</u> ile	<u>E</u> dit	<u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp				
••													▼
1	L												
													- 11
													- 11
													- 11
													- 11
													- 1
													- 1
													- 1
													- 11
													- 11
													- 11
													- 1
	Line 1	1, Column 1							٦	ab Size: 4		Plain Tex	t

Vamos a configurarlo para trabajar con Python.

Del menú seleccionaremos Tools y de este Command Palette...



Escribiremos Install Package Control.

A continuación en la misma línea escribiremos: Install Package.

Ahora le decimos que nos busque sublimeREPL.



Hacemos Enter.

Desde el menú Tools en la parte inferior aparecerá SublimeREPL.

untitled • - Sublime Text (UNREGISTERED)	_		\times
File Edit Selection Find View Goto Tools Project Preferences Help			
untitled Command Palette Ctrl+Shift+P 1 install Snippets		Ļ	T
Build System > Build Ctrl+B Build With Ctrl+Shift+B Cancel Build Ctrl+Break Build Results > Save All on Build			
Record MacroCtrl+QPlayback MacroCtrl+Shift+QSave MacroMacros			
Developer >			
SublimeREPL >			
Tab Size: - Tab Size: -	4	Plain Te	xt

untitled • - Sublime Text (UNREGISTERED)	– 🗆 🗙	-
File Edit Selection Find View Goto	Tools Project Preferences Help	Eval in REPL >	
◆ untitled ●	Command Palette Ctrl+Shift+P Snippets	OCaml >	
	Build System >> Build Ctrl+B Build With Ctrl+Shift+B Cancel Build Ctrl+Break Build Results >> ✓ Save All on Build	Ruby > Clojure > ClojureScript > CoffeeScript Common Lisp > Elixir	
	Record Macro Ctrl+Q	Python >	Python - execnet
	Playback Macro Ctrl+Shift+Q Save Macro Macros >	F# GDB Groovy	Python Python - virtualenv Python - PDB current file
	Developer >	Haskell	Python - RUN current file
	SublimeREPL >	lo	Python - IPython
		Matlab	
		Maxima MozRepl	
		Node Octave	
7 characters selected	Tab Size:	Perl	

Ahora de Tools, seleccionaremos SublimeREPL, de este seleccionaremos Python y por último Python.

Así nos quedará el editor.

File Edit Selection Find View Goto Jools Project Preferences Help Intitled "REPL*[python] x Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v. 1926 32 bit (Intel)] on win32 Type "help", "copyright", "credits" or "license" for more information. >>>	*REPL* [python] - Sublime Text (UNREGISTERED)	_	
<pre> wuntitled *REPL*[python] x Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v. 1926 32 bit (Intel)] on win32 Type "help", "copyright", "credits" or "license" for more information. >>> </pre>	<u>File Edit Selection Find View Goto Tools Project Preferences Help</u>		
<pre>Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v. 1926 32 bit (Intel)] on win32 Type "help", "copyright", "credits" or "license" for more information. >>></pre>	untitled • *REPL* [python] ×		•
Type "help", "copyright", "credits" or "license" for more information.	Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08)	[MSC v.	
Line 3 Column 5	<pre>Type "help", "copyright", "credits" or "license" for more information. >>></pre>		
Line 3. Column 5. Tab Size: 4. Python			
	Line 3 Column 5	Size: 4	Puthon

Este editor no ayudará a la hora de realizar nuestros programas.







SINTAXIS BÁSICA II Tipos, operadores y variables (VIDEO 4)

• Variables





Operadores

- Aritméticos: Suma (+), Resta (-), Multiplicación (*), División (/), Módulo (%), Exponente (**) y División entera (//).
- Comparación: Igual que (==), Diferente que (i=), Mayor que (>), Menor que (<), Mayor o igual que (>=) y Menor igual que (<=).

- Lógicos: AND, OR y NOT.
- Asignación: Igual (=), Incremento (+=), Decremento (-=), (*=), (/=), (%=), (**=) y (//=).
- Especiales: IS, IS NOT, IN, NOT IN.

Variable

Espacio en la memoria del ordenador donde se almacenará un valor que podrá cambiar durante la ejecución del programa.

A continuación vamos a ejecutar Sublime Text.



Del menú Tools seleccionaremos SublimeREPL de este Python y de nuevo Python.



Operadores

REPL [python] - Sublime Text (UNREGISTERED)	_	
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp		
★ *REPL* [python] ×		•
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [1926 32 bit (Intel)] on win32	MSC v.	
Type "help", "copyright", "credits" or "license" for more		
>>> 5+6		
11		
>>> # modulo		
>>> 10%3		
1		
>>> # exponente		
>>> 5**3		
125		
>>> # division entera		
>>> 9//2		
Line 14, Column 5 Tab Siz	te: 4	Python

Variables

Ejemplo de nombres de variables, Nombre, nombre, nombre3, mi_nombre.

El tipo de variable no lo establece el contendor, sino el contenido.

Para saber el tipo de variable utilizaremos la función type().





Utilizando las tres comillas.

>> type(numero)
class 'float'>

Line 6, Column 5

>> |



Tab Size: 4

Python

Condiciones



Operador de asignación = y operador de comparación ==.





SINTAXIS III FUNCIONES I (VIDEO 5)

- Funciones
 - o Qué es
 - o Utilidad
 - o Sintaxis
 - o Ejecución

¿Qué son?

- Conjunto de líneas de código agrupadas (bloque de código) que funcionan como una unidad realizando una tarea específica.
- Las funciones en Python pueden devolver valores.
- Las funciones en Python pueden tener parámetros/argumentos.
- A las funciones también se las denomina "método- cuando se encuentran definidas dentro de una clase.

Utilidad

• Reutilización de código (cuando sea necesario o si es necesario)

Sintaxis

- Def nombre_funcion():
 - Instrucciones de la función
 - Return (opcional)
- Def nombre_función(parámetros)
 - Instrucciones de la función
 - Return (opcional)

Ejecución

- Nombre_función() \rightarrow Si no recibe parámetros.
- Nombre_función(parámetros) → Si recibe parámetros.

A continuación vamos a ejecutar Sublime Text.

Vamos a comprobar lo siguiente:

📁 untitled	- Sublime	Text (U	NREGIS	TERED)					_			\times
File Edit	Selection	Find	View	Goto	Tools	Project	Preferences	Help				
∢ ►					0	Command	d Palette	Ctrl+Shift+P				▼
1		_	_	_	5	Snippets				_	_	
					E	Build Syste	em	>		Auto	omatic	
					E	Build		Ctrl+B		Acti	onScrip	t
					E	Build With		Ctrl+Shift+B		Ant		
					(Cancel Bu	ild	Ctrl+Break		C Si	ngle File	2
					E	Suild Resu	ilts	>		C++	Single	File
					~ 5	ave All or	n Build			Carg	jo	
					F	Record Ma	acro	Ctrl+Q		D		
					ŀ	Playback I	Macro	Ctrl+Shift+Q		D du	ıb	
					9	ave Macr	·o			Erlar	ng	
					ſ	Vlacros		>		Has	kell C	
Line 1,	Column 1					Developer		>		Java	C	
					5	SublimeRE	EPL	>		Mak	e	
									~	Pyth	ion	
										R		
										Rub	у	
										Rust	:	
										Shel	lScript	
										subl	imerepl	_build_s
										Synt	ax Tests	;
										New	/ Build S	ystem

Comprobar que estamos trabajando en el lenguaje adecuado del menú Tools, la opción Build System veamos la opción Python activada, sino la tenemos que activar.



Para guardar nuestro proyecto seleccionaremos File y de este Save.

🧧 Guardar como								×
\leftrightarrow \rightarrow \checkmark \uparrow	« Di	isco local (D:) > Cu	urso Python	~	ē	,	en Curso Python	
Organizar 👻 N	ueva c	arpeta						?
💻 Este equipo	^	Nombre	^	Fe	echa de	modificación	Тіро	Ti
Descargas			Ningún elemen	nto coincide co	n el crit	erio de búsqueda	ı.	
Documentos	÷							
📰 Imágenes								
👌 Música								
Objetos 3D	J	<						>
IN VICEOS								
No <u>m</u> bre:	Prim	eras_funciones.py						~
<u>T</u> ipo:	All Fi	les (*.*)						~
∧ Ocultar carpetas						<u>G</u> uardar	Cancela	r

Seleccionaremos la carpeta donde queremos guardar los proyectos del curso.

El nombre del proyecto tiene que tener una extensión py, seguido del botón Guardar.



Observarás que el editor colorea el texto.

Para poder ver el resultado del menú Tools seleccionamos Build o con las teclas rápidas Ctrol +B.



En la parte inferior veremos el resultado.

Si modificamos el código este lo tendremos que guardar del menú File y de este Save o con las teclas rápidas Ctrol +S.

A continuación le damos a Ctrol +B



Ahora queremos que estas instrucciones se ejecuten 5 veces.



Esto sería lo más fácil pero no lo más correcto.

Lo correcto será crear la función que esta no se ejecutará hasta que sea llamada, en el siguiente ejemplo hemos creado una función def mensaje(): que imprime tres mensaje que para que estén dentro de la función estos tienen que estar dentados hacia la derecha, a continuación esta función es llamada 5 veces.

Ctrol + S para guardar los cambios y Ctrol + B para ver el resultado.



Ahora escribe el siguiente código:

funciones predefinidas
def mensaje():

print("Estamos aprendiendo Python") print("Estamos aprendiendo intruccioines básica") print("Poco a poco iremos avanzando")

```
print("---mensaje 1---")
mensaje()
print("---mensaje 2---")
mensaje()
print("---mensaje 3---")
mensaje()
print("---mensaje 4---")
mensaje()
```

print("---mensaje 5---") mensaje()

Podrás observar que entre llamada de función y función podemos insertar más código.

Guardamos y ejecutamos.







SINTAXIS BÁSICA IV FUNCIONES II (VIDEO 6)

- Funciones
 - Paso de parámetros

Ejecutaremos Sublime Text

D:\Curso Python\Primeras_funciones.py - Sublime Text (UNREGISTERED)			×
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
Primeras_funciones.py ×			▼
1 def suma(): 2 num1=5 3 num2=7 4 print(num1+num2) 5 6 suma() 7		<u>Y</u>	
12 [Finished in 0.1s]			
Line 7, Column 1	Tab Size: 4	Python	

Basándonos en el ejemplo anterior hemos creado una función llamada suma(): en la que declaramos dos variables num1, num2 y le asignamos los respectivos valores de 5 y 7.

En la línea 6 llamamos a la función, recuerda que una función se ejecuta si es llamada.

Recuerda que una función se puede llamar las veces que la necesitemos.

En este ejemplo cada vez que llamemos la función esta sumará los mismos valores, lo ideal sería que pudiera sumar valores distintos, para ello necesitamos crear una función con parámetros.

Ctrol+S y Ctrol + B





Si en una función con parámetros utilizamos la opción return devuelve un valor como estas máquinas nos retorna un artículo.

Además si a esta máquina le hemos de introducir un código para que nos retorne un determinado artículo, la similitud aun en mayor.

El código introducido sería los parámetros que le pasamos a la función.

Def suma(mum1, num2): Resultado = num1 + num2 return resultado

print(suma(5,7))

print(suma(2,3))

print(suma(34,358))





También podemos pasar el valor de la función a una variable y utilizarla cuando la necesitemos.





SINTAXIS BÁSICA V. LAS LISTAS (VIDEO 7)

• Las listas

¿Qué son las listas?

- Estructura de datos que nos permite almacenar gran cantidad de valores (equivalente a los arrays en otros lenguajes de programación).
- En Python las listas pueden guardar diferentes tipos de valores (en otros lenguajes no ocurre esto con los array)
- Se puede expandir dinámicamente añadiendo nuevos elementos (otra novedad respecto a los arrays en otros lenguajes.

Sintaxis de las listas

nombreLista=[elem1, elem2, elem3,.....]

Índice es la posición del elemento de la lista, empezando por la posición 0.

Vamos a guardar un nuevo proyecto llamado Listas.py.



Declaración de una lista.

miLista=["María", "Pepe", "Marta", "Antonio"]

Índice	Elemento de la lista
0	María
1	Рере
2	Marta
3	Antonio

D:\Curso Python\Listas.py - Sublime Text (UNREGISTERED)	_		×
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
▲ Listas.py ×			▼
1 miLista=["María", "Pepe", "Marta", "Antonio"] 2 print(miLista[:]) #> Para imprimir toda la lista			
['María', 'Pepe', 'Marta', 'Antonio'] [Finished in 0.1s]			
Line 2, Column 51 Tab Size: 4		Python	

Para imprimir un elemento de la lista

D:\Curso Python\Listas.py - Sublime Text (UNREGISTERED)	-		×
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
<► Listas.py ×			▼
<pre>1 miLista=["María", "Pepe", "Marta", "Antonio"] 2 print(miLista[2]) #> Posición 3 de la lista</pre>			ļ
Marta [Finished in 0.1s]			
Line 2, Column 46	Tab Size: 4	Python	
		-	

Si queremos acceder a un elemento de la lista que no existe, nos generará un error.

En el ejemplo anterior la lista está entre 0-3, si hacemos:

Print(miLista[7]) \rightarrow al ser un elemento que no existe nos dará error con el siguiente mensaje:

IndexError: list index out of range.

Al poner un valor negativo la búsqueda lo hace a la inversa, empieza a contar por el valor que se encuentra a la derecha con el valor -1 y se va desplazando hacia la izquierda -2, -3, etc.

D:\Curso Python\Listas.py • - Sublime Text (UNREGISTERED)	_		\times
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
< ► Listas.py			▼
1 miLista=["María", "Pepe", "Marta", "Antonio"] 2 print(miLista[-3])			ļ
Pepe [Finished in 0.1s]			
Line 2, Column 20 Tab Size: 4		Python	

Porción de lista

D:\Curso Python\Listas.py - Sublime Text (UNREGISTERED) -		×
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp		
↓ Listas.py ×		▼
1 miLista=["María", "Pepe", "Marta", "Antonio"] 2 print(miLista[1:3])		
['Pepe', 'Marta'] [Finished in 0.1s]		
Line 2, Column 18 Tab Size: 4	Pyth	on

Accedemos los valores desde la posición 2 con el índice 1 hasta la posición 3 con el índice 2, la posición con el índice 3 no la muestra.

Si queremos mostrar desde el principio hasta el índice 2 una forma de hacerlo es la siguiente:

Print(miLista[:2], al omitir el valor de inicio entiende que quieres imprimir desde el primer valor de la lista.

Print(miLiusta[2:], al omitir el valor de fin entiende que imprimirá de la lista desde el valor que se encuentra en el índice 2 hasta el final de la lista.

Para agregar elemento en una lista utilizaremos la función append().

D:\Curso Python\Listas.py - Sublime Text (UNREGISTERED)	_		Х
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
<► Listas.py ×			▼
<pre>1 miLista=["María", "Pepe", "Marta", "Antonio"] 2 miLista.append("Sandra") 3 print(miLista[:])</pre>			
['María', 'Pepe', 'Marta', 'Antonio', 'Sandra'] [Finished in 0.1s]			
Line 2, Column 25 Tab Size: 4		Python	

Si queremos agregar un nuevo valor a la lista y no queremos que sea al final utilizaremos la función insert().



Para agregar varios elementos a la lista utilizaremos la función extend().



Para saber dónde se encuentra un elemento dentro de la lista index().

D:\Curso Python\Listas.py - Sublime Text (UNREGISTERED)	_		×
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
↓ Listas.py ×			▼
1 miLista=["María", "Pepe", "Marta", "Antonio"]		Line of the second	1
<pre>2 miLista.extend(["Sandra","Ana","Lucia"])</pre>			
<pre>3 print(miLista.index("Antonio"))</pre>			
3			
[Finished in 0.1s]			
Line 3, Column 33 Tab Size: 4		Python	

Pueden haber varios elementos con el mismo nombre pero la función index retorna el numero del primer valor que encuentra.

Para saber si un elemento se encuentra o no utilizaremos in



Como este valor se encuentra veremos que en nuestra consola imprime un True si el elemento no existe imprime un False:



Una lista puede almacenar valores de diferente tipo.

miLista=["María", 5, True, 78.35]

Para eliminar elemento de una lista remove()



Hemos eliminado Sandra de la lista.

Para eliminar el último elemento de la lista utilizaremos la función pop().



El último elemento de la lista que es Lucia ahora no se muestra.

Si queremos sumas el contenido de dos listas en una tercera lista:

D:\Curso Python\Listas.py - Sublime Text (UNREGISTERED)	_		×
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
<► Listas.py ×			▼
<pre>1 miLista1=["María", "Pepe", "Marta", "Antonio"] 2 miLista2=["Sandra","Ana","Lucia"] 3 miLista3=miLista1+miLista2 4 print(miLista3[:])</pre>			
['María', 'Pepe', 'Marta', 'Antonio', 'Sandra', 'Ana', 'Lucia'] [Finished in 0.1s]			
Line 4, Column 15 Tab Siz	e: 4	Python	

El operador + lo utilizaremos como concatenado de dos listas.

Con el operador * podemos repetir la lista el número de veces que deseemos.






SINTAXIS BÁSICA VI. LAS TUPLAS (VIDEO 8)

• Las tuplas

¿Qué son las tuplas?

- Las tuplas son listas inmutables, es decir, no se pueden modificar después de su creación.
 - No permite añadir, eliminar, mover elementos, etc (no append, extend, remove)
 - Si permite extraer porciones, pero el resultado de la extracción es una tupla nueva.
 - Si permite comprobar si un elemento se encuentra en la tupla.
- ¿Qué utilidad o ventaja tienen con respecto a las listas?
 - Más rápidas
 - Menos espacio (mayor optimización)
 - o Formatean String
 - Puede utilizarse como claves en un diccionario. (las listas no)

Sintaxis de la tuplas

nombreTupla=(elem1, elem2, elem3.....)

No se utilizan corchetes como en las listas.

Vamos a crear un nuevo proyecto llamado practica_Tupas.py



Ejemplo de Tuplas.



En la posición 3 hay un 1, recuerda que empieza por la posición 0.

Para convertir una tupla en lista vamos a utilizar el método list.

5	:\Curs	o Python\P	ractica	_Tuplas	.py - Su	blime T	ext (UNRE	GISTERED)				_		×
<u>F</u> ile	<u>E</u> dit	<u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp					
	F	Practica_Tupl	as.py	×										▼
1	. m:	itupla=	("Jua	in", 1	13, 1	, 199	5)						Hiltor	l
2	m	ilista=	list(mitu	pla)									
3	р	rint(mi	lista	ā)										
[']	uan',	13, 1,	199	5]										
	nrsue	<u>u in</u> 0.	12]											
	Line 2,	Column 13									Tab Size	:: 4	Pythor	n

Sabemos que es una lista porque está entre corchetes.

D:\Curso Python\Practica_Tuplas.py - Sublime Text (UNREGISTERED)	_		×
<u>File Edit Selection Find View Goto Tools Project Preferences Help</u>			
♦ Practica_Tuplas.py ×			▼
1 mitupla=("Juan", 13, 1, 1995) 2 milista= <i>list</i> (mitupla)		Hillin.	
3 print(mitupla)			
('Juan', 13, 1, 1995) [Finished in 0 15]			
Line 3, Column 14 Tab Size: 4		Python	1

La tupla la muestra entre paréntesis.

Para convertir una lista en tupla utilizaremos el método tuple.



Con la instrucción in podemos comprobar si hay elementos.

D:\Curso Python\Practica_Tuplas.py - Sublime Text (UNREGISTERED) -		×
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp		
♦ Practica_Tuplas.py ×		▼
1 milista=["Juan", 13, 1, 1995]	BOAK"	
2 mitupla= <i>tuple</i> (milista)		
3 print <u>(</u> "Juan" in mitupla <u>)</u>		
True [Finished in 0.1s]		
Line 3, Column 24 Tab Size: 4	Python	

El método count nos permite mostrar cuentos elementos que nosotros le preguntemos hay en una tupla.



En este caso el numero 13 está 2 veces.

El método len nos permite averiguar la longitud de una tupla.

D:\Curso Python\Practica_Tuplas.py - Sublime Text (UNREGISTERED)	-		\times
<u>File Edit Selection Find View Goto Tools Project Preferences Help</u>			
♦ Practica_Tuplas.py ×			▼
1 milista=["Juan", 13, 1, 1995, 13]		Harry -	
2 mitupla=tuple(milista)			
3 print(len(mitunla))			
5			
[Finished in 0.1s]			
		D 11	
Line 3, Column 20	Tab Size: 4	Python	

Tupla con un único elemento, tupla unitaria.

D:\Curso Python\Practica_Tuplas.py - Sublime Text (UNREGISTERED) -												×	
<u>F</u> ile <u>E</u> dit	Selection Find	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp						
↓	Practica_Tuplas.py	×											▼
<u> </u>	nitupla=("յս	ian" ,)										PH 252	1
2 r	orint(len(mi	tupla))										
1 [Finish	ed in 0.1s]												
Line 2	2, Column 1; Build f	inished							Т	ab Size: 4		Python	1

Empaquetado de tupla, consiste en definir una tupla sin los paréntesis:

Mitupla="Juan", 13, 1, 1995

Desempaquetado de tupla:







SINTAXIS BÁSICA VII. LOS DICCIONARIOS (VIDEO 9)

• Los diccionarios

¿Qué son los diccionarios?

- Estructura de datos que nos permiten almacenar valores de diferente tipo (enteros, cadena de texto, decimales) e incluso listas y otros diccionarios.
- Las principal característica de los diccionarios es que los datos se almacenan asociados a una clave de forma que se crea una asociación de tipo clave : valor para cada elemento almacenado.
- Los elementos almacenados no están ordenados. El orden es indiferente a la hora de almacenar información en un diccionario.

Vamos a crear un nuevo proyecto llamado Diccionarios.py.

La sintaxis:

midiccionario={"Alemania":"Berlín", "Francia":"París", "Reino Unido":"Londres", "España":"Madrid"}



Para imprimir todo el diccionario.

D:\Curso Python\Diccionarios.py - Sublime Text (UNREGISTERED)	_		×
<u>File Edit Selection Find View Goto Tools Project Preferences Help</u>			
✓ Diccionarios.py ×			▼
<pre>1 midiccionario={"Alemania":"Berlín","Francia":"París", "Reino Unido":"Lo 2 print(midiccionario)</pre>	ondres		
{'Alemania': 'Berlín', 'Francia': 'París', 'Reino Unido': 'Londres', 'España' [Finished in 0.1s]	: 'Ma	drid'}	
Line 2, Column 20; Build finished Tab Size: 4		Python	

Agregar más elementos a un diccionario.



En la línea cuatro lo hemos sobre escrito. Ahora vamos a eliminar elementos a un diccionario (del).



También podemos crear un diccionario con distintos tipos de valores:

Midiccionario={"Alemania":"Berlín", 23:"Jordan", "Mosaqueteros":3}

Convertir una tupla a un diccionario:



Para imprimir un valor del diccionario:



Crear un diccionario que además contiene una lista.

midiccionario={23:"Jordan", "Nombre":"Michale", "Equipo":"Chicago", "anillos":[1991,1992, 1993, 1996, 1997,1998]}



Ahora queremos consultar el Equipo y los anillos, recuerda que anillos en una lista.

🗾 D	:\Curs	so Python\D	icciona	arios.py	- Sublir	ne Text	(UNREGIS	TERED)			_		×
<u>F</u> ile	<u>E</u> dit	<u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp				
.		Diccionarios.	ру	×									▼
1 2 3	p p	idiccio print(mio print <u>(</u> mio	nario dicci dicci	e={23 .onari .onari	:"Jord io["Ed io["ai	dan", quipo nillo	"Nomb "]) s"] <u>)</u>	re":"Mich	ale",	"Equipo":"Chicago",	"anill		
Chic [199 [Fir	cago)1, nish	1992, 19 ed in 0.	93, 1s]	1996,	1997	, 199	98]						
	Line 3	, Column 32								Tab Size:	4	Python	

Si en lugar de una lista queremos agregar una tupla:

midiccionario={23:"Jordan", "Nombre":"Michael","Equipo":"Chicago",

"anillos":{"temporada":[1991,1992,1993,1996,1997,1998]}}



Para consultar las claves del diccionario utilizaremos keys()

D:\Curso Python\Diccionarios.py - Sublime Text (UNREGISTERED)	_		\times
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
♦ Diccionarios.py ×			▼
<pre>1 midiccionario={23:"Jordan", "Nombre":"Michael","Equipo":"Chicago", "anillo 2 print(midiccionario.keys()) 3 4</pre>	s":{		
dict_keys([23, 'Nombre', 'Equipo', 'anillos']) [Finished in 0.1s]			
Line 3, Column 1 Tab Size: 4		Pythor	

Si queremos imprimir los valores de las claves utilizaremos values()

Sublime Text (UNREGISTERED) -											
<u>File E</u> dit <u>S</u> election Find <u>V</u> iew <u>G</u> oto <u>T</u> ools <u>P</u> roject Prefere <u>n</u> ces <u>H</u> elp											
✓ ► Diccionarios.py ×			▼								
<pre>1 midiccionario={23:"Jordan", "Nombre":"Michael","Equipo":"Chicago", "ani 2 print(midiccionario.values()) 3</pre>	11os":{										
dict_values(['Jordan', 'Michael', 'Chicago', {'temporada': [1991, 1992, 1993, 1996, 1997, 1998]}]) [Finished in 0.1s]											
Line 2, Column 27 Tab Size	: 4	Python	1								

Si queremos saber la longitud de diccionario utilizaremos len().

5	D:\Cur	so Python\D	iccion	arios.py	- Subli	me Text	(UNREGIS	TERED)				_		×
<u>F</u> ile	<u>E</u> dit	<u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp					
••		Diccionarios.	ру	×										▼
1 2 3	. n . p	idiccio rint <u>(</u> le	nario n(mio	o={23: diccio	"Jor onari	dan", o) <u>)</u>	"Nomb	re":"Mich	ael",	"Equipo":"C	hicago",	"anillos":{		
4 [Fi	nish	ed in 0.	.1s]											
	Line 2	, Column 26	; Build f	inished						_		Tab Size: 4	Pythor	1

Basándonos de la siguiente información:

Formula para calcular NIF

La formula para calcular la letra del DNI y obtener el NIF es la siguiente:

Tomamos el número completo de hasta 8 cifras de nuestro DNI, lo dividimos entre 23 y nos quedamos con el resto de dicha división, o dicho de otro modo, **calculamos el módulo 23 del DNI**.

El resultado anterior es un número entre 0 y 22. A cada uno de estos posibles números le corresponde una letra, según la siguiente tabla:

RESTO	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
LETRA	Т	R	W	А	G	М	Y	F	Р	D	Х	В	Ν	J	Ζ	S	Q	V	Н	L	С	K	E

Crea dos proyecto uno utilizando diccionarios y otro utilizando listas, el usuario tiene que introducir su documento nacional de identidad sin la letra y el programa tiene que imprimir la letra del documento.

Trabajando con Diccionarios

D:\Curso Python\nif.py - Sublime Text (UNREGISTERED)		×
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp		
▲ ► nif.py × *REPL* [python] ×		▼
<pre>1 letra={0:"T",1:"R",2:"W",3:"A",4:"G",5:"M", 6:"Y", 7:"F", 2 8:"P",9:"D",10:"X", 11:"B", 12:"N", 13:"J", 14:"Z", 15:"S", 3 16:"Q", 17:"V", 18:"H", 19:"L", 20:"C", 21:"K", 22:"E"} 4 dni=input("Dime tu DNI para que calcule la letra: ") 5 dni1=int(dni) % 23 6 print("La letra es " + letra[dni1])</pre>		
Line 6, Column 36 Tab Size: 4	Pytho	n

El resultado será:



Puede probar con tu número del documento nacional de identidad.

Trabajando con listas



Ejecutamos la consola.



Puedes probar si funciona con tu número del documento nacional de identidad.





CONDICIONALES I (VIDEO 10)

- Condicionales
 - o Instrucción IF



Operadores de comparación:

Mayor que \rightarrow ">" Menor que \rightarrow "<" Igual \rightarrow "==" Mayor igual que \rightarrow ">=" Menor igual que \rightarrow "<=" Distinto \rightarrow "!="

Vamos a crear un nuevo proyecto llamado practicalf.py.



Para este programa funcione hemos de realizar los siguientes pasos: Del menú Tools, seleccionaremos SublimeREPL, de este Python y por último Python – RUN current file.



Observamos un error porque los valores que se introducen por input son de texto, al llegar la condición IF el programa no lo entiende, para ello vamos a utilizar la función int().



Repetimos Del menú Tools, seleccionaremos SublimeREPL, de este Python y por último Python – RUN current file.



Repetimos Del menú Tools, seleccionaremos SublimeREPL, de este Python y por último Python – RUN current file.



También es correcto "nota_alumno=input("Introduce la nota del alumno: ")

Definición de ámbito: significa que no podremos obtener su valor, así como su modificación si no estamos en el ámbito de la variable.





CONDICIONALES II (VIDEO 11)

- Condicionales
 - Instrucción IF. Instrucciones else y elif

La función else \rightarrow si no es verdad.

Vamos a crear un proyecto nuevo.



Ejecutamos la consola: Del menú Tools, seleccionaremos SublimeREPL, de este Python y por último Python – RUN current file.

	REDI *	[nython] -	Sublim	e Tevt (l	INREG	STERED)					_	Π	×
		python	Jubiin		/ AILEON								\sim
<u>F</u> ile	<u>E</u> dit	Selection	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp				
••	Γ	Practica_if_e	lse_elif.p		*R	EPL* (pyt	hon]	×					▼
Veri	ific	ación de	acc	es								101115	
Intr	odu	ce tu ed	lad,	por f	avor	21							
Pued	les	pasar											
***	Rep1	Closed*											
	Line (5, Column 1							Ta	ab Size: 4		Python	

Si eres mayor de 18 años puedes pasar.



Si eres menor de 18 años no puedes pasar.



La línea 8 se ejecutará siempre ya que esta se encuentra fuera del if.

REPL [python] - Sublime Text (UNREGISTERED)	_		Х
<u>File Edit Selection Find View Goto Tools Project Preferences Help</u>			
Practica_if_else_elif.py × *REPL* [python] ×			▼
Verificación de acces Introduce tu edad, por favor 25 Puedes pasar El programa ha finalizado ****Repl Closed***		Part of the second seco	
Line 7, Column 1 Ta	ab Size: 4	Python	

Para controlar más condiciones utilizaremos al función elif.

Queremos controlar que si el usuario introduce una edad superior a 100 que nos aparezca un mensaje de Edad incorrecta.



Ejecutamos la consola:



El siguiente ejemplo nos valorará la nota.



Ejecutaremos la consola e introduciremos una nota.

REPL [python] - Sublime Text (UNREGISTERED)	_		Х
<u>File Edit Selection Find View G</u> oto <u>T</u> ools <u>P</u> roject Prefere <u>n</u> ces <u>H</u> elp			
▲ Practica_if_else_elif.py × *REPL* [python] ×			▼
Verificación de nota Introduce tu nota, por favor: 9 Sobresaliente ***Repl Closed***		The sec	
Line 6, Column 1 Tab Size:	4	Python	

Vamos a realizar los siguiente ejercicios:



Ejercicios Python

Ejercicio 1:

 Crea un programa que pida dos números por teclado. El programa tendrá una función llamada "DevuelveMax" encargada de devolver el número más alto de los dos introducidos.

Ejercicio 2:

 Crea un programa que pida por teclado "Nombre", "Dirección" y "Tfno". Esos tres datos deberán ser almacenados en una lista y mostrar en consola el mensaje: "Los datos personales son: nombre apellido teléfono" (Se mostrarán los datos introducidos por teclado).

Ejercicio 3:

 Crea un programa que pida tres números por teclado. El programa imprime en consola la media aritmética de los números introducidos.

Las soluciones se encuentran en la siguiente página.





Solución ejercicio 1:



Solución ejercicio 2:



Solución ejercicio 3:



CONDICIONES III (VIDEO 12)

- Condicionales
 - Switch ¿Switch?...
 - Concatenación de operadores de comparación

Python prescinde de Switch.

¿Cómo conseguir un acceso directo a Python - RUN current file?



Del menú Preferences seleccionamos Key Bindings.

Copiaremos el siguiente texto:

```
{ "keys": ["ctrl+alt+r"], "command": "run_existing_window_command", "args":
{
    "id": "repl_python_run",
    "file": "config/Python/Main.sublime-menu"
}}
```

C\Users\pmyer\Deskton\SublimeText\Data\Packages\User\Default (Windows) sublime-keyman - SublimeText (UNREGISTERED)		_	пх
E consequences of the second product of the			
Petaut (Windows) sublime-keymap - Default x Default (Windows) sublime-keymap - User			•
<pre> 1 [{ "keys": ['ctrl+shift+n'], "command": "new_window" }, 3 { { "keys": ['ctrl+shift+n'], "command": "close_window' }, 4 { { "keys": ['ctrl+shift+n'], "command": "run_existing_window' }, 5 { { "keys": ['ctrl+shift+n'], "command": "run_existing_window' }, 5 { { "keys": ['ctrl+shift+n'], "command": "new_last_file" }, 6 { { "keys": ['ctrl+shift+n'], "command": "new_last_file" }, 7 { { { "keys": ['ctrl+shift+n'], "command": "new_last_file" }, 7 { { { { "keys": ['ctrl+shift+n'], "command": "new_last_file" }, 7 { { { { { { [thest_file", "angs": ['ctrl+shift+n'], "command": "new_last_file" }, 7 { { { { { { [thest_file", "angs": ['ctrl+shift+n'], "command": "new_last_file" }, 7 { { { { { { { [thest_file", "angs": ['ctrl+shift+n'], "command": "new_last_file" }, 7 { { { { { { { { { { { { { { { { { { {</pre>	indow_command'	, "args":	
Line 6, Column S		Tab Size: 4	JSON
			~
	_		~
Default (Windows).sublime-keymap — User •			▼
Default (Windows).sublime-keymap — User		STREET SHOT SH	•
Default (Windows).sublime-keymap — User 1 [2 [4 [".	Service Service 2700000.com	•
<pre>Default (Windows).sublime-keymap — User • 1 [2 { "keys": ["ctrl+alt+r"], "command": "run_existing_window_command", "ar</pre>	gs":	Stranger unt un 27/0000 mener	•
<pre>Default (Windows).sublime-keymap — User • 1 [2 { "keys": ["ctrl+alt+r"], "command": "run_existing_window_command", "ar 3 {</pre>	gs":	No room war war S ⁷⁰ SSS home	•
<pre>Default (Windows).sublime-keymap — User • 1 [2 { "keys": ["ctrl+alt+r"], "command": "run_existing_window_command", "ar 3 { 4 "id": "ren] python run".</pre>	gs":	Not describe your two Not describe your of the second of Not describe your of the second of the seco	•
<pre>Default (Windows).sublime-keymap — User • 1 [2 { "keys": ["ctrl+alt+r"], "command": "run_existing_window_command", "ar 3 { 4 "id": "repl_python_run", 5 "file": "sonfig(D):then (Main_sublime_menu")</pre>	gs":	The reserve war too	•
<pre>Default (Windows).sublime-keymap — User • 1 [2 { "keys": ["ctrl+alt+r"], "command": "run_existing_window_command", "ar 3 { 4</pre>	gs":	an Paurite suri tun Bi Tabalkumur	
<pre>Default (Windows).sublime-keymap — User • 1 [2 { "keys": ["ctrl+alt+r"], "command": "run_existing_window_command", "ar 3 { 4</pre>	gs":	the finance band to a grant of the second se	
<pre>Default (Windows).sublime-keymap — User • 1 [2 { "keys": ["ctrl+alt+r"], "command": "run_existing_window_command", "ar 3 { 4 "id": "repl_python_run", 5 "file": "config/Python/Main.sublime-menu" 6 }} 7]</pre>	gs":	t net finance, band to or g 20-76/0000	•
<pre>Default (Windows).sublime-keymap — User • 1 [2 { "keys": ["ctrl+alt+r"], "command": "run_existing_window_command", "ar 3 { 4 "id": "repl_python_run", 5 "file": "config/Python/Main.sublime-menu" 6</pre>	gs":	The Paperty way have	

Del menú seleccionares File y de este Save.

Cada vez que presionemos Ctrol + Alt + R ejecutaremos Python – Run current file.

Para cerrar consola Ctrol + W.

Creamos un nuevo proyecto:

50	D:\Curso Python\Practicas_condicional.py - Sublime Text (UNREGISTERED)						_		×				
<u>F</u> ile	<u>E</u> dit	<u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp				
•		Practicas_co	ndiciona	al.py ×									▼
1													
	Line 1	1, Column 1								Tab Size: 4		Python	

Vamos a utilizar una concatenación de condiciones que pare este ejemplo es:

If 0<edad<100:

Si la primera condición que 0 es menor que edad hace la siguiente comparación que es edad menor de 100, si este se cumple imprimirá "Edad es correcta" sino se cumple imprimirá "Edad incorrecta.



Si por error ponemos un número negativo este será el mensaje:



Si ponemos una edad mayor de 100



Este será el resultado:

REPL [python] - Sublime Text (UNREGISTERED)	_		×
<u>File Edit Selection Find View Goto Tools Project Preferences Help</u>			
Practicas_condicional.py × *REPL* [python] ×			▼
Edad incorrecta			
Repl Closed			
Line 4, Column 1	Tab Size: 4	Python	

Queremos comprobar si el salario de nuestros empleados cobran según su cargo, es decir bien equiparados para ello vamos a realizar el siguiente proyecto:



El salario del presidente tiene que ser el mayor, el siguiente el director, a continuación el jefe de área y por último el administrativo, si este criterio se respeta saldrá el mensaje "Salarios equiparados" y si no se cumple el mensaje que saldrá "Salarios no equiparados".

Ejecuta la consola y observa el resultado.



Ahora vamos a decirle que el administrativo cobra más que el jefe de área.

FREPL* [python] - Sublime Text (UNREGISTERED)	_		×
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
Practicas_condicional.py × *REPL* [python] ×			▼
Introduce salario presidente: 3500			
Salario presidente: 3500			
Introduce salario director: 2500			
Salario director: 2500			
Introduce salario jefe area: 2000			
Salario jefe area: 2000			
Introduce salario administrativo: 2100			
Salario administrativo: 2100			
Salarios no equiparados			
Repl Closed			
Line 12, Column 1 Tab Size: 4		Pytho	n





CONDICIONALES IV (VIDEO 13)

- Condicionales
 - Operadores lógicos "and" y "or"
 - Operador "in"

Vamos a realizar un proyecto para valorar si un alumno tiene o no derecho a beca.

Vamos a valorar la distancia superior a 40 km., tengan más de 2 hermanos en el centro y el salario familiar sea menor o igual a 20.000 €. Tiene que cumplir los tres requisitos.



Ejecutamos la consola



Ahora ejecuta de nuevo la consola y pon los siguientes valores 65 Km, 1 hermano y salario familiar 18000 €.

Para que puedas tener beca y como valor prioritario sea el salario_familiar, es decir que si cobra igual o menos de 20000 ya se te pueda otorgar la beca.



Ejecutamos la consola y ahora solo cumplirá el salario familiar.



El alumno a de elegir una asignatura optativa.

El operador in da por verdadero si introducen una de las tres asignaturas especificadas.



Ejecutamos la consola:



Ahora vamos a poner una asignatura que no existe en la relación.



Python es case senstitive es decir diferencia las mayúsculas de las minúsculas es para que lo escribáis igual.

Con la función lower() convierte a minúsculas.

Con la función upper() lo transforma a mayúsculas.



En la línea 5 lo hemos puesto toda la línea en minúsculas.

Con la función lower() paso a la variable asignatura en minúscula, si yo escribo en mayúsculas no lo tendrá en cuenta.

Ejecutamos la consola:







BUCLES I. FOR (VIDEO 14)

- Bucles. Tipos
 - Bucle for

Si no sabemos lo que es un bucle y quereos imprimir tres veces un mensaje, no nos queda más remedio que repetirla 3 veces.

Código
Código 😡
Código
Print "ejemplo de mensaje"
Print "ejemplo de mensaje"
Print "ejemplo de mensaje"
Código
Código
Código

El programa funcionaría, pero esto no es eficiente ni recomendable.

Bucles

Determinados

- Se ejecutan un número determinado de veces.
- Se sabe a priori cuantas veces se va a ejecutar el código del interior del bucle.

Indeterminados

- Se ejecutan un número indeterminado de veces.
- No sabe a priori cuantas veces se va a ejecutar el código del interior del bucle.
- El número de veces que se ejecutará dependerá de las circunstancias durante la ejecución del programa.

Bucles	
Instrucción Instrucción Instrucción	-
Declaración del bucle: Cuerpo del bucle	
Instrucción	
Instrucción	
Instrucción	

Un error que podemos hacer es un bucle infinito, este no sale y no ejecutaría el resto del líneas que tiene el programa.

Bucle for (determinado).Sintaxis

For variable in elemento a reccorrer:

Cuerpo del bucle

Identación

Elemento a recorrer: puede ser una Lista tupla, cadena de texto, etc.

Después del for encontramos una identación esto nos dirá las líneas que pertenecen al bucle.

Vamos a realizar un nuevo proyecto:



La variable i va asumiendo los valores de la lista, empezando por primavera y terminando con invierno.







BUCLES II BUCLE FOR (VIDEO 15)

- Bucle for
 - o Recorido string
 - o Tipo rage
 - Notaciones especiales con print

D:\Curso Python\Practica_For.py - Sublime Text (UNREGISTERED)	_		×
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
✓ Practica_For.py ×			▼
1 for i in["Pildoras", "Informática", 3]: 2 print <u>(</u> "Hola " <u>)</u>		Think area a	
Hola Hola Hola			
[Finished in 0.1s]			
Line 2, Column 17 Tab Size: 4		Pytho	n

Repite e bucle tres veces porque la lista tiene tres elementos.

Ahora en la función print vamos a utilizar el argumento end.

D:\Curso Python\Practica_For.py - Sublime Text (UNREGISTERED)	_		×
<u>File Edit Selection Find View Goto Tools Project Preferences Help</u>			
✓ Practica_For.py ×			▼
1 for i in["Pildoras", "Informática", 3]: 2 print <u>(</u> "Hola", <i>end=</i> "" <u>)</u>			
HolaHolaHola[Finished in 0.1s]			
Line 2, Column 16 Tab Si	ze: 4	Pytho	n

De este modo omitimos el salto de línea, ahora para que la palabra hola tenga un espacio en el argumento end pondremos igual comillas espacio comillas.

D:\Curso Python\Practica_For.py - Sublime Text (UNREGISTERED)	_		×
<u>File Edit Selection Find View Goto Tools Project Preferences Help</u>			
✓ Practica_For.py ×			▼
1 for i in["Pildoras", "Informática", 3]: 2 print <u>(</u> "Hola", <i>end=</i> " " <u>)</u>		a Third Park	
Hola Hola [Finished in 0.1s]			
Line 2, Column 27; Build finished Tab Siz	.e: 4	Pythor	ı


Ha escrito la palabra hola tantas veces como caracteres tiene el correo electrónico.

Ahora vamos a verificar si un correo electrónico en falso o no.

D:\Curso Python\Practica_For.py - Sublime Text (UNREGISTERED)									
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp									
♦ Practica_For.py ×			▼						
<pre>1 email=False 2 for i in "juan@pildorasinformatica.es": 3 if(i=="@"): 4 email=True 5 6 if email==True: 7 print("email es correcto") 8 else: 9 print<u>(</u>"el email no es correcto") 10</pre>									
email es correcto [Finished in 0.1s]									
Line 9, Column 37	ab Size: 4	Python							

Ahora vamos a borrar la arroba.



En esta parte de código:

if emai	l==True:	if email:				
	print("email es correcto")		print("email es correcto")			
else:		else:				
	print("el email no es correcto")		print("el email no es correcto")			

Si la línea "if email==True:" lo cambiamos por "if email:" al ser una variable Boleana funcionará correctamente. Es decir que por defecto la reconoce como True.



Recuerda que si utilizamos == es para un operador de comparación al operador de asignación que es =.

Ahora vamos a realizar un proyecto en el que nos preguntará por una dirección de correo y que la pueda evaluar.



Vamos a ejecutar la consola



Ahora introduce un correo que no tenga arroba @ y podrás ver como es email no es correcto.



Ahora para verificar que el correo electrónico que introducimos es correcto, es decir que encuentra la arroba y el punto.



Ahora si omites la arroba o el punto el mensaje será "el email no es correcto".

Tipo range:

D:\Curso Python\Practica_For.py - Sublime Text (UNREGISTERED) - D										×		
<u>F</u> ile <u>E</u> dit	<u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp				
	Practica_For.	py rand	×								100	•
2	print	t <u>(</u> "hc	ola")	•								
hola hola hola hola hola [Finish	ed in Ø.	1s]										
Line 2	2, Column 18								Tab Size: 4		Python	

Si cambiamos por i veremos el valor en cada bucle.







BUCLES II BUCLE FOR (VIDEO 16)

Realiza los siguientes ejercicios:



Ejercicios Python

Ejercicio 1:

 Crea un programa que muestre los números impares del 1 al 100. Los números deberán aparecer una al lado del otro sin salto de línea.

Ejercicio 2:

 Crea un programa que pida por teclado introducir una contraseña. La contraseña no podrá tener menos de 8 caracteres ni espacios en blanco. Si la contraseña es correcta, el programa imprime "Contraseña OK". En caso contrario imprime "Contraseña errónea"

Ejercicio 3:

 Crea un programa que evalúe si una dirección de correo electrónico es válida o no en función de si tiene "@" o "." Hay que tener en cuenta que la dirección se considera correcta si solo tiene una "@" y si tiene uno o más "."

(Soluciones en la página siguiente)

Soluciones:

Yo lo he resuelto así:

D:\Curso Python\ejercicio4.py - Sublime Text (UNREGISTERED)	_		Х
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
✓ ► ejercicio4.py ×			▼
<pre>1 for i in range(101): 2 if (i/2!=i//2): 3 print (i, end=(" ")) 4</pre>		- Stars	
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 59 61 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91 93 95 97 99 [Finish	51 53 ed in	55 5 0.1s	7]
Line 2, Column 14 Tab Size: 4		Python	

Propuesta del profesor:

Los ejercicios se deben hacer utilizando exclusivamente los conceptos vistos hasta el vídeo 11 del curso. Recuerda que "En programación siempre hay varios caminos para llegar a Roma". Las soluciones que se proponen a continuación son válidas, pero no las únicas posibles. Tal vez tus programas funcionen con códigos diferentes.

Solución Ejercicio 1:



Yo lo he resulto así:



Propuesta del profesor.

Solución Ejercicio 2

4.	Practica for.py ×
1	<pre>contra=input("Introduce contraseña: ")</pre>
2	
3	contador=0
4	
5	<pre>for i in range(len(contra)):</pre>
6	
7	<pre>if contra[i]==" ":</pre>
8	
9	contador=1
10	
11	if len(contra)<8 or contador>0:
12	
13	<pre>print("Contraseña errónea")</pre>
14	
15	else:
16	
17	<pre>print("Contraseña correcta")</pre>
18	
19	

Yo lo he resuelto así:



Propuesta del profesor:

Solución Ejercicio 3

41	Practica for.py ×
1	<pre>email=input("Introduce email: ")</pre>
2	
3	contadorArroba=0
4	contadorPunto=0
5	
6	<pre>for i in range(len(email)):</pre>
7	
8	<pre>if email[i]=="@":</pre>
9	
10	contadorArroba=contadorArroba+1
11	
12	<pre>if email[i]==".":</pre>
13	
14	contadorPunto=1
15	
16	if contadorPunto==0 or contadorArroba!=1:
17	
18	print("email incorrecto")
19	
20	else:
21	2 4 4 H
22	print ("email correcto")
23	
24	

- Bucle for
 - o Tipo range
 - Notaciones especiales con print



Si en la función print usamos el parámetro "f" nos permite concatenar un texto con una viariale que irá entre llaves.



Si en el rango modificamos poniendo (5,10), los valores de la variable i será desde el 5 hasta el 9.

D:\Curso Python\Practica_For.py - Sublime Text (UNREGISTERED)	_		(
<u>File Edit Selection Find View Goto Tools Project Preferences Help</u>			
◆ Practica_For.py ×			•
1 for i in range(5,50,3): 2 print(f"valor de la variable {i}")		" MARTIN and an	ļ
valor de la variable 5			
valor de la variable 8			
valor de la variable II			
valor de la variable 14			
valor de la variable 17			
valor de la variable 23			
valor de la variable 26			
valor de la variable 29			
valor de la variable 32			
valor de la variable 35			
valor de la variable 38			
valor de la variable 41			
valor de la variable 44			
Valor de la Variable 4/			
Line 1, Column 22	Tab Size: 4	Python	

Si en rango modificamos los valores por 5,50,3 los valores que asumirá la variable i empezará por el 5 y al siguiente un incremento de 3 hasta llegar a un valor inferior a 50, en este ejemplo llega al 47.

La función len() nos retorna el número de caracteres que tiene una variable.

Len("Juan") \rightarrow 4







BUCLES IV. BUCLE WHILE (VIDEO 17)

- Bucle While
 - o Sintaxix

While (Sintaxis)

While condición:

Cuerpo del bucle

En un bucle indeterminado porque no sabemos las veces que se ejecutará.

Vamos con un nuevo proyecto:



Vamos a crear un proyecto, que hasta que no introduzcamos una edad correcta este nos irá preguntando.



Si ejecutamos la consola este será el resultado:



Con esta función también puedes utilizar los operadores de comparación y lógicos que utilizamos con la sentencia if().

Queremos controla que la edad correcta sea entre 5 y 100 años y si no es así que pregunte de nuevo.



Si ejecutamos la consola este será el resultado:



¿Cómo hacer para que un bucle no sea infinito?



Vamos a introducir tres valores negativos.



Vamos a introducir un valor negativo y luego otro positivo.







BUCLES V. CONTINUE, PASS Y ELSE (VIDEO 18)

• Continue, pass y else



Este será el resultado:





En la línea 2 le estamos diciendo que cuando encuentra la "h" continue, esto quiere decir que salte el resto de código y vuelva de nuevo al ciclo.

REPL [python] - Sublime Text (UNREGISTERED)								
<u>File Edit Selection Find View Goto Tools Project Preferences Help</u>								
♦ Practica_While.py × *REPL* [python] ×			▼					
Viendo la letra: P								
Viendo la letra: y								
Viendo la letra: t								
Viendo la letra: o								
Viendo la letra: n								
Repl Closed								
Line 8, Column 1 Tab	Size: 4	Python						

Por ese motivo la letra h no se visualiza.

En el siguiente texto "Píldoras Informáticas" si utilizamos la función len() esta nos retornará el número de caracteres incluido el espacio en blanco que se encuentra entre las dos palabras ya que el espacio en blanco también es un carácter.

Len(Píldoras Informáticas) es igual a 21.

Vamos a realizar un proyecto que nos diga el número de caracteres que hay sin contar el espacio en blanco.



Pass

Por ultimo tenemos a pass, que tal como su nombre lo indica es una operación nula, o sea que no pasa nada cuando se ejecuta. Se utiliza cuando se requiere por sintaxis una declaración pero no se quiere ejecutar ningún comando o código. También se utiliza en lugares donde donde el código irá finalmente, pero no ha sido escrita todavía (utilizándolo como un relleno temporal, hasta que se escriba el código final).

D:\Curso Python\Practica_While.py - Sublime Text (UNREGISTERED) -								
<u>File Edit Selection Find View Goto Tools Project Preferences Help</u>								
♦ Practica_While.py ×			▼					
<pre>1 # Primer ejemplo 2 for letra in "Python": 3 if letra == "h": 4 pass 5 print ("Letra actual :" + letra) 6 7</pre>		San						
Letra actual :P Letra actual :y Letra actual :t Letra actual :h Letra actual :o Letra actual :n [Finished in 0.1s]								
Line 7, Column 1	lab Size: 4	Python						

Segundo ejemplo:

D:\Curso Python\Practica_While.py - Sublime Text (UNREGISTERED)	_		\times
<u>F</u> ile <u>E</u> dit <u>S</u> election Find <u>V</u> iew <u>G</u> oto <u>T</u> ools <u>P</u> roject Prefere <u>n</u> ces <u>H</u> elp			
Practica_While.py ×			▼
<pre>3 while var > 0: 4 var = var -1 5 if var == 5: 6 pass 7 print ("Valor actual de la variable :" + str(var)) 8 9 print ("fin del script")</pre>			
Valor actual de la variable :9 Valor actual de la variable :8 Valor actual de la variable :7 Valor actual de la variable :6 Valor actual de la variable :5 Valor actual de la variable :4 Valor actual de la variable :3 Valor actual de la variable :2 Valor actual de la variable :1 Valor actual de la variable :0 fin del script [Finished in 0.1s]			
Line 9, Column 25 Tab Size: 4		Python	

D:\Curso Python\Practica_While.py • - Sublime Text (UNREGISTERED) - [
<u>F</u> ile <u>I</u>	dit <u>S</u> election F <u>i</u> nd <u>V</u> iew <u>G</u> oto <u>T</u> ools <u>P</u> roject Prefere <u>n</u> ces <u>H</u> elp							
4 ►	Practica_While.py			▼				
1	<pre>email=input("Intrduce tu email, por favor: ")</pre>							
2								
3	for i in email:							
4	if i=="@":							
5	arroba=True							
6	break;							
7	else: 🔶							
8	arroba=False							
9								
10	print(arroba)							
	ine 10, Column 14	Tab Size: 4	Python					

Como podemos ver la instrucción else está alineada con el for, su finalidad consiste que cuando ha completado el bucle irá a ejecutar las instrucciones que hay después del else.

Si en el ciclo for hay una condición que cuando se cumple ejecuta la instrucción break, ignorará las instrucciones que hay después del else.

Cuando ejecutemos la consola e introduzcamos un correo electrónico si este no tiene arroba arroba será a False.



Si introducimos un correo con su correspondiente arroba nos dará True.







GENERADORES I (VIDEO 19)

- Generadores
 - ¿Qué son?
 - o Funcionamiento
 - ¿Qué utilidad tienen?
 - o ¿Cuál es su sintaxis?

Generadores ¿Qué son?

- Estructuras que extraen valores de una función y se almacenan en objetos iterables (que se pueden recorrer).
- Estos valores se almacenan de uno en uno.
- Cada vez que un generador almacena un valor, esta permanece en un estado pausado hasta que se solicita el siguiente. Esta característica es conocida como "suspensión de estado".



Generadores ¿Qué utilidad tienen?

- Son más eficientes que las funciones tradicionales.
- Muy útiles con listas de valores infinitos.
- Bajo determinados escenarios, será muy útil que un generador devuelva los valores de uno en uno.

Generador. Sintaxis

Def generaNumeros():

Yield números

Ahora veremos un caso practico creando un función que después convertiremos a generador.

D:\Curso Python\practica_generador.py - Sublime Text (UNREGISTERED)							_		×				
<u>F</u> ile	<u>E</u> dit	<u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp				
.	Γ	practica_gen	erador.	oy ×									▼
1 2	(<i>def</i> gene	raPar	res(L	imite):							
3 4		num=	1										
5 6		miLi	sta=[]									
7		whil	e nur	n≺limi	ite:								
9 10			miLis num=r	ta.ap num+1	pend	(num*	2)						
11 12 13		retu	<mark>rn m</mark> i	Lista	à								
14 15	13 14 print(generaPares(10)) 15												
16													
[2, [Fir	[2, 4, 6, 8, 10, 12, 14, 16, 18] [Finished in 0.1s]												
	Line	14, Column 2	3							Tab Size: 4		Python	

Este mismo ejemplo usando un generador.

D:\Curso Python\practica_generador.py - Sublime Text (UNREGISTERED)	_		×
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
practica_generador.py ×			▼
1 def generaPares(limite):			
2 3 num=1 4			1
5 while num <limite:< td=""><td></td><td></td><td></td></limite:<>			
7 yield num*2 8			
9 num=num+1 10			
11 devuelvePares=generaPares(10) 12			
13 for i in devuelvePares: 14			
15 print(i, end=" ")			
10			<u> </u>
2 4 6 8 10 12 14 16 18 [Finished in 0.1s]			
Line 11, Column 18	lab Size: 4	Pythor	1

Podrás observar que el resultado es el mismo.

Ahora solo quiero imprimir los tres primeros elementos.



Este método ahorra recursos en el sistema.





GENERADORES II (VIDEO 20)

- Nueva instrucción yield from
- Utilidad: Simplificar el código de los generadores en el caso de utilizar bucles anidados.





El resultado:



¿Cómo podemos acceder a cada una de las letras que forman parte de los elementos?



El resultado:



Otra forma de hacerlo más simplificado:







EXCEPCIONES | (VIDEO 21)

- Excepciones
 - ¿Qué es una excepción?
 - ¿Cómo manejar excepciones?
- Las excepciones son errores que ocurren durante la ejecución del programa. La sintaxis del código es correcta pero durante la ejecución ha ocurrido "algo inesperado".

Partiendo de este código:



Nos encontramos con un código que no tiene errores, pero a la hora de ejecutarlos según los valores que vayamos a introducir esto puede generar un error.

Primero lo vamos a ejecutar haciendo una suma.



Ahora vamos a realizar un división entre 0, esto genera número infinitos y provocará un error.



El programa cae nos muestra la línea que ha generado el error y no termina su ejecución.

La solución es hacer una captura o control de excepción.



Si realizamos estos cambios, le estamos diciendo con try que intenten ejecutar la siguiente línea y si esta genera un error porque el usuario a introducido un valor incorrecto, salte a la línea except con su respectivo error, muestre un mensaje y continúe con su ejecución.







EXCEPCIONES II (VIDEO 22)

- Capturas de varias excepciones
- Cláusula finally

Siguiendo con el programa anterior, queremos controlar que si el usuario en lugar de introducir un numero introduce un texto veremos lo que pasa:



Observamos el siguiente mensaje. "ValueError"

Esto afecta a las siguientes líneas.



Para controla que el usuario no pone textos en lugar de número modificaremos el siguiente código:



Cuando lo ejecutemos e introduzcamos un texto el lugar de un número observaremos el siguiente mensaje:



Y preguntará de nuevo has que introduzcamos los valores correctos.

Vamos con otro ejemplo:



Hemos creado una función para dividir dos números, con lo comentado anterior aquí se pueden generar dos errores por parte del usuario, introducir texto en lugar de números o hacer una división entre 0.



Ahora si introducimos un numero incorrecto:

StepL* [python] - Sublime Text (UNREGISTERED)	_		×
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
✓ ▶ prueba_excepciones2.py × *REPL* (python) ×			▼
Introduce el primer número: uno El valor introducido es erróneo Cálculo finalizado ***Repl Closed***	ų	100 TM	
Line 6, Column 1 Tab Size	± 4	Pythor	1

Si dividimos entre 0.

StepL* [python] - Sublime Text (UNREGISTERED)	-		×
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
✓ ▶ prueba_excepciones2.py × *REPL* [python] ×			▼
Introduce el primer número: 7	000	bae.	
Introduce el segundo número: 0			
No se puede dividir entre 0!			
Cálculo finalizado			
Repl Closed			
	<u></u>		
Line 7, Column 1 Tab Size: 4		Pythor	1

Y si introducimos todos los datos correctamente.



Imagina que tienes un código que podría tener muchos errores, necesitaríamos introducir muchas excepciones.



Definimos except sin especificar el tipo de error, si introducimos un mensaje este será mas genérico y menos información dará al usuario.

Ejecutamos el proyecto:



Hemos introducido un valor incorrecto.



División por 0.



Introducidos los valores correctamente.



Con la instrucción finally le estamos diciendo que si no se comenten errores, es decir que ignora los except, ejecute lo que se encuentra en finally, de esta forma nos aseguramos de que al no haber ningún error imprima el mensaje.



Al eliminar los except el programa no mostrará los mensajes, pero al estar la función finally mostrará el mensaje "Cálculo finalizado" y después mostrará los errores.



Si borramos el finally y dejamos el try esto genera un error, ya que el try necesita un except o un finally.




EXCEPCIONES III (VIDEO 23)

- Lanzamiento de excepciones
 - o Instrucción Raise
 - Creación de excepciones propias (más adelante en el curso cuando veamos la POO)

Partiendo del siguiente ejemplo:



Si lo ejecutamos

5 *	*REPL* [python] - Sublime Text (UNREGISTERED)								_		×	
<u>F</u> ile	<u>E</u> dit	Selection	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp			
4 Þ		preuba_exce	pciones	_3.py	×	*REPL*	[python]	×				▼
Eres muy joven												
***	***Repl Closed***											
	Line 4	4, Column 1							Tab Size: 4		Python	

El problema está cuando el usuario introduce una edad negativa.



Como programador puedes generar errores al ejecutar un programa, para ello utilizaremos la función raise que le podemos asignar el tipo de error y su correspondiente mensaje.

Como verás en el ejemplo en este caso ponemos una edad negativa y hemos puesto una condición de que si la edad es menor de 0 genere el error.



Si el programa cambiamos el -15 por 15 verás el no hay error.

5	Sublime Text (UNREGISTERED)							_		×			
<u>F</u> ile	<u>E</u> dit	<u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp				
		preuba_exce	pciones	_3.py	×	*REPL*	[python]	×					▼
Eres muy joven													
***	***Repl Closed***												
	Line 4	4, Column 1								Tab Size: 4	Ļ	Python	

También podemos cambiar el tipo de error, por ejemplo ZeroDivisionError.

Si lo cambias en el código e introduces de nuevo una edad negativa, el tipo de error que generará será de este tipo.

A provocar este tipo de error hace que si hubiere más líneas de programación estas no se ejecutarían.

La fusión raise no admite nombre de error propios por ejemplo MiPropioError, ya que antes habría que definirla.

En el siguiente proyecto vamos a calcular la raíz cuadrada de un número, pero si este número es negativo nos tiene que dar un error.



Vamos a ejecutar el programa e introduciremos un número positivo:



El programa funciona y el flujo llega hasta el final del programa.

Ahora vamos a introducir un número negativo.



Ahora se genera el error que hemos definido, con el mensaje que hemos introducido y el programa no llega al final de flujo.

La línea 10 es la que genera el error porque no se puede hacer la raíz cuadrado de un número negativo y en la línea 5 es la que provocamos el error con un condicionante y un determinado mensaje.



En este programa si introducimos un número positivo.



Si introducimos un número negativo.



El programa se ejecuta hasta el final.





POO I (PROGRAMACIÓN ORIENTADA A OBJETOS) (VIDEO 24)

• ¿Qué es la POO?

• Ventajas de este paradigma de la programación.



- Programación Orientada a procedimientos:
 - Algunos ejemplos de lenguajes: Fortran, Cobol, Basic, etc.
- Desventajas
 - Unidades de código muy grandes en aplicaciones complejas.
 - En aplicaciones complejas el código resultaba difícil de descifrar.
 - Poco reutilizable.
 - Si existe fallo en alguna línea del código, es muy probable que el programa caiga.
 - Aparición frecuente del código espagueti.
 - Difícil de depurar por otros programadores en caso de necesidad o error.

```
10 LET X=0

30 LET Y=X

40 LET J=1

45 LET I=X

50 GO SUB 1000

55 LET J=AB5 (J-1)

50 LET I*INT (6*RND+1)

75 IF X=31 OR Y=31 THEN PRINT

"BRAUO ";U

60 PRINT "JUGADOR ";J+1;" :";

I: BEEP 0.2,10+10+J

85 PAUSE 4E4

90 GO SUB 1000

100 LET I=INT (4*RND)

110 GO SUB 1200

120 GO TO 50

150 RETURN

200 LET I=-I

210 RETURN
```

Programación Orientada a objetos

- ¿En qué consiste?
 - Trasladar la naturaleza de los objetos de la vida real al código de programación.
- ¿Cuál es la naturaleza de un objeto de la vida real?
 - Los objetos tienen un estado, un comportamiento (¿Qué puede hacer?), y una propiedades.
- Pongamos un ejemplo: El objeto coche.
 - ¿Cuál es el estado de un coche? Un coche puede estar parado, circulando, aparcado, etc.
 - ¿Qué propiedades tiene un coche? Un coche tiene un color, un peso, un tamaño, etc.
 - ¿Qué comportamiento tiene un coche? Un coche puede arrancar, frenar, acelerar, girar, etc.

Objeto ¿Qué es?

- Objeto:
 - Tiene propiedades (atributos):
 - Color
 - Peso
 - Alto
 - Largo
 - Tiene un comportamiento (¿Qué es capaz de hacer?
 - Arrancar
 - Frenar
 - Girar
 - Acelerar



- Programación Orientada a objetos:
 - Algunos ejemplos de lenguajes, C++, Java, Visual.NET, etc.
- Ventajas:
 - Programas divididos en "trozos", "partes", "módulos" o "clases". Modulación.
 - Muy reutilizable. Herencia.
 - Si existe fallo en alguna línea de código el programa continuará con su funcionamiento. Tratamiento de Excepciones.
 - Encapsulamiento.

Vocabulario de la POO

- Clase
- Objeto
- Ejemplar de clase. Instancia de clase. Ejemplarizar una clase. Instanciar una clase.
- Modularización.
- Encapsulamiento / encapsulación.
- Herencia
- Polimorfismo





POO II (VIDEO 25)

• Conceptos y términos de la POO

Vocabulario de la POO

- Clase
- Objeto
- Ejemplar de clase. Instancia de clase. Ejemplarizar una clase. Instanciar una clase.
- Modularización.
- Encapsulamiento / encapsulalción.
- Herencia
- Polimorfismo

Clase: Modelo donde se redactan las características comunes de un grupo de objetos.



En este caso pueden ser el chasis y las ruedas, es frecuente que varios fabricantes utilicen el mismo chasis y las mismas ruedas.

La clase podría ser el chasis y las ruedas.

Algo común de los coches es que tienen 4 ruedas.

INSTANCIA

• Ejemplar perteneciente a una clase.



Estos coches al ser de dos fabricantes distintos comparten el chasis y las ruedas.

Objeto es sinónimo de instancia y de ejemplar.

MODULARIZACIÓN



En un equipo de música tenemos los módulos, que por sí solos funcionan.

Tu podrías llevarte un módulo del equipo de música y conectarlo con otro equipo de música que le faltara este módulo.

Encapsulación: la Clase 1 no sabe nada del funcionamiento de la Clase 2, de este modo no es accesible de una clase a otra.

Métodos de acceso: consigues conectar unas clases a otras para que funcionen como una unidad o equipo.

- Objeto:
 - Accediendo a propiedades de objeto desde código (pseudocódigo):
 - miCoche.color="rojo"
 - miCoche.peso=1500
 - miCoche.ancho=2000
 - miCoche.alto=900
 - Accediendo a comportamiento de objeto desde código (pseudocódigo):
 - miCoche.arranca()
 - miCoche.frena()
 - miCoche.girta()
 - miCopche.acelera()

nomenclatura del punto: miCoche.color="rojo"





POO III (VIDEO 26)

• Traslado a código los conceptos de la POO vistos anteriormente.

Vamos a construir en código lo que será una clase.

```
D:\Curso Python\poo_1.py - Sublime Text (UNREGISTERED)
                                                                                  \times
File Edit Selection Find View Goto Tools Project Preferences Help
4
       poo_1.py
       class Coche(): # El nombre la clase primera en mayúsculas
           largoChasis=250 #Propiedades
           anchoChasis=120
          ruedas=4
           enmarcha=False
          def arrancar(self): # Metodo de la clase
               self.enmarcha=True
           def estado(self):
               if(self.enmarcha):
                    return "El coche está en marcha"
                   return "El coche está parado"
      miCoche=Coche() # Acabamos de crear nuestro objeto de tipo Coche
      print("El largo del coche es: ",miCoche.largoChasis)
      print("El coche tiene ", miCoche.ruedas, " ruedas")
      miCoche.arrancar()
      print(miCoche.estado())
Line 24, Column 24
                                                                    Tab Size: 4
                                                                                  Python
```

Este será el resultado:

REPL [python] - Sublime Text (UNREGISTERED)								
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp								
♦ poo_1.py × *REPL* [python] ×			▼					
El largo del coche es: 250 El coche tiene 4 ruedas El coche está en marcha								
Repl Closed								
Line 6, Column 1 Tab	Size: 4	Python						

Nuestra clase Coche() tiene 4 propiedades y dos comportamientos, arranca y estado.





POO IV (VIDEO 27)

• Traslado a código los conceptos de la POO vistos anteriormente.



Este será el resultado:





Eliminamos las líneas que nos sobran, este será el resultado:



Constructor: es un método que le da el estado inicial a los objetos.



El resultado no cambia.



Ahora queremos cambiar el numero de ruedas de miCoche2



Si imprimimos el segundo coche nos dirá que tiene 5 ruedas, para impedir que se puedan realizar estos tipos de cambios vamos a ver encapsulación para proteger este valor y no se pueda modificar desde fuera de la clase.



Una vez encapsulado ___ruedas si intentamos cambiar el valor, este no cambiará, en la línea 38 le decimos que miCoche2 tiene 5 ruedas, pero cuando consultamos el estado de miCoche2 este será el resultado:





Ponemos dos guiones bajos en cada una de las variables de la clase y si queremos modificar la propiedad ____enmarcha lo haremos dentro de la clase y desde fuera no se podrá modificar.





POO V (VIDEO 28)

- Encapsulación de método
 - o ¿Qué es?
 - ¿Por qué utilizar la encapsulación?

class Coche():

def __init__(self):

self.__largoChasis=250 self.__anchoChasis=120 self.__ruedas=4 self.__enmarcha=False

def arrancar(self,arrancamos):

self.___enmarcha=arrancamos

if(self.__enmarcha):

chequeo=self.__chequeo_interno()

if(self.__enmarcha and chequeo):

return "El coche está en marcha"

elif(self.__enmarcha and chequeo==False):

return("Algo ha ido mal en el chequeo. No podemos arrancar.")

else:

return "El coche está parado"

def estado(self):

print("El coche tiene ", self.__ruedas, " ruedas, un ancho de ", self.__anchoChasis, " y un largo de ", self.__largoChasis)

```
def __chequeo_interno(self):
```

print("Realizando chequeo interno.")

self.gasolina="ok"

self.aceite="ok"

self.puerta="cerradas"

if(self.gasolina=="ok" and self.aceite=="ok" and self.puerta=="cerradas"): return True

leet

else:

return False

```
miCoche=Coche()
print(miCoche.arrancar(True))
miCoche.estado()
print("------A continuación creamos el segundo obeto------")
miCoche2=Coche()
print(miCoche2.arrancar(False))
miCoche2.estado()
```

Este será el resultado:



En este ejemplo el coche antes de ponerlo en marcha hace un chequeo interno.





POO VI. HERENCIA (VIDEO 29)

- La herencia
 - o ¿Qué es?
 - ¿Para qué sirve?
 - o ¿Cómo hacer que las clases hereden?





La clase1 clase padre es superclase. La clase 2 es clase subclase y a su vez es superclase de la clase 3, clase 4 y clase 5.

¿Para qué sirve la herencia?

• Para reutilizar código en caso de crear objetos similares.







Vamos a crear una clase principal que heredarán otras clases.



La clase Moto() hereda las clases y métodos de la clase Vehiculos().







POO VII HERENCIA II (VIDEO 30)

- Herencia:
 - Sobre escritura de métodos
 - Herencia simple y múltiple



Este será el resultado:

REPL [python] - Sublime Text (UNREGISTERED)						
<u>File E</u> dit <u>S</u> election Find <u>V</u> iew <u>G</u> oto <u>T</u> ools <u>P</u> roject Prefere <u>n</u> ces <u>H</u> elp						
✓ Herencia.py × *REPL* [python] ×			▼			
Marca: Honda Modelo: CBR En Marcha: False Acelerando: False Frenando: False Voy haciendo el caballito ◀		<u>.</u>				
Repl Closed						
Line 8, Column 18 Tab Size: 4		Python				

Ahora la clase Moto tiene todas las propiedades y métodos de Vehiculo() además de sus propias propiedades y métodos.



Si no llamamos al método caballito al ejecutar observaremos que dicho método no se imprime.



Esto es lo que se conoce como sobreescritura de métodos.



Este será el resultado:

REPL [python] - Sublime Text (UNREGISTERED)	_		×
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
Herencia.py • *REPL* [python] ×			▼
Marca: Honda		¥-	
Modelo: CBR			
En Marcha: False			
Acelerando: False			
Frenando: False			
Voy haciendo el caballito			
Marca: Renault			
Modelo: Kango			
En Marcha: True			
Acelerando: False			
Frenando: False			
La furgoneta está cargada			
Repl Closed			
Line 15, Column 1 Tab Size	: 4	Pythor	ı

Aquí tenemos los parámetros y el estado de la furgoneta.

Herencias múltiples:

40	<pre>class VElectricos():</pre>
41	<pre>definit(self):</pre>
42	<i>self</i> .autonomia=100
43	<pre>def cargarEnergia(self):</pre>
44	<i>self</i> .cargando=True

Hemos creado una nueva clase llamada VElectricos().



Pones pass porque está vacía pero le asignamos dos clases, a esto le llamamos herencia múltiple.

```
58 miBice=BicicletaElectrica("Orbea", "HC1030")
```

Para que funcione correctamente le hemos de poner los parámetros en orden de cuando definimos la clase, primero los parámetros de Vehiculo y luego los de VElectricos, este último está vacío.





POO VIII. HERENCIAS III (VIDEO 31)

- Herencia:
 - Super()
 - Isinstance()

La función super() va a llamar al método de la clase padre.



Este será el resultado:



Principio de sustitución: "es siempre un/a" en nuestro ejemplo un empleado es siempre una persona.

isinstance(): nos va a decir si un objeto es una instancia de una clase determinada.



Este será el resultado:



No está diciendo que Manuel pertenece a la clase Empleado.

27 print(isinstance(Manuel, Persona))

Si queremos saber si Manuel también pertenece a la clase Persona.

Este será el resultado:

REPL [python] - Sublime Text (UNREGISTERED)	_		\times
File Edit Selection Find View Goto Tools Project Preferences	Help		
♦ Herencia2.py ● *REPL* [python] ×			▼
Nombe: Manuel Edad: 55 Residencia: Colombia Salario: 1500 Antigüedad: 15 True ***Repl Closed***		Print Print	
Line 6, Column 1 Tab Size: 4		Python	Ľ

También pertenece a la clase Persona.

Con referencia al ejemplo anterior vamos a aplicarlo lo que hemos comentado.

```
class Vehiculos():
           self.marca=marca
           self.modelo=modelo
           self.enmarcha=False
           self.acelera=False
self.frena=False
           self.enmarcha=True
       def acelerar(self):
           self.acelera=True
       def frenar(self):
           self.frena=True
       def estado(self):
           class Furgoneta(Vehiculos):

def carga(self, cargar):

self.cargado=cargar
        if(self.cargado):
           return "La furgoneta está cargada"
           return "La furgoneta no está cargada"
class Moto(Vehiculos):
    hcaballito="
    def caballito(self):
        self.hcaballito="Voy haciendo el caballito"
    def estado(self):
       class VElectricos(Vehiculos):
   def __init__(self, marca, modelo):
    super(). __init__(marca, modelo)
    self.autonomia=100
    def cargarEnergia(self):
       self.cargando=True
```



Hemos realizado los cambios el ejemplo anterior.





POO IX. POLIMORFISMO (VIDEO 32)

Un objeto puede cambiar de forma dependiendo del contexto que se utiliza, al cambiar de forma también cambia de comportamiento.



Hemos creado tres clase, en cada una de ellas un método con el mismo nombre.



Si definimos un primer objeto con la clase Moto() y un segundo objeto con la clase Coche(), al ejecutar el método cada objeto ejecutará el método a la clase que pertenece.



Hemos creado tres tipos de clases en cada una hay un método con el mismo nombre.

Para hacer que el método desplazamientoVehiculo elija el método de la clase que deseemos esta se la pasaremos con parámetros.

En este ejemplo creamos un objeto miVehiculo de tipo Camion, cuando llamemos al método desplazamientoVehiculo con el parámetro miVehiculo que a su vez es de tipo camión ejecutará el método que hay en dicha clase.



Si cambiamos miVehiculo=Moto()


Este será el resultado:







Para hacer un repaso de todo lo aprendido hasta ahora te propongo que copies el siguiente proyecto:

class Persona:

def __init__(self, nombre,apellidos,edad):
 self.Nombre=nombre
 self.Apellidos=apellidos
 self.Edad=edad
def MostarPersona(self):
 print("Nmbre: ",self.Nombre)
 print("Apellidos: ",self.Apellidos)
 print("Edad: ",self.Edad)

class Empleado:

def __init__(self, numero, sueldo, departamento, categoria): self.Numero=numero self.Sueldo=sueldo self.Departamento=departamento self.Categoria=categoria

def AñadirPersona(self,persona): self.Persona=persona

 print("Numero de empleado: ", str(self.Numero))
self.Persona.MostarPersona()
print("------")

def obtenerNumero(self): return self.Numero

class Empresa:

def __init__(self): self.ListaEmpleados=[]

def NumeroEmpleados(self): return len(self.ListaEmpleados)

def AñadirEmpleado(self,empleado): self.ListaEmpleados=self.ListaEmpleados+[empleado]

def MostrarEmpresa(self):

def BorrarEmpleado(self,numero):

encontrado=False posicionaBorrar=-1 for item in self.ListaEmpleados: posicionaBorrar +=1 if item.obtenerNumero()==numero: encontrado=True break

if encontrado:

del self.ListaEmpleados[posicionaBorrar]

print("Empleado borrado correctamente.")

else:

print("iEmpleado no encontrado!")

def MostrarMenu():

print(""" Menú

1) Añadir empleado a empresa

2) Mostrar empleados de empresa

3) Borrar empleado

4) ¿Número de empleados?

5) Salir""")

def AñadirEmpleadoEmpresa(empresa):

numero=input("Introuzca el número de empleado:") personanombre=input("Introduzca el nombre del empleado: ") personaapellidos=input("Introduzca los apellidos del empleado: ") personaedad=input("Introduzca la edad del empleado: ") sueldo=input("Introduzca el sueldo del empleado: ") departamento=input("A que departameto está asignado el empleado: ") categoria=input("Que categoría profesional tiene el empleado: ")

persona=Persona(personanombre,personaapellidos,personaedad) empleado=Empleado(numero, sueldo, departamento, categoria) empresa.AñadirEmpleado(empleado) empleado.AñadirPersona(persona) return empresa

def MostrarEmpresa(empresa): empresa.MostrarEmpresa()

def BorrarEmpleado(Empresa): numero=input("Introduce el número de trabajador a borrar: ") Empresa.BorrarEmpleado(numero)

```
def NumeroEmpleados(empresa):
```

print("El número de empleados en la empresa es: ",empresa.NumeroEmpleados())

```
fin=False
empresa=Empresa()
```

```
while not fin:
```

```
MostrarMenu()

opcion=int(input("Seleccione una opcion: "))

if(opcion==1):

empresa=AñadirEmpleadoEmpresa(empresa)

elif(opcion==2):

MostrarEmpresa(empresa)

elif(opcion==3):

BorrarEmpleado(empresa)

elif(opcion==4):

NumeroEmpleados(empresa)

elif(opcion==5):

fin=True
```

print("¡Fin del proyecto!")

Después comprueba el resultado.

MÉTODOS DE CADENA (VIDEO 33)

- Uso de métodos de cadenas.
- Objetos String



upper(): Convierte en mayúsculas las letras de un string.

lower(): Convierte en minúsculas las letras de un string.

capitalize(): En un string poner la primera letra en mayúsculas.

count(): Cuenta cuantas veces aparece una letra o una cadena en un texto.

find(): Representa la posición donde aparece un carácter o un grupo de caracteres.

isdigit(): devuelve un voleano, True o False dice si el valor introducido es un dígito un valor numérico o no.

isalum(): Comprobar si son alfanuméricos.

isalpha(): Comprobar si solo hay letras, los espacios no cuentan.

split(): Separa palabras por palabra utilizando espacio.

strip(): Borras espacios sobrante al principio y al final.

replace(): Cambia una letra o palabra por otra.

rfind(): Representa la posición donde aparece el carácter o un grupo de caracteres desde atrás.

Mas información en la siguiente url: <u>http://pyspanishdoc.sourceforge.net/</u>

108) Curso de Python. Métodos × Socumentación de Python	× + – – ×
 ← → C (① No es seguro pyspanishdoc.sourceforge.net ↔ Aplicaciones 	⊡ ☆ 🧔 🛊 🗊 🥠 :
Proyecto de Documentación de Python en c SOURCEFORGE Incluye CVS, versiones en PDF, zips y tarballs con voluntarios.	astellano en Sourceforge la documentación en HTML, noticias Y siempre se buscan
Documentad	ción de Python modelis
 <u>Guía de aprendizaje</u> (empezar por aquí) <u>Índice global de módulos</u> (acceso rápido a toda la documentación) <u>Referencia de bibliotecas</u> (mantén esto bajo tu almohada) <u>Referencia de módulos para Macinto</u> (y esto si usas un Macintosh) <u>Instalación de módulos de Python</u> (para administradores) 	 ión de Python <u>Referencia del lenguaje</u> (para leguleyos del lenguaje) <u>Extensión y empotrado</u> (guía para programadores de C/C++) <u>API de Python/C</u> (referencia para programadores de C/C++) <u>bocumentar Python</u> (cómo hacer documentación) <u>Distribución de módulos de Python</u> (para desarrolladores y empaquetadores)
 <u>Central de documentación de Pythor</u> (para todo el mundo) 	• <u>Guías Cómo de Python</u> (temas especiales)
Ver <u>Acerca de la documentación de Python</u> si se desea sugerir can	nbios.

Vamos a ver unos ejemplos:



Este será el resultado:

REPL [python] - Sublime Text (UNREGISTERED)	-		×
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
♦ Practica_Strings.py × *REPL* [python] ×			▼
Introduce un nombre de usuario: pere MANEL el nombre es: pere MANEL el nombre en mayúsculas es: PERE MANEL el nombre en minúsculas es: pere manel el nombre con la primera letra en mayuúsculas: Pere manel ***Repl Closed***		netor.	
Line 8, Column 1 Tab Size: 4		Python	

Para comprobar si el valor introducido en un digito o es un texto:



Vamos a ejecutar introduciendo un texto.

1	REPL*	[python] - S	Sublim	e Text (l	JNREGIS	STERED)					—		X
<u>F</u> ile	<u>E</u> dit	<u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp				
•	Γ	Practica_Strir	ngs.py		*RI	EPL* (pyt	hon]	×					▼
Intr Fals	oduo e	ce la ed	lad: 1	tres									
***6	lep1	Closed*											
	Line 5	5, Column 1								Tab Size: 4		Python	

Ahora vamos a introducir un número.

5	*REPL*	[python] -	Sublim	e Text (l	JNREGI	STERED)					_		Х
<u>F</u> ile	<u>E</u> dit	Selection	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp				
••	• /	Practica_Stri	ngs.py		*R	EPL* (pyt	thon]	×					▼
Int Tru	rodu e	ce la ed	lad:	3									
***	Repl	Closed*											
	Line	5, Column 1								Tab Size	e: 4	Python	

En el siguiente ejemplo vamos a comprobar si introducimos un valor numérico, sino es así irá preguntando hasta que introduzcas un valor numérico.



Si ejecutamos este será el resultado:



Ahora vamos a realizar la siguiente práctica.



Ejercicios Python

Ejercicio 1:

 Crea un programa que pida introducir una dirección de email por teclado. El programa debe imprimir en consola si la dirección de email es correcta o no en función de si esta tiene el símbolo '@'. Si tiene una '@' la dirección será correcta. Si tiene más de una o ninguna '@' la dirección será errónea. Si la '@' está al comienzo de la dirección de email o al final, la dirección también será errónea,

(Soluciones en la página siguiente)

SOLUCIONES

Los ejercicios se deben hacer utilizando exclusivamente los conceptos vistos hasta el vídeo 33 del curso. Recuerda que "En programación siempre hay varios caminos para llegar a Roma". Las soluciones que se proponen a continuación son válidas, pero no las únicas posibles. Tal vez tus programas funcionen con códigos diferentes.

Solución Ejercicio 1:







MÓDULOS (VIDEO 34)

- Qué son los módulos, utilidad y cración.
- ¿Qué son?
 - Un archivo con extensión .py .pyc (Python compilado) o archivo escrito en C para CPython, que posee su propio espacio de nombre y que puede contener variables, funciones, clases e incluso otros módulos.
- ¿Para qué sirven?
 - o Para organizar y reutilizar el código (modularización y reutilización)

Modularización: En lugar de hacer un proyecto con un código de muchas líneas, hacer pequeños códigos que luego unificaremos en el programa.

- ¿Cómo se crear un módulo?
 - Tan sencillo como crear un archivo con extensión .py (o pyc o archivo C) y guardarlo donde nos interese.

Para ver con más claridad vamos a crear una carpeta llamada "modulos" donde guardo los proyectos de Python.



Vamos a crear un nuevo proyecto que guardaremos en la carpeta modulos con el nombre de funciones_matematicas.py con el siguiente código:





Ahora en la misma carpeta vamos a crear un nuevo proyecto llamado uso_funciones.py.

Al ejecutar este proyecto obtendremos el siguiente resultado:



¿Cómo podemos evitar de tener que escribir el módulo?



Este será el resultado:

5	*REPL*	[python] -	Sublim	e Text (UNREGI	STERED)				_		×
<u>F</u> ile	<u>E</u> dit	<u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp			
.	• /	funciones_m		cas.py	×		unciones.p	y x	*REPL* [python]	×		▼
E1	resu	ltado de	la :	suma	es:	12						
***	Repl	Closed*										
	Line 4	4, Column 1							Tab Size: 4	4	Python	

Pero para que puedan funcionar el resto de funciones vamos a realizar lo siguiente:



Este será el resultado:



La diferencia de hacerlo de una forma u otra en la optimización de espacio en memoria, si vas a utilizar pocos recursos lo correcto es usar el primer ejemplo, pero si vas a utilizar gran parte de ello el segundo ejemplo es mejor.

Ahora vamos a realizar un ejemplo con clases y herencias.

Creamos un nuevo proyecto llamado modulo_vehiculos.py que agregaremos el siguiente código:



Recuerda de guardarlo en la carpeta de "modulos" que hicimos para este capítulo.

Ahora realizamos un proyecto nuevo llamado "uso_vehiculos.py"



Ejecutaremos para ver el resultado:



Si cambiamos la ubicación del modulo cuando ejecutemos de nuevo el interprete de Python no encuentra el archivo.

Python por defecto va a buscar el módulo en la misma carpeta donde se encuentra el proyecto que lo llama.



PAQUETES (VIDEO 35)

- Que son los paquetes, utilidad y creación
- ¿Qué son?
 - o Directorios donde se almacenarán módulos relacionados entre sí
- ¿Para qué sirven?
 - Para organizar y reutilizar los módulos
- ¿Cómo se crea un paquete?
 - Tan sencillo como crear una carpeta con un archivo __init__.py

Lo primero que vamos a realizar es crear una carpeta donde guardamos nuestros proyectos de Python llamada cálculos.

📙 🛛 🛃 🚽 🗍 🖘				- 0	×
Archivo Inicio Compartir	Vista				~ ?
Anclar al Acceso rápido	Mover a ▼ ★ Eliminar ▼ Copiar a ▼ ➡ Cambiar nombre	a a Propiedades	Seleccionar todo		
Portapapeles	Organizar Nu	ievo Abrir	Seleccionar		
← → · ↑ 📕 > Este equi	po > Disco local (D:) > Curso Python >		< ∿ €	Buscar en (Curso P
Escritorio ^ No	ombre	Fecha de modificación	Тіро Та	maño	^
📰 Imágenes	modulos	10/08/2020 8:07	Carpeta de archivos		
👌 Música 🛛	calculos	10/08/2020 9:46	Carpeta de archivos		
🗊 Objetos 3D 🛛 🛛	Biblioteca	06/08/2020 15:48	Python File	3 KB	
📕 Vídeos 🛛	clase1	06/08/2020 11:05	Python File	1 KB	
Windows (C:)	Control_correo	08/08/2020 15:18	Python File	1 KB	
Discolocal (D)	controlPersonal	08/08/2020 7:08	Python File	4 KB	
	Curso de Python basados en los vídeos de	03/08/2020 17:33	Archivo PDF	2.708 KB	
💣 Red 📃 🙍	Curso de Python basados en los vídeos de	08/08/2020 8:00	Archivo PDF	6.537 KB	
39 elementos 1 elemento selec	cionado	02/00/2020 7 50	B.0. 51	4.120	

En ella hemos de guardar un archivo vacío con el nombre de __init__.py

🗧 Guardar como									X
← → * ↑	Este equipo → Disco local (D:) > Curso Python > (calculos			ٽ ~	P Buscar en	alculos	
Organizar 👻 Nueva	a carpeta							== -	
Documentos	Nombre	^	Fecha de modificación	Тіро	Tamaño				
Escritorio			Ningún element	o coincide con el criterio	de búsqueda.				
Imágenes			2						
Música									
Vídeos									
Windows (C:)									
Disco local (D:)	~								
Nombre:i	nitpy								~
<u>T</u> ipo: All	Files (*.*)								\sim
 Ocultar carpetas 							<u>G</u> uardar	Cancelar	
D:\Curso	Python\calculos	initpy - Sı	ublime Text (UN	REGISTERED)			- C	ı x	
<u>F</u> ile <u>E</u> dit <u>S</u> e	election F <u>i</u> nd	<u>V</u> iew <u>G</u> oto	<u>T</u> ools <u>P</u> rojec	t Prefere <u>n</u> ce	es <u>H</u> elp				
. ↓ ir	nit ov	×							
	··· <u>_</u> ·P)								
	olumn 1				T:	ab Size: 4	D	thon	
Line i, e	oronnin i				16	00 012014	(1	CHOIL .	

Si miramos el contenido de la carpeta:

📙 🛛 🚽 🗖 🖛 🗸 🖌 calculos					_		
Archivo Inicio Compartir	Vista					~ ?	
Anclar al Acceso rápido	Mover a • K Eliminar •	Nueva carpeta	des 🖉 👻 🔡 Sele	ccionar todo seleccionar nada rtir selección			
Portapapeles	Organizar	Nuevo Al	orir S	eleccionar			
← → · ↑ 🕞 > Este equipo > Disco local (D:) > Curso Python > calculos v 👌 🔎 Buscar en calculos							
Escritorio ^ No	ombre	Fecha de modifica	ción Tipo	Tam	año		
📰 Imágenes	initpy	10/08/2020 9:48	Python File		0 KB		
🁌 Música 🗸 🗸							
1 elemento 1 elemento seleccio	onado 0 bytes						

Vamos a crear un nuevo proyecto llamado cálculos_generales.py



Esto es lo que hay en la carpeta "calculos".

📙 🛃 📕 🖛 ca	alculos							-		×
Archivo Inicio	Compartir	Vista								~ ?
Anclar al Acceso rápido	r Pegar	🖕 Mover a ▾ 📑 Copiar a ▾	¥ Eliminar ▼ ■ Cambiar nombre	Nueva carpeta	Propiedades	Selecci Selecci No se Selecci	cionar to lecciona ir selecci	do r nada ión		
Portapape	les	0	rganizar	Nuevo	Abrir	Sel	eccionar			
\leftrightarrow \rightarrow \checkmark	→ Este equi	po → Disco loca	II (D:) → Curso Python	> calculos		~	ē	,	car en ca	lculos
Escritorio	^ No	ombre	^	Fecha de	modificación	Тіро		Tamaño		
📰 Imágenes		_initpy		10/08/20	20 9:48	Python File		0	KB	
🁌 Música		calculos_genera	ales.py	10/08/20	20 9:58	Python File		1	KB	
🧊 Objetos 3D	~									
2 elementos 1 el	emento selecci	onado 0 bytes								==

Ahora vamos a crear un nuevo proyecto que irá a donde guardamos los proyectos, es decir fuera de la carpeta calculos con el nombre "Uso_modulos.py".

→ * ↑ - Ste equipo → Disco local (D:) → Curso Python →	ڻ ~	🔎 Buscar en C	Curso Python
anizar 🔻 Nueva carpeta			
Documentos ^ Nombre	Fecha de modificación	Тіро	Tamaño
Escritorio 🛛 🖗 Nacimiento.py	09/08/2020 18:05	Python File	
🖬 Imágenes 🔋 📴 HerenciaMultiple.py	09/08/2020 10:47	Python File	
Música 📴 Television.py	09/08/2020 7:22	Python File	
👕 Objetos 3D 🛛 🕞 gelatina.py	09/08/2020 7:04	Python File	
Vídeos 🦻 Preubas.py	08/08/2020 15:56	Python File	
Windows (C)	08/08/2020 15:18	Python File	
Disco local (D:) V C	08/08/2020 12:56	Python File	
No <u>m</u> bre: Uso_modulos.py			
Tipo: All Files (*.*)			

Vamos a escribir los siguiente:

D:\Curso Python\Uso_modulos.py - Sublime Text (UNREGISTERED)	_		×
<u>File Edit Selection Find View Goto Tools Project Preferences Help</u>			
Initpy x calculos_generales.py x Uso_modulos.py x *REPL* [p]		×	▼
<pre>1 from calculos.calculos_generales import dividir 2 3 dividir(4,6)</pre>			
Line 3, Column 13 Tab Siz	ze: 4	Python	

Este será el resultado:

REPL [python] - Sublime Text (UNREGISTERED)	_		×
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
▲▶initpy x Calculos_generales.py x Uso_modulos.py x *REPL* [p	ython]	×	▼
El resultado de la división es: 0.666666666666666666666666666666666666			
Repl Closed			
Line 4, Column 1 Tab Siz	ze: 4	Python	

Ahora vamos a importar todos los módulos:

D:\Curso Python\Uso_modulos.py - Sublime Text (UNREGISTERED)	_		\times
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
initpy x calculos_generales.py x Uso_modulos.py x *REPL	L* [python]	×	▼
<pre>1 from calculos.calculos_generales import * 2 3 sumar(5,11) 4 restar(12,8) 5 multiplicar(9,7) 6 dividir(4,6) 7 potencia(5,3) 8 redondear(4.8)</pre>			
Line 8, Column 15 Ta	b Size: 4	Python	

Este será el resultado:



La carpeta _pycache__ se genera automáticamente, en ella podremos observar:

<mark> </mark> 🕑 📙 - _	pycache								_		\times
Archivo Inicio	Compartir	Vista									^ ?
Anclar al Acceso rápido	r Pegar	🛃 Mover a 👻	X Eliminar ▼ ■ Cambiar nombre	Nueva carpeta	Propiedades	Select No select No select No select No select	cionar to elecciona tir selecc	odo ir nada ión			
Portapape	les	0	rganizar	Nuevo	Abrir	Se	lecciona	r			
$\leftarrow \rightarrow \cdot \cdot \uparrow$	→ Este equi	po → Disco loca	al (D:) > Curso Python	> calculos >	_pycache_	~	Ō	P	Buscar	enpy	/cac
👌 Música	^ No	ombre	^	Fecha de	e modificación	Тіро		Tamañ	io		
🧊 Objetos 3D			n-38.pyc	10/08/20	020 10:05	Compiled Py	thon		1 KB		
🐺 Vídeos		calculos_gener	ales.cpython-38.pyc	10/08/20	20 10:10	Compiled Py	thon		2 KB		
🏪 Windows (C	:) 🗸										
2 elementos											:==

Para que la pueda interpretar como un paquete de Python.

Ahora dentro de la carpeta cálculos vamos a crear una nueva carpeta llamada basicos

📙 🖂 🥃 🗧 🗸				– 🗆 ×			
Archivo Inicio Compartir	Vista			^ ?			
Anclar al Acceso rápido	Mover a • KEliminar •	Nueva carpeta	Seleccionar todo	ia			
Portapapeles	Organizar	Nuevo Abrir	Seleccionar				
\leftarrow \rightarrow \checkmark \uparrow \Box > Este equipo > Disco local (D:) > Curso Python > calculos > \checkmark \circlearrowright P Busca							
📰 Imágenes \land No	ombre	Fecha de modificación	Tipo Tar	naño			
👌 Música	pycache	10/08/2020 10:10	Carpeta de archivos				
🗊 Objetos 3D	basicos	10/08/2020 10:17	Carpeta de archivos				
Vídeos 🛛	initpy	10/08/2020 9:48	Python File	0 KB			
🏪 Windows (C:)	calculos_generales.py	10/08/2020 10:10	Python File	1 KB			
Disco local (D:)							
💣 Red 🗸							
4 elementos 1 elemento selecc	4 elementos 🔰 1 elemento seleccionado 🔤 📰 🖿						

Para que este funcione habrá que copiar en ella el archivo __init__.py

Crearemos una carpeta redondeo_potencia y también le copiaremos el archivo.

Ar F vo Inicio Compartir	Vista			- □ ×	
Anclar al Acceso rápido	Mover a V Eliminar V Copiar a Cambiar nombre	Nueva carpeta	Seleccionar tod	lo nada in	
Portapapeles	Organizar	Nuevo Abrir	Seleccionar		
\leftarrow \rightarrow \checkmark \uparrow \square > Este equipo > Disco local (D:) > Curso Python > calculos > \checkmark \circlearrowright \bigcirc Buscar en calculos >					
🔚 Imágenes \land No	ombre	Fecha de modificación	Тіро	Tamaño	
👌 Música	pycache	10/08/2020 10:10	Carpeta de archivos		
🗊 Objetos 3D	basicos	10/08/2020 10:20	Carpeta de archivos		
Vídeos	redondeo_potencia	10/08/2020 10:20	Carpeta de archivos		
Windows (C:)	🖡initpy	10/08/2020 9:48	Python File	0 KB	
Disco local (D:)	calculos_generales.py	10/08/2020 10:10	Python File	1 KB	
🔿 Red 🗸 🗸					
5 elementos 1 elemento selecc	ionado				

Vamos a crear un módulo en básicos llamado operaciones_basicas.py

Este será el texto que irá:



Vamos a crear un nuevo proyecto en la carpeta redondeo_potencia con el nombre redondeaYpotencia.py

Este será el código:



Esta es la estructura que tenemos:

~	calculos
	pycache
	basicos
	📙 redondeo_potencia
Dentro de básicos hay:	
[initpy
[🏓 operaciones_basicas.py
Y dentro de redondeo_potencia ha	у:

😼initpy	
redondeoYpotencia.py	

Ahora vamos a modificar un proyecto llamado Uso_modulos.py que se encuentre en la carpeta Python, es decir fuera de la carpeta cálculos.



Este será el resutado:







PAQUETES DISTRIBUIBLES (VIDEO 36)

• ¿Cómo crear paquetes distribuibles y reutilizables



- 1.- Crear paquete
- 2.-. Instalar paquete

📕 untitled - Sublime Text (UNRE	GISTERED)		_	
File Edit Selection Find Vie	w Goto Tools Project Prefe	rences Help		
 	Side Bar	>	Hide Side Bar	Ctrl+K, Ctrl+B
1	Hide Minimap		Show Open Files	
	Hide Tabs Hide Status Bar Hide Menu			
	Show Console	Ctrl+		
	Enter Full Screen Enter Distraction Free Mode	F11 Shift+F11		
	Layout Groups	>		
	Focus Group Move File to Group	>		
	Syntax Indentation	>		
Line 1, Column 1	Word Wrap		Tab Size: 4	Plain Text
	Word Wrap Column Ruler	>		
	Spell Check	F6		
	Next Misspelling	Ctrl+F6		
	Prev Misspelling	Ctrl+Shift+F6		
	Dictionary	>		

Vamos a ejecutar Sublime Text del menú View seleccionaremos Side Bar y de este Show Open Files.



Hemos activado a la izquierda como un navegador de archivos.

Vamos a arrastrar la carpeta del curso en el apartado OPEN FILES.



Vamos a crear un archivo setup.py en la carpeta Curso Python.

5	untitle	ed (Curso Pyt	hon) -	Sublim	e Text (U	INREGIS	STERED)					_		×
<u>F</u> ile	<u>E</u> dit	<u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp					
0	PEN F	ILES			↓ ▶ 1		-	-		-				•
FC	DLDE	RS		- 11										
~	►	urso Pyth calculos	Nev	w File										
	>	modulos						-						
	/*	clace1 p	Nev	w Folder	r									
	/*	Control	Del	ete Fold	er									
	/*	controlP	Fine	d in Fold	der									
	101 011	Curso de	Ren	nove Fo	lder froi	m Proje	ct							
	101 011	Curso de Py	thon b	asi										
	/*	Diccionarios	.ру											
	/*	Ejercicio1.py	,											
	/*	Ejercicio2.py	1											
	/*	ejercicio3.py	/											
	/*	ejercicio4.py	/											
	/*	gelatina.py												
1	/*	Herencia.py												
	/*	Herencia2.p	y i											
		nerenciaMu	itipie.p	y I										
	Line	1, Column 1								Tab	Size: 4		Plain Te	đ

Botón derecho del ratón sobre la carpeta Curso Python y de este New File.

📕 Guardar como		×
$\leftrightarrow \rightarrow \uparrow $	> Este equipo > Disco local (D:) > Curso Python > V 💍 🖉 Buscar en Curso Python	
Organizar 👻 N	ueva carpeta	2
 Documentos Escritorio Imágenes Música Objetos 3D Vídeos Windows (C:) Disco local (D 	Nombre Image: Second state of the second	< > >
No <u>m</u> bre:	setup.py	~
	All Files (*.*) Guardar Cancelar	×

Daremos como nombre setup.py seguido del botón Guardar.



Ctrol + s para guardar.

Desde CMD accederemos a la carpeta de Curso Python

Simbolo del sistema	-	×
D:\Curso Pvthon>_		^
		\sim

Vamos a escribir lo siguiente:

Simbolo del sistema	_	Х
		~
D:\Curso Python>python setup.py sdist		
		\sim

Pulsamos enter.



Si vamos al explorador entramos en la carpeta Curso Python y nos ha creado dos carpetas, una lamada paquetecalculos.egg-info y otra llamada dist.

☐ = Curso Python				- 🗆 ×
Ar Fvo Inicio Compartir	Vista			^ 🗨
* 🗎 📋 🐇	Mover a • K Eliminar •	📜 🖫 📝 📑	Seleccionar todo	
Anclaral Copiar Pegar Acceso rápido	🗓 Copiar a 👻 📑 Cambiar nombre	Nueva Propiedades of the second secon	Invertir selección	
Portapapeles	Organizar	Nuevo Abrir	Seleccionar	
← → × ↑ 📙 > Este equi	ipo → Disco local (D:) → Curso Python		ې ن	Buscar en Curso P
o Creative Cloud File	ombre	Fecha de modificación	Тіро Та	imaño ^
OneDrive Persons	, calculos	10/08/2020 10:20	Carpeta de archivos	
	dist	10/08/2020 19:44	Carpeta de archivos	
💻 Este equipo	modulos	10/08/2020 8:07	Carpeta de archivos	
🕂 Descargas	paquetecalculos.egg-info	10/08/2020 19:44	Carpeta de archivos	
Documentos	Biblioteca.py	06/08/2020 15:48	Python File	3 KB
Escritorio 🗸	clase1.py	06/08/2020 11:05	Python File	1 KB 🗸 🗸
43 elementos 2 elementos sele	ccionados			

Si abrimos la carpeta dist

📙 🛃 📕 🗕		Extraer	d	ist		– 🗆 X		
Archivo Inicio Compart	ir Vista	Herramientas de carpeta co	mprimida			~ 😮		
Anclar al Acceso rápido	Move	ra ▼ X Eliminar ▼ ra ▼ I Cambiar nombre	Nueva carpeta	Propiedades	Seleccionar to	do r nada ón		
Portapapeles		Organizar	Nuevo	Abrir	Seleccionar			
← → × ↑ 🕞 > Este equipo > Disco local (D:) > Curso Python > dist v 🖸 🔎 Buscar en								
Escritorio	Nombre	^	Fecha d	e modificación 🛛 Ti	ро	Tamaño		
📰 Imágenes	🛄 paqueteo	alculos-0.0.0.tar.gz	10/08/2	020 19:44 A	rchivo WinRAR	2 KB		
b Música								
Objetos 3D 1 elemento 1 elemento sel	eccionado 1,22	КВ						

Observaremos un archivo comprimido, este es el que deberíamos de suministrar a las personas interesadas.

Si lo descomprimimos:

🟪 paque	🧱 paquetecalculos-0.0.0.tar.gz (copia de evaluación) —												×
<u>A</u> rchivo	Ór <u>d</u> enes	<u>H</u> erramientas	<u>F</u> avoritos	Opcio <u>n</u> e	a Ay <u>u</u> da	9							
(113) Añadir	Extraer e	n Comprobar	Ver	Eliminar	Q Buscar	Asist	ente Informa	ción	Buscar virus				
↑ 🚦	😐 paquet	ecalculos-0.0.0.t	tar.gz - arch	ivo TAR+G	ZIP, tama	ño de	scomprimido 1	.841 by	tes				~
Nombre		^			Tar	naño	Comprimido	Tipo		Modificado	CRC32		
								Carpe	eta de archivos				
📙 paquet	tecalculos-	0.0.0						Carpe	eta de archivos	10/08/2020 19:44			
-							Ţ	otal 1 c	arpeta				.::

Si abrimos la carpeta paquetecalculos-0.0.0

aquetecalculos-0.0.0.tar.gz (copia de evaluación)											
<u>A</u> rchivo Ór <u>d</u> enes <u>H</u> erramientas <u>F</u> avoritos Opcio <u>n</u>	es Ay <u>u</u> da										
Añadir Extraer en Comprobar Ver Eliminar	O Asis	stente Informad	ción Buscar virus								
↑ Image: The two set of two set of the two set of two set	s-0.0.0 - archivo	o TAR+GZIP, tam	iaño descomprimido 1	.841 bytes				~			
Nombre	Tamaño	Comprimido	Тіро	Modificado	CRC32						
			Carpeta de archivos								
calculos			Carpeta de archivos	10/08/2020 19:44							
paquetecalculos.egg-info			Carpeta de archivos	10/08/2020 19:44							
PKG-INFO	262	?	Archivo	10/08/2020 19:44							
setup.cfg	42	?	Archivo CFG	10/08/2020 19:44							
🗟 setup.py	292	?	Python File	10/08/2020 19:44							
Total 2 campetas y 596 bytes en 3 ficheros											

Ahora para instalarlo desde CMD.

🔤 Símbolo del sistema	-		×
<pre>copying paquetecalculos.egg-info\dependency_links.txt -> paquetecalculos-0.0.0\paquetecalculos copying paquetecalculos.egg-info\top_level.txt -> paquetecalculos-0.0.0\paquetecalculos.egg-in Writing paquetecalculos-0.0.0\setup.cfg</pre>	.egg-i fo	nfo	^
creating dist Creating tar archive removing 'paquetecalculos-0.0.0' (and everything under it)			
D:\Curso Python>cd dist			
D:\Curso Python\dist>pip3 install paquetecalculos-0.0.0.tar.gz			~

Pulsamos enter y empieza la instalación.



cd dist \rightarrow para acceder a la carpeta dist

pip3 install paquetecaldulos-0.0.0 \rightarrow hacemos enter.

Ya empieza la instalación.

Ahora abrimos el archivo Uso_modulos.py que estaba en la carpeta modulos, esta como la movimos de su anterior ubicación al ejecutar daba error.



Este será el resultado:

streep *REPL* [python] (Curso Python) - Sublime Text (UNREGISTERED)	_		×
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
✓► Uso_modulos.py × *REPL* [python] ×			▼
El resultado de la resta es: 125 El resultado de redondear es: 5 ***Repl Closed***			
Line 5, Column 1 Tab Size: 4		Pythor	1

Como podrás observar ahora sí que funciona.

Cerramos todos y ahora vamos a copiar este archivo en descargas y de esta nueva ubicación lo vamos a abrir.



Lo vamos a ejecutar.



Y también funciona.

Funciona desde cualquier lugar, este paquete es capar de leerlo desde donde estés.

Forma parte del Python de nuestro sistema operativo.

Si queremos desinstalar el paquete abrimos de nuevo CMD.



No hace falta ir a ningún directorio en concreto.

🖼 Símbolo del sistema	_	×
C:\Users\pmver>pip3 uninstall paquetecalculos		^
		Ŧ

Pulsamos enter.

Símbolo del sistema - pip3 uninstall paquetecalculos	_		\times
Would remove:			^
<pre>c:\users\pmver\appdata\local\programs\python\python38-32\lib\site</pre>	e-pack	cages\c	alc
ulos*			
<pre>c:\users\pmver\appdata\local\programs\python\python38-32\lib\site</pre>	e-pack	cages∖p	aqu
etecalculos-0.0.0-py3.8.egg-info			
Proceed (y/n)? y			\sim

Contestamos con una y de yes.

🖼 Símbolo del sistema	_		\times
c:\users\pmver\appdata\local\programs\python\python38-32\lib\site	-pack	ages\p	aqu 🔺
etecalculos-0.0.0-py3.8.egg-info			
Proceed (y/n)? y			
Successfully uninstalled paquetecalculos-0.0.0			
C:\Users\pmver>			\sim

Nos informa que el paquete ya se ha desinstalado.

Si ahora ejecutamos el proyecto anterior que funcionaba correctamente, ahora nos dará un mensaje de error.







ARCHIVOS EXTERNOS I (VIDEO 37)

- Cómo trabajar con ficheros externos de texto con el módulo io
- Objetivo: persistencia de datos
- Dos alternativas
 - Manejo de archivos externos.
 - Trabajo con BBDD



• Fases necesarias para guardar información en archivos externos:



Para ello tendremos que trabajar con el módulo estándar io.

Vamos a crear un nuevo proyecto llamado Manejo_archivos.py que guardaremos en nuestra carpeta de trabajo.

D:\Curso Python\Manejo_archivos.py (Curso Python) - Sublime Text (UNREGISTERED)	_		×
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
▲ ► Manejo_archivos.py ×			▼
1 from io import open			
2 3 archivo texto=open("archivo.txt", "w") # Abriendo archivo en modo escritura.			
			<u> </u>
Line 3, Column 77; Saved D:\Curso Python\Manejo_archivos.py (UTF-8) Tab S	ize: 4	Pythor	ı

Lo ejecutamos.

5 *	*REPL* [python] (Curso Python) - Sublime Text (UNREGISTERED)									_		\times	
<u>F</u> ile	<u>E</u> dit	\underline{S} election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp				
4 •	Γ				*RI	EPL* (pyt	hon]	×					▼
***R 	epl	Closed*	**										
	Line 3	3, Column 1								Tab Size: 4		Pythor	1

Vemos que no ha realizado nada, pero si nos vamos a nuestra carpeta de trabajo:

Ar Fivo Inicio Compartir	Vista			×	د ٩
Anclar al Acceso rápido	v X Eliminar ▼ Copiar a × Iminar ▼	Nueva carpeta	Seleccionar todo		
Portapapeles	Organizar	Nuevo Abrir	Seleccionar		
← → · ↑ 🔒 > Este equi	po > Disco local (D:) > Curso Python	>	ې ق 🗸	Buscar en Curso P.	
Acceso rápido	ombre	Fecha de modificación	Тіро Та	maño	^
	calculos	10/08/2020 10:20	Carpeta de archivos		
Escritorio 🗶	dist	10/08/2020 19:44	Carpeta de archivos		
🕂 Descargas 🖈	modulos	10/08/2020 20:09	Carpeta de archivos		
🔮 Documentos 🖈	paquetecalculos.egg-info	10/08/2020 19:44	Carpeta de archivos		
📰 Imágenes 🖈 📲	archivo.txt	11/08/2020 7:19	Documento de te	0 KB	
Curso Python	Biblioteca.py	06/08/2020 15:48	Python File	3 KB	
Documentos	clase1.py	06/08/2020 11:05	Python File	1 KB	
madulas	Control_correo.py	08/08/2020 15:18	Python File	1 KB	
	controlPersonal.py	08/08/2020 7:08	Python File	4 KB	\checkmark
45 elementos					-

Ha creado el archivo, si lo abrimos observaremos que está vacío.





Con el método write escribimos la frase y con el método close cerramos el archivo.

Ejecutamos:

Streps * REPL* [python] (Curso Python) - Sublime Text (UNREGISTERED)												×
<u>F</u> ile	<u>E</u> dit	Selection	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp			
•	► /				*R	EPL* (pyt	thon]	×				▼
**	*Rep]	l Closedª										
	Line	3, Column 1							Tab Size: 4		Python	

Vemos que no ha hecho nada, pero si miramos el contenido del archivo archivo.txt veremos que tiene una frase.

archivo.txt: Bloc de notas	_		Х
<u>A</u> rchivo <u>E</u> dición F <u>o</u> rmato <u>V</u> er <u>A</u> yuda			
Estupendo día para estudiar Python el miércoles.			^
			\sim
D:\Curso Python\Manejo_archivos.py (Curso Python) - Sublime Text (UNREGISTERED)		- 0	×
<u>File Edit Selection Find View Goto Tools Project Preferences Help</u>			
Manejo_archivos.py ×			▼
1 from io import open 2			
<pre>3 archivo_texto=open("archivo.txt", "r") # read</pre>			
4 5 texto=archivo_texto.read() 6			
7 archivo_texto.close()			
9 print <u>(</u> texto <u>)</u>			
Line 9, Column 13	Tab Size: 4	Pytho	n

Abrimos el texto en modo lectura, leemos el contenido del archivo, cerramos el archivo y por último imprimimos su contenido.

Sublime Text (UNREGISTERED)									
<u>File Edit Selection Find View Goto Tools Project Preferences Help</u>									
Manejo_archivos.py × *REPL* [python] ×			▼						
Estupendo día para estudiar Python el miércoles.		and a second s							
Repl Closed									
Line 5, Column 1 Tab	Size: 4	Python							

El método readlines() lee la información línea a línea y lo almacena en una lista.



Si lo ejecutamos

REPL [python] (Curso Python) - Sublime Text (UNREGISTERED)			Х
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
Manejo_archivos.py × *REPL* [python] ×			▼
['Estupendo día para estudiar Python \n', ' el miércoles.']	-		
Repl Closed			
Line 4, Column 1	lab Size: 4	Python	

Vemos como lo trata de una lista.

¿Cómo acceder a la primera línea:



Si ejecutamos:

5 *	REPL*	[python] (C	urso P	ython) -	Sublim	ne Text (UNREGIST	ERED)		_		×
<u>F</u> ile	<u>E</u> dit	<u>S</u> election	Find	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	Help			
4 •					*R	EPL* (pyt	hon]	×				▼
Estupendo día para estudiar Python												
***R	epl	Closed*										
	Line 5	, Column 1							Tab Size:	ļ.	Python	

Abrir archivo para agregar información.



Ejecutamos:

REPL [python] (Curso Python) - Sublime Text (UNREGISTERED)										
<u>F</u> ile	<u>E</u> dit	<u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	Help	
<					*RE	PL* (pyt	hon]	×		▼
***	Repl	Closed*								
	Line 3	3, Column 1							Tab Size: 4 Python	

No vemos nada, pero si abrimos el archivo archivo.txt

archivo.txt: Bloc de notas	_	×
<u>A</u> rchivo <u>E</u> dición F <u>o</u> rmato <u>V</u> er <u>A</u> yuda		
Estupendo día para estudiar Python el miércoles. siempre es una buena ocasión para estudir Python		^
		\sim

Veras como le hemos agregado el contenido.




ARCHIVOS EXTERNOS II (VIDEO 38)

- ¿Cómo trabajar con ficheros externos de texto con el módulo io
 - Punteros de texto

Recordando el capitulo anterior vamos a leer con contenido de un archivo de texto.



Este será el resultado:



Si abrimos el archivo de texto verás que el contenido es el mismo.

archivo.txt: Bloc de notas	_	×
<u>A</u> rchivo <u>E</u> dición F <u>o</u> rmato <u>V</u> er <u>A</u> yuda		
Estupendo día para estudiar Python el miércoles. siempre es una buena ocasión para estudir Python		^
		\sim

Si no le decimos nada el puntero se posición al principio del texto.

Cuando leemos el fichero el cursor se pone al final del texto, si queremos leer de nuevo el texto como el cursor está al final este no muestra nada.





¿Cómo podemos modificar la posición de un puntero? Con el método seek()

Antes de imprimir por segunda vez nos iremos de nuevo al principio del texto seek(0)



Este será el resultado:



Vamos a imprimir desde la posición 11

D:\Curso Python\Manejo_archivos.py (Curso Python) - Sublime Text (UNREGISTERED)	_		×
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
Manejo_archivos.py ×			▼
1 from io import open			
2 3 archivo_texto=open("archivo.txt", "r") 4			
5 archivo_texto.seek(11) 6			
7 print(archivo_texto.read())			
Line 7, Column 28; Saved D:\Curso Python\Manejo_archivos.py (UTF-8) Tab Size	4	Python	

Este será el resultado:



Con la función read() podemos poner un valor e imprimirá hasta el valor que le asignemos.

D:\Curso Python\Manejo_archivos.py (Curso Python) - Sublime Text (UNREGISTERED)	_		×							
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp										
Manejo_archivos.py ×			▼							
1 from io import open 2		NELSON EN LA CALENCE AND								
<pre>3 archivo_texto=open("archivo.txt", "r") 4</pre>	<pre>3 archivo_texto=open("archivo.txt", "r") 4</pre>									
5 print(archivo_texto.read(11))										
Line 5, Column 30; Saved D:\Curso Python\Manejo_archivos.py (UTF-8) Tab Size: 4	ł	Pythor	1							

Este será el resultado:

REPL [python] (Curso Python) - Sublime Text (UNREGISTERED)												×
<u>F</u> ile	<u>E</u> dit	Selection	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp			
4 Þ	Γ				*RI	EPL* (pyt	hon]	×				▼
Est ***	ipen Repl	do d Closed*										
	Line	4, Column 1							Tab Size: 4		Python	

D:\Curso Python\Manejo_archivos.py (Curso Python) - Sublime Text (UNREGISTERED)	_		×
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
▲ ► Manejo_archivos.py ×			▼
1 from io import open			
2 3 archivo_texto=open("archivo.txt", "r") 4			
5 print(archivo_texto.read(11))			
6 7 print(archivo_texto.read∯)			
Line 7, Column 26 Tab Size:-	4	Pythor	1

Ahora como la posición de puntero está en el 11 si realizamos un segundo print este imprimirá desde la posición 11 hasta el final.



Primer read y segundo read.

Ahora queremos imprimir desde la mitad del texto hasta el final.



Len(archivo_texto.read())/2 \rightarrow nos retorna el valor que será la posición intermedia del texto, el puntero se posicionar justo a la mitad del texto.

Cuando ejecutemos este será el resultado:



También nos podemos posicionar en líneas.



El cursor se sitúa al final de la primera línea y luego imprime el resto.



Si queremos abrir un archivo de escritura y lectura le agregaremos los parámetros "r+"



Como por defecto cuando abrimos un archivo, el puntero se encuentra al principio del texto y le damos una instrucción write este texto sustituirá al texto que había.



Si imprimimos utilizando la función readlines(), nos devuelve una lista con sus respectivos saltos de línea representados con la \n.





En este ejemplo vamos a modificar la segunda línea del archivo de texto desde el exterior.

Si ejecutamos veremos los cambios del archivo de texto.

archivo.txt: Bloc de notas	_		×
<u>A</u> rchivo <u>E</u> dición F <u>o</u> rmato <u>V</u> er <u>A</u> yuda			
Comienzo del texto estudiar Pvthon			\sim
Esta línea ha sido incluida desde el exterior			
siempre es una buena ocasión para estudir PythonComienzo	del	texto	
			~





SERIALIZACIÓN I (VIDEO 39)

• Serialización de colecciones, objetos.



- Pickle
 - Método dump(): volcado de datos al fichero binario externo.
 - Método load(): carga de los datos del fichero binario externo.

Vamos a crear un proyecto nuevo llamado trabajo_serializacion.py.



Lo ejecutamos:

5	*REPL* [python] (Curso Python) - Sublime Text (UNREGISTERED)												Х
<u>F</u> ile	<u>E</u> dit	Selection	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	Help				
••					*R	EPL* (py	thon]	×					▼
***	Repl	Closed*											
	Line 3	8, Column 1								Tab Size: 4		Python	

Ahora veremos en nuestra carpeta de trabajo el archivo "lista_nombres".

2 = Curso Python										
Ar F vo Inicio Compart	AD F 1/00 Inicio Compartir Vista									
*	Mover a → X Eliminar →		Seleccionar todo	da						
Anciarai Copiar Pegar Acceso rápido	Copiar a 👻 💷 Cambiar nombre	carpeta	💾 Invertir selección							
Portapapeles	Organizar	Nuevo Abrir	Seleccionar							
\leftarrow \rightarrow \checkmark \uparrow \square \rightarrow Este	equipo > Disco local (D:) > Curso Python	• •		O Buscar en Cu	irso P					
👆 Descargas 🔷 🔺	Nombre	Fecha de modificación	Тіро	Tamaño	^					
Documentos	📑 gelatina.py	09/08/2020 7:04	Python File	1 KB						
Escritorio	📄 Herencia.py	07/08/2020 12:30	Python File	2 KB						
	📄 Herencia2.py	07/08/2020 12:26	Python File	1 KB						
Imagenes	📴 HerenciaMultiple.py	09/08/2020 10:47	Python File	1 KB						
👌 Música	lista_nombre	11/08/2020 11:32	Archivo	1 KB						
🧊 Objetos 3D	📄 listaNif.py	04/08/2020 21:28	Python File	1 KB						
Vídeos	📄 Listas.py	02/08/2020 12:59	Python File	1 KB						
Windows (C:)	📄 Manejo_archivos.py	11/08/2020 10:05	Python File	1 KB						
Disco local (D:)	澷 Nacimiento.py	09/08/2020 18:05	Python File	1 KB						
47 elementos		01/00/0000 00 07	B.(1 E)	2.00						

Este no lo podremos abrir.

Ahora lo queremos rescatar y leer la información.



Ejecutamos:

REPL [python] (Curso Python) - Sublime Text (UNREGISTERED)		×	
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
trabajo_serializacion.py × *REPL* [python] ×			▼
['Pedro', 'Ana', 'María', 'Isabel']			
Repl Closed			
Line 4, Column 1	Tab Size: 4	Pythor	1





SERIALIZACION II (VIDEO 40)

• Serialización de objetos

Vamos a crear un nuevo proyecto llamado "Serializar objetos.py".

Aprovecharemos parte del código "módulo vehículos.py" que copiaremos al nuevo proyecto.



Vamos a realizar las correspondientes modificaciones:



Ejecutamos.

5	*REPL* [python] (Curso Python) - Sublime Text (UNREGISTERED)											×
<u>F</u> ile	<u>E</u> d	it <u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	Help			
•		Serializar o			*RE	EPL* (pyt	hon]	×				
**	*Rep	ol Closed	***									
	Lin	ne 3, Column	I						Tab Size: 4		Python	i i

Ahora vamos a buscar el fichero "losCoches".

Anclar al Acceso rápido Portapapeles	Python	Vista V Mover a V Copiar a V Copiar a V Organizar	Nueva carpeta Nuevo	Propiedades Abrir	Seleccionar t No seleccion Invertir selec Seleccion	odo ar nada ar	× ^ ¶
← → · ↑ <mark>·</mark> ·	Este equi	ipo → Disco local (D:) → Curso Python	1		ٽ ~	🔎 Buscar en G	Curso P
👆 Descargas	^ N	ombre	Fecha	a de modificación	Тіро	Tamaño	^
Documentos		HerenciaMultiple.py	09/08	/2020 10:47	Python File	1 KB	
Escritorio] lista_nombre	11/08	/2020 11:32	Archivo	1 KB	
📰 Imágenes		🔋 listaNif.py	04/08	/2020 21:28	Python File	1 KB	
Música		listas.py	02/08	/2020 12:59	Python File	1 KB	
Chistor 2D		losCoches	12/08	/2020 8:35	Archivo	1 KB	
Jobjetos 3D		Manejo_archivos.py	11/08	/2020 10:05	Python File	1 KB	
Vídeos	4	Nacimiento.py	09/08	/2020 18:05	Python File	1 KB	
🏪 Windows (C:)		훩 nif.py	04/08	/2020 20:07	Python File	1 KB	
Disco local (D:)	v .	Polimorfismo.py	08/08	/2020 7:51	Python File	1 KB	~
49 elementos							

Ahora vamos a rescatar la información que hay en este fichero:



Al final de este proyecto agregamos el siguiente código.

Lo ejecutamos:

REPL [python] (Curso Python) - Sublime Text (UNREGISTERED)							
Eile Edit Selection Find View Goto Tools Project Preferences Help							
Serializar objetos.py × *REPL* [python] ×							
<mainvehiculos 0x015ea1c0="" at="" object=""> <mainvehiculos 0x015ea880="" at="" object=""> <mainvehiculos 0x01be6088="" at="" object=""></mainvehiculos></mainvehiculos></mainvehiculos>	Ϋ́.						
Line 6, Column 1 Tab Size:	4	Pytho	n				

La información tiene mucho que desear.

La línea print(c) la cambiaremos por print (c.estado) ya que aprovecharemos este método.



Ejecutamos, este será el resultado:



En este proyecto hemos hecho la escritura y la lectura en el mismo, pero podríamos haber realizado uno para escribir y otro para leer.



Para que este funcione hemos de agregar la clase, ya que la variable c que es de este tipo no lo reconocería, abrimos el fichero en lectura binario y así podremos leer la información.







GUARDADO PERMANENTE (VIDEO 41)

• Guardar datos de forma permanente en ficheros

Vamos a crear un nuevo proyecto llamado infoPermanente.py



Si lo ejecutamos observaremos el siguiente mensaje:

REPL* [python] (Curso Python) - Sublime Text (UNREGISTERED)	-	- 🗆	×
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
✓ ► infoPermanente.py × *REPL* [python] ×			▼
Se ha creado una persona nueva con el nombre de Sandra			
Repl Closed			
Line 4, Column 1	Tab Size: 4	Python	1

Vamos a agregar más código:



Este será el resultado:

REPL [python] (Curso Python) - Sublime Text (UNREGISTERED)	_		×
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
✓ ► infoPermanente.py × *REPL* [python] ×			▼
Se ha creado una persona nueva con el nombre de Sandra		Minerinte.	
Se ha creado una persona nueva con el nombre de Antonio			
Se ha creado una persona nueva con el nombre de Ana			
Sandra Femenino 29			
Antonio Masculino 39			
Ana Femenino 19			
****Repl Closed***			
Line 9, Column 1 Tab Size: 4		Python	

Ahora el siguiente objetico es que esta información se grabara en un fichero externo.

import pickle

class Persona:

def __init__(self, nombre, genero, edad):
 self.nombre=nombre
 self.genero=genero
 self.edad=edad
 print("Se ha creado una persona nueva con el nombre de ", self.nombre)

def __str__(self):

return "{} {} {}".format(self.nombre, self.genero, self.edad)

class listaPersonas:

personas=[]

def __init__(self):

listaDePersonas=open("ficheroExterno", "ab+") #Agregar información binaria listaDePersonas.seek(0)

try:

self.personas=pickle.load(listaDePersonas)
print("Se cargaron {} personas del fichero

externo".format(len(self.personas)))

except:

print("El fichero está vacío")

finally:

listaDePersonas.close() del(listaDePersonas)

def agregarPersonas(self,p):
 self.personas.append(p)
 self.guardarPersonasEnFicheroExterno()

def mostrarPersonas(self): for p in self.personas: print(p)

def guardarPersonasEnFicheroExterno(self): listaDePersonas=open("ficheroExterno", "wb") pickle.dump(self.personas, listaDePersonas) listaDePersonas.close() del(listaDePersonas)

miLista=listaPersonas()

persona=Persona("Sandra","Femenino",29)

miLista.agregarPersonas(persona)

Este será el resultado:

REPL [python] (Curso Python) - Sublime Text (UNREGISTERED)	_		×			
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp						
infoPermanente.py × *REPL* [python] ×			▼			
El fichero está vacío		UNATA AND A	· 1			
Se ha creado una persona nueva con el nombre de Sandra			- 1			
Repl Closed						
Line 5, Column 1 Tab Size: 4		Python				

Agregamos



Este será el resultado:



Muestra 4 porque se ha ejecutado 4 veces.

Vamos a comprobar el fichero externo.

📙 🛃 🥃 🗧 🗸 Curso Python				- 🗆	×
Archivo Inicio Compartir	Vista				^ ?
Anclar al Copiar Pegar Acceso rápido	Mover a * Kliminar *	Nueva Annevo Nuevo Nuevo Nuevo Nuevo Nuevo Abrir	Seleccionar todo	da	
Fortapapeles	Organizai	NUEVO ADITI	Seleccional	_	
← → × ↑ 📙 > Este eq	uipo > Disco local (D:) > Curso Python	l >	v Ö /	Buscar en Cur	so P
🔶 Descargas 🛛 ^	Nombre	Fecha de modificación	Тіро	Tamaño	^
🔮 Documentos	Diccionarios.py	03/08/2020 7:50	Python File	1 KB	
Escritorio	Ejercicio1.py	03/08/2020 12:40	Python File	1 KB	- 6
🔤 Imágenes	Ejercicio2.py	03/08/2020 13:00	Python File	1 KB	
Música	📄 ejercicio3.py	03/08/2020 13:06	Python File	1 KB	
	📄 ejercicio4.py	04/08/2020 11:04	Python File	1 KB	
Ubjetos 3D	📄 ficheroExterno	12/08/2020 10:58	Archivo	1 KB	
🤮 Vídeos	澷 gelatina.py	Tino: Archivo		1 KB	
🏪 Windows (C:)	澷 Herencia.py	Tamaño: 208 bytes	2	2 KB	
👝 Disco local (D:) 🔍	澷 Herencia2.py	Fecha de modificación: 12/08/	2020 10:58	1 KB	~
52 elementos					::: 🛌

Este fichero ya contiene información.

Lo vamos a borrar.

Ejecutamos de nuevo.

REPL [python] (Curso Python) - Sublime Text (UNREGISTERED)	_		×
<u>File Edit Selection Find View G</u> oto <u>T</u> ools <u>P</u> roject Prefere <u>n</u> ces <u>H</u> elp			
✓ ► infoPermanente.py × *REPL* [python] ×			▼
El fichero está vacío		1000 Sections	
Se ha creado una persona nueva con el nombre de Sandra			- 1
La información del fichero externo es la siguiente:			- U
Sandra Femenino 29			- 1
Repl Closed			
Line 7, Column 1 Tab Size: 4		Python	

Ahora ejecutaremos nuestro programa cambiando los datos de la persona.



		~
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp		
infoPermanente.py × *REPL* [python] ×		▼
Se cargaron 2 personas del fichero externo	in the second second	
Se ha creado una persona nueva con el nombre de Ana		
La información del fichero externo es la siguiente:		
Sandra Femenino 29		
Antonio Masculino 49		
Ana Feminino 19		
Repl Closed		
Line 9, Column 1 Tab Size: 4	Python	





INTERFACES GRÁFICAS (VIDEO 42)

• Creación de interfaces gráficas o Librería Tkinter

¿Qué son las interfaces gráficas?

Son ventanas en las que los usuarios pueden interactuar con ellas.

<u>ه</u>		-	\times
	Fuente Estilo Tamaño		
N K S	Arial Courier Verdana		
=			

- También denominadas GUI, son intermediarios entre el programa y el usuario. Formadas por un conjunto de gráficos como ventanas, botones, menús, casilla de verificación, etc.
- Librerías con las que trabajar cara crear GUI:
 - o Tkinter
 - o WxPython
 - o PyQT
 - PyGTK
- Tkinter es un "puente" entre Python y la librería TCL/TK
- Si trabajas en Linux la podrás instalar con la siguiente instrucción

\$ sudo apt-get install python3-tk, lo realizaremos desde la consola.

Estructura de ventana con Tkinter.



En este capitulo vamos a ver como se construye la ventana, lo que es la raíz.

Dentro de nuestra carpeta de trabajo hemos creado una nueva carpeta llamada gráfico, para guardar los proyectos de interfaces gráficas.

Nuestro primer proyecto se llamará "primera_interface.py".



Guardamos y ejecutamos



mainloop() : es un bucle infinito, este tiene que esta siempre al final.

Ahora vamos agregar un título a la ventana:



Este será el resultado:



Vamos a evitar que el usuario pueda redimensionar una ventana.



Ejecutamos



No nos deja modificar el ancho ni el alto, además el botón de maximizar y restaurar están deshabilitados.

Si lo cambiamos por raíz.resizable(1,0), podremos modificar a lo ancho pero no a lo alto. También es correcto raíz.resizable(True,False).

Vamos a cambiar el icono que aparece por defecto que es una pluma.

Tenemos que tener un nuestro directorio de trabajo un archivo de tipo .ico También podemos preguntar a Google por conversores .ico



Este será el resultado:



Para modificar el tamaño de la ventana:



Este será el resultado:

🗯 Ventana de prueba	_	×

Cambiar el color de fondo.



Este será el resultado:

🧯 Ventana de prueba	—	×

Si nos vamos a la carpeta donde tenemos el proyecto.

	graficos			_	· 🗆	×
Ar Evo Inicio	Compa	rtir Vista				^ ° E
	Icono	s muy grandes 🛋 Iconos grandes 🔺	Casillas de elemento		N N	
Panel de 📺 navegación •	E Lista	s medianos il iconos pequeños Vi EE Detalles Vi actu	Extensiones de nombre de archivo ta al Elementos ocultos	Ocultar elementos seleccionados	Opciones	
Paneles		Diseño	Mostrar u ocultar			
$\leftarrow \rightarrow ~ \star ~ \uparrow$	> Est	e equipo > Disco local (D:) > Curso Python >	graficos 🗸	Ö , B	uscar en grafic	os
🖊 Descargas	^	🗯 manzana.ico				
🔮 Document	tos	📄 primera_interface.py				
Escritorio						
📰 Imágenes						
👌 Música						
🧊 Objetos 3D)					
📑 Vídeos						
🏪 Windows ((C:)					
🕳 Disco local	I (D:) 🗸					
2 elementos 1	elemento	eleccionado 199 bytes			•	-

Y hacemos doble clic en dicho proyecto.

C:\WI	NDOWS\py.exe				_	×
	🍯 Ventana de prueba	_	\times			^
						~

Se nos abre la ventana pero con la consola detrás.

Para que la consola no se abra vamos a cambiar la extensión py por pyw, la w viene de Window.

12 =	graficos	-		×
Ar Evo Inicio	Compartir Vista			<u>^</u>
Panel de mavegación •	■ Iconos muy grandes ■ Iconos grandes ■ Iconos medianos ■ Iconos pequeños ■ Iconos medianos ■ Iconos pequeños ■ Iconos pequeños ■ Iconos medianos ■ Iconos pequeños ■ Iconos pequeños ■ Iconos medianos ■ Iconos pequeños ■ Iconos ■ Icono	Ocultar elementos seleccionados	Opciones	
Paneles	Diseno Mostrar u ocultar			
$\leftarrow \ \ \rightarrow \ \checkmark \ \wedge$	> Este equipo > Disco local (D:) > Curso Python > graficos	O D Bus	scar en grafic	:os
🖶 Descargas 🔮 Document	 manzana.ico primera_interface.pyw 			
📃 Escritorio				
📰 Imágenes				
👌 Música				
🧊 Objetos 3[
📲 Vídeos				
······	(c) Y		_	
2 elementos 1	elemento seleccionado 199 bytes		•	-

Hacemos doble clic en él.



Ya se abre sin la consola.





INTERFACE GRÁFICAS II (VIDEO 43)

Creación de interfaces gráficos

 Librería Tkinter. Manejo de Frames

En este capítulo vamos a ver cómo crear Frames.





Este será el resultado cuando lo ejecutemos:

🧯 Ventana de prueba	_	×

La raíz se adapta al tamaño del Frame.

Si redimensionamos la raíz veremos la diferencia.

🗯 Ventana de prueba	_	×



El parámetro side puede ser igual a left, right, top y bottom.

En el ejemplo anterior este será el resultado:

🧯 Ventana de prueba	-	×

El parámetro anchor maneja los puntos cardinales: n ,ne, e, se, s, sw, w, nw, o center.



Este será el resultado:

🇯 Ventana de prueba	_	×

Para hacer el rellanado utilizaremos el parámetro fill, que puede tener los siguiente parámetros: x, y, both y none.



Este será el resultado:




Para poder redimensionar hacia abajo.

💰 Ventana de prueba	_	×

Si queremos que se expanda en las dos direcciones utilizaremos both.



Este será el resultado:



También podemos definir bordes.



🗯 Ventana de prueba	_	×

También hay un tipo de borde llamado sunken.

Podemos cambiar el icono del cursor.



Cuando lo ejecutemos y nuestro cursos pase por encima del Frame, este cambiará.



Otro tipo de cursor, pirate.

Tolo lo que hemos aplicado al Frame con el método config se lo podemos aplicar a la raíz.

Ejemplos:

raiz.config(bg="blue")

raiz.config(bd=45)

raiz.config(cursor="hand2)





INTERFACES GRÁFICAS III (VIDEO 44)

- Creación de interfaces gráficas
 - o Liberia Tkinter. Widget Label

¿Qué son los Label?

- Wigets utilizados para mostrar texto o imágen.
- Tiene como única finalidad mostrar texto, no se puede interactuar con él (borrar, arrastrar, etc)

Sintaxis

• variableLabel=Label(contenedor, opciones)

Opción	Descripción
Text	Texto que se muestra en el Label
Anchor	Controla la posición del texto si hay espacio suficiente para él (center por defecto)
Bg	Color de fondo
Bitmap	Mapa de bits que se mostrará como gráfico
Bd	Grosor del borde (2 px por defecto)
Font	Tipo de fuente a mostrar
Fg	Color de la fuente
Width	Ancho de Label en caracteres (no en píxeles)
Height	Altura de Label en caracteres (no en píxeles)
Image	Muestra imagen en el Label en lugar de texto
justify	Justificación del texto del Label

Vamos a realizar un nuevo proyecto llamado practicaLabel.py

Este será el resultado:



Hemos dado dimensiones en miFrame pero al empaquetar miLabel este se ajusta al texto, para que esto no suceda realizaremos lo siguiente:





Esta dos líneas las podremos simplificar:

miLabel=Label(miFrame, text="Hola alumnos de Python")

miLabel.place(x=100, y=200)

En una sola línea.

Label(miFrame, text="Hola alumnos de Python).place(x=100,y=200)



Este será el resultado:





🖗 tk		_	×
	Hola alumnos de Python		

Si en lugar de utilizar texto queremos utilizar una imagen para ello trabajaremos con imágenes png y gif que son las que admite Python.









INTERFACES GRÁFICAS IV (VIDEO 45)

- Creación de interfaces gráficas
 - Liberia Tkinter. Widget Entry

Partimos de un nuevo proyecto llamado EjemploEntry.py

D:\Curso Python\graficos\EjemploEntry.py • (Curso Python) - Sublime Text (UNREGISTEREE)) —		\times
<u>File E</u> dit <u>Selection Find View G</u> oto <u>T</u> ools <u>P</u> roject Prefere <u>n</u> ces <u>H</u> elp			
◄ ► EjemploEntry.py ●			▼
1 from tkinter import * 2			
3 raiz=Tk() 4			
5 CuadroTexto=Entry(raiz) 6			
7 CuadroTexto.pack() 8			
9 raiz.mainloop <u>()</u>			
Line 9, Column 16 Tab Siz	e: 4	Python	

Hemos creado nuestro primer cuadro de texto, este será el resultado:

	-4		×		
🧳 tk		_		ı ×	(
	!Hola Py	thon!			

Ahora vamos a modificar el tamaño de la ventana, crearemos un Frame llamado miFrame le diremos que el CuadroTexto de tipo Entry pertenece a miFrame y por último con la función place le ubicaremos una posición.





El siguiente paso será crear una etiqueta al lado del cuadro de texto:





El método grid(), crea como una tabla donde podremos colocar nuestros objetos con mayor facilidad.

Los parámetros son row que hace referencia a la fila y columna que hace referencia a la columna.





Este será el resultado:



Si modificamos el tamaño de la ventana, este será el resultado:

🖗 tk		_	×
Non	bre:		

Intenta agregar una etiqueta y cuadro de texto para los apellidos, en la siguiente página encontrarás el resultado:



Ø	tk		_	×
		Nombre:		
		Apellidos:		

Ahora vamos a crear otro cuadro de texto y etiqueta para dirección.

17 18 19	CuadroTextoDireccion=Entry(miFrame) CuadroTextoDireccion.grid(<i>row</i> =2, <i>column</i> =1)		
20 21	<pre>DireccionLabel=Label(miFrame, text="Dirección:")</pre>		
22 23 24	DireccionLabel.grid(<i>row</i> =2, <i>column</i> =0)		
25	raiz.mainloop <u>()</u>		
	ine 25, Column 16	Tab Size: 4	Python

🦸 tk	_	\times
Nombre: Apellidos:		_
Dirección:		

Ahora vamos a modificar el texto Dirección por dirección de casa:



Este será el resultado:

🧳 tk	_	\times
Nombre: Apellidos:		
Dirección de casa:		

Para poder alinear los objetos utilizaremos la propiedad sticky, y los valores que admite son los puntos cardinales.

nw	n	ne
w		е
SW	S	se





pady(vertical), padx(horizontal): nos ayudará a separar elementos uno verticalmente y el otro horizontalmente.



Este será el resultado:

🖗 tk	_	×
Nombre:		
Apellidos:		
Dirección de casa:		

Ahora también se lo ponemos a los cuadros de texto:

D:\Curso Python\graficos\EjemploEntry.py • (Curso Python) - Sublime Text (UNREGISTERED)	_		×
<u>F</u> ile <u>E</u> dit <u>S</u> election Find <u>V</u> iew <u>G</u> oto <u>T</u> ools <u>P</u> roject Prefere <u>n</u> ces <u>H</u> elp			
✓ ► EjemploEntry.py ●			▼
1 from tkinter import *		Bill Provinces	
2 raiz=Tk()		Manual Contractors Manual Contractors Manual Contractors	
<pre>3 miFrame=Frame(raiz, width=1200, height=600)</pre>			
4 miFrame.pack()			
5 6 CurdesTauteNeetes Estav(ciEsers)			
Cuadro lex toxomore entry (mirrame)			
8			
9 NombreLabel=Label(miFrame, text="Nombre:")			
10 NombreLabel.grid(row=0, column=0,sticky="e",padx=10,pady=10)			
12 CuadroTextoApellidos=Entry(miFrame)			
13 CuadroTextoApellidos.grid(row=1,column=1,padx=10,pady=10)	<u> </u>		
14			
15 ApellidosLabel=Label(miFrame, <i>text</i> ="Apellidos:")			
16 ApellidosLabel.grid(row=1, column=0,sticky="e",padx=10,pady=10)			
17			
18 CuadroTextoDireccion=Entry(miFrame)			
19 CuadrolextoDireccion.grid(row=2,column=1,padx=10,pady=10)			
21 DirectionLabel=Label(mirrame, text= Direction de Casa;)			
22 DirectionLabel.grid(row=2, column=0,sticky= e, poax=10, poay=10)			
24			
25 raiz.mainloon()			
Line 25, Column 16 Tab Size:	4	Python	

Este será el resultado:

🖗 tk	_	×
Nombre:		
Apellidos:		
Dirección de casa:		

Con el método config podemos cambiar el color de los textos de las cajas de texto.



🖗 tk	_	×
Nombre:	Juan	
Apellidos:		
Dirección de casa:		

Para modificar la alineación del texto utilizaremos el método justify.



Este será el resultado:

🖉 tk	_		×
Nombre:		Juan	
Apellidos:			
Dirección de casa:			_

También admite right.

🖉 tk	_	\times
Nombre:		Juan
Apellidos:		
Dirección de casa:		

Como utilizar un campo Password.

Vamos agregar debajo del nombre un campo password.

D:\Curso Python\graficos\EjemploEntry.py (Curso Python) - Sublime Text (UNREGISTERED)	_		×
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
✓ ► EjemploEntry.py ×			▼
1 from tkinter import *			
3 miFrame=Frame(raiz, width=1200, height=600)			
4 miFrame.pack()			
5			
<pre>6 CuadroTextoNombre=Entry(miFrame)</pre>			
7 CuadroTextoNombre.grid(<i>row=0,column=1,padx=10,pady=10</i>)			
o 9 Nombrelabel=label(miFrame_text="Nombre:")			
10 NombreLabel.grid(row=0, column=0,sticky="e",padx=10,pady=10)			
11			
12 CuadroTextoPass=Entry(miFrame)			
13 CuadroTextoPass.grid(row=1,column=1,padx=10,pady=10)			
14 CuadrolextoPass.config(show="*")			
16 PassLabel=Label(miFrame, text="Password:")			
17 PassLabel.grid(row=1, column=0,sticky="e",padx=10,pady=10)			
18			
19 CuadroTextoApellidos=Entry(miFrame)			
20 CuadroTextoApellidos.grid(<i>row=2,column=1,padx=10,pady=10</i>)			
21 22 ApellidosLabel=Label(miErame, text="Apellidos:")			
23 ApellidosLabel.grid(row=2, column=0, sticky="e", padx=10, pady=10	0)		
24			
25 CuadroTextoDireccion=Entry(miFrame)			
26 CuadroTextoDireccion.grid(row=3,column=1,padx=10,pady=10)			
27 28 DireccionLabel=Label(miFrame text="Dirección de casa:")			
29 DireccionLabel.grid(row=3, column=0, sticky="e", padx=10, pady=10	0)		
30	<u> </u>		
31			
32 raiz.mainloop <u>()</u>			
Line 32, Column 16 Tab Size: 4		Python	

🦸 tk	_		×
Nombre:			
Password:	******	****	
Apellidos:			
Dirección de casa:			





INTERFACES GRÁFICAS V (VIDEO 46)

- Creación de interfaces gráficas
 - Librería Tkinter. Widgets text y Button

Text: Wigets utilizados para introducir texto largo.

Button: Botones para interactuar con la interfaz

Partiendo del proyecto anterior vamos a agregar el siguiente código:



🖉 tk	-	×
Nombre:		
Password:		
Apellidos:		
Dirección:		
Comentarios:		

Vamos a darle una dimensiones más razonables.



Este será el resultado:

Ø +k	_	×
Ψ LN		^
Nombre:		
Password:		
Apellidos:		
Dirección:		
Comentarios:		

Vamos a crear una barra de desplazamiento vertical, para ello construiremos un objeto scrollbar y decirlo que pertenece al texto que nosotros le especifiquemos.



Agregamos el correspondiente código:

🧳 tk	_	×
Nombre:		
Password:		
Apellidos:		 -
Dirección:		 -
		_
Comentarios:		Ê
		~

Vamos a mejorar el scrollbar.





Para que el posicionar funcione correctamente vamos a realizar lo siguiente:



🖗 tk	– 🗆 ×
Nombre:	
Password:	
Apellidos:	
Dirección:	
Comentarios:	y fácil de apre nder por lo que sigo con mucho i nterés el curso.
	· · · · · · · · · · · · · · · · · · ·

Ahora vamos a agregar un botón con el método Button.



Este será el resultado:

🖗 tk	_	×
Nombre:		
Password:		_
Apellidos:		 -
Dirección:		_
Comentarios:		_
	Enviar	~

Vamos a agregar instrucciones en el botón.



Agrega las siguientes modificaciones:

Este será el resultado cuando presionemos el botón:

🖉 tk	_	×
Nombre:	Juan	
Password:		
Apellidos:		_
Dirección:		 -
Comentarios:		
	Enviar	~

Para obtener información de un cuadro de texto utilizaremos la función get.





INTERFACES GRÁFICAS VI (VIDEO 47)

- Creación de interfaces gráficas
 - Interfaz gráfica calculadora

Vamos a realizar un nuevo proyecto llamado practicaCalculadora.py

🗾 D:\	Curso Python\g	praficos	\practic	aCalcu	ladora.p	y (Curso F	yth —		\times
<u>F</u> ile <u>E</u>	dit <u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp	
4 ►	practicaCalco	uladora.	ру ×						▼
1 2	from tki	nter	impo						
3 4	raiz=Tk()							
5	miFrame=	Frame	(rai	z)					
7 8 9	miFrame.	pack(0						
10 11	raiz.mai	nloop	<u>)</u>						
	ine 11, Column 1	6					Tab Size: 4	Pyth	non

Este será el resultado:



Vamos a diseñar la pantalla.





Vamos a realizar la primera fila de botones:



Este será el resultado:



Como podrás observar la pantalla no está centrada.

	#Pantalla
10	pantalla=Entry(miFrame)
11	<pre>pantalla.grid(row=1,column=1, padx=10,pady=10, columnspan=4)</pre>
12	<pre>pantalla.config(background="black", fg="#03f943", justify="right")</pre>
13	



Ahora vamos a realizar la fila 2 de botones



Este será el resultado:

<i>q</i> –	-		×
7	8	9	1
4	5	6	x

Vamos a realizar la tercera fila



<i>q</i> –	-		×
7	8	9	1
4	5	6	x
1	2	3	-

Ahora vamos a realizar la fila 4

43 44 45 46 47 48 49 50 51 52 53 54	<pre>#Fila 4 boton0=Button(miFrame, text="0", width=3) boton0.grid(row=5,column=1) botonComa=Button(miFrame, text=", width=3) botonComa.grid(row=5,column=2) botonIgual=Button(miFrame, text="=", width=3) botonIgual.grid(row=5,column=3) botonSum=Button(miFrame, text="+", width=3) botonSum.grid(row=5,column=4) raiz.mainloop()</pre>		
Lin	ne 54, Column 16	Tab Size: 4	Python

Este será el resultado:

<i>q</i> –	-		×
7	8	9	1
4	5	6	x
1	2	3	-
0		=	+

Vamos a modificar la imagen que se encuentra en la parte superior izquierda que es una pluma por otra imagen.





El siguiente paso será que al pulsar un botón la información aparezca en pantalla.





INTERFACES GRÁFICAS VII (VIDEO 48)

- Creación de interface gráficas
 - Funcionalidad calculadora

Ahora lo primero que vamos a realizar es que al pulsar el botón 4 aparezca en pantalla el número 4.



En el apartado de pantalla definimos la variable numeroPantalla de tipo StringVar().

El siguiente paso es asociar pantalla con la variable numeroPantalla.



Definimos una función llamada numeroPulsado() para asignar a la variable que tenemos asignada a la pantalla el valor "4"

D:\Curso Python\graficos\practicaCalculadora.py • (Curso Python) - Sublime Text (UNREGISTERED)	- 🗆	×
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp		
✓ ▶ practicaCalculadora.py ●		▼
34 #Fila 2		· 1
35 boton4=Button(miFrame, text="4", width=3, command=numeroPulsado) 👡		
36 boton4.grid(row=3,column=1)	in the second second	
37 boton5=Button(miFrame, <i>text</i> ="5", <i>width</i> =3)		
38 boton5.grid(row=3,column=2)		
<pre>39 boton6=Button(miFrame, text="6", width=3)</pre>		
40 boton6.grid(row=3, column=3)		
41 botonMult=Button(miFrame, text="x", width=3)	Bill Street of the	
42 botonMult.grid(row=3,column=4)		
Line 42, Column 31 Tab Size: 4	Pytho	on

Desde el boton4 llamamos a la función numeroPulsado.

Si ejecutamos el proyecto y pulsamos al número 4 este será el resultado:

H -	-		×
			4
7	8	9	1
4	5	6	x
1	2	3	-
0		=	+

Si pulsamos varias veces al número cuatro, solo muestra 1 como lo vamos a solucionar.

El método get() lo que hace es recoger la información que hay en pantalla.

D:\Curso Python\graficos\practicaCalculadora.py • (Curso Python) - Sublime Text (UNREGISTERED)	-		\times
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
practicaCalculadora.py *REPL* [python] ×			▼
16 pantalla.grld(row=1,column=1, paax=10,paay=10, columnspan=4)			
17 pantalla.config(background="black", fg="#03f943", justify="right")		Andreas Andrea	
18		Contraction of the	
19 #Pulsaciones teclado			
20 <i>def</i> numeroPulsado():			
21 numeroPantalla.set(numeroPantalla.get() +"4")			
22			
23			
Line 24, Column 18 Tab Size: -	4	Python	

Si ejecutamos el proyecto y pulsamos varias veces al botón 4 este será el resultado:

88	-		×		
			44444		
7	8	9	1		
4	5	6	x		
1	2	3	-		
0		=	+		

Ahora que tenemos que hacer para que esto funcione con el resto de números.

D:\Curso Python\graficos\practicaCalculadora.py • (Curso Python) - Sublime Text (UNREGISTERED)	_		×	
<u>File Edit Selection Find View G</u> oto <u>T</u> ools <u>P</u> roject Prefere <u>n</u> ces <u>H</u> elp				
✓ ▶ practicaCalculadora.py ●			▼	
16 pantalla.grid(row=1, column=1, paax=10, paay=10, columnspan=4)			1	
1/ pantalla.config(background= black , fg= #03f943 , justify= right)				
10 #Pulsaciones teclado		I BEE		
20 def numeroPulsado(num):				
21 numeroPantalla.set(numeroPantalla.get() +num)		IBS:		
22				
23				
Line 22, Column 1 Tab Size: 4				

La función númeroPulsado(num): le tendremos que pasar un parámetro.

D:\Curso Python\graficos\practicaCalculadora.py • (Curso Python) - Sublime Text (UNREGISTERED)	_		×
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
✓ ▶ practicaCalculadora.py ●			▼
34 #Fila 2			
35 boton4=Button(miFrame, text="4", width=3, command=numeroPulsado("4"))		and the second s	
36 boton4.grid(row=3,column=1)		NAME AND ADDRESS OF TAXABLE PARTY OF TAX	
37 boton5=Button(miFrame, <i>text</i> ="5", <i>width</i> =3)			
38 boton5.grid(row=3,column=2)			
<pre>39 boton6=Button(miFrame, text="6", width=3)</pre>			
40 boton6.grid(row=3,column=3)			
41 botonMult=Button(miFrame, text="x", width=3)			
42 botonMult.grid(row=3,column=4)			
43			
Line 42, Column 31 Tab Size:	4	Pythor	

Desde el boton4 llamaremos a la función pero pasándole el parámetro 4.

El problema que nos vamos a encontrar ahora es que al ejecutar el programa sin haber pulsado el número 4 este ya aparece en pantalla.

.	-		×
			4
7	8	9	1
4	5	6	x
1	2	3	-
0		=	+

Si pulsamos el botón4 este no hace absolutamente nada.

En el momento a una función le agregamos un parámetro este la ejecuta automáticamente sin que nosotros la hubieras llamado.

Para eso tenemos que usar las funciones lambda.



Ahora si ejecutamos el proyecto en pantalla no saldrá ningún número y cuando pulsemos el botón 4 este ya aparecerá las veces que pulsemos el botón.


H -			×	
44444				
7	8	9	1	
4	5	6	x	
1	2	3	-	
0		=	+	

Ahora esta instrucción la copiaremos en todos los botones cambiando a su número.



Una vez agregado el código vamos a ejecutar el proyecto:

B -			×	
01234,56789				
7	8	9	1	
4	5	6	x	
1	2	3	-	
0		=	+	

Verás que le 0 no puede aparecer al principio, esto lo controlaremos en el siguiente capítulo:

Otra opción es que cuando se ejecute la calculadora que muestre el número 0 por defecto.

Cuando le demos a un operador de división, multiplicación, resta o suma que la pantalla se borre para admitir el siguiente conjunto de números.





INTERFACES GRÁFICAS VIII (VIDEO 49)

- Creación de interfaces gráficas
 - Funcionalidad calculadora

Cuando introducimos un valor a la calculadora y le damos al + e introducimos otro valor se borra el valor inicial para introducir el segundo.

Vamos a crear una variable global que pueda ser accesible desde todas las partes del programa y la vamos a llamar operación.

Esta variable global almacenará la operación que queremos realizar, suma, resta, multiplicación o división.



Ahora vamos a crear la función operación.

D:\Curso Python\graficos\practicaCalculadora.py • (Curso Python) - Sublime Text (UNREGISTERED)	_		×
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
✓ ▶ practicaCalculadora.py ●			▼
<pre>19 pantalla.config(background="black", fg="#03f943", justify="right") 20 21 #Pulsaciones teclado</pre>		A set of the set of th	-
<pre>22 def numeroPulsado(num): 23 numeroPantalla.set(numeroPantalla.get() +num) 24</pre>			
<pre>25 #Función suma 26 def suma(): 27 global operacion 28 operacion="suma" 29</pre>			
30 Image: Selected Table	o Size: 4	Python	

Desde el botón suma llamaremos a la función:



En la función suma() definimos la variable global operación:



Ahora queremos que sume para ello vamos a crear una variable llamada resultado y de tipo global.





La llamada de suma hay que pasarle parámetros.



Ahora vamos a comprobar si podemos sumar con la calculadora.

Ahora al pulsar en el botón igual aparezca en pantalla la suma de lo que llevo en este momento.



Ahora vamos a probar la calculadora:

La calculadora suma bien con el igual pero se le damos de nuevo al signo igual este nos vuelve a sumar.



Así al pulsar dos veces al botón igual, en la segunda vez este no sumara.

Prueba la calculadora realizando sumas y comprobando el igual.





INTERFACES GRÁFICAS IX (VIDEO 50)

A continuación el código completo que a aportado el profe:

```
from tkinter import *
raiz=Tk()
miFrame=Frame(raiz)
miFrame.pack()
operacion=""
reset pantalla=False
resultado=0
#-----pantalla-----
numeroPantalla=StringVar()
pantalla=Entry(miFrame, textvariable=numeroPantalla)
pantalla.grid(row=1, column=1, padx=10, pady=10, columnspan=4)
pantalla.config(background="black", fg="#03f943", justify="right")
#-----pulsaciones teclado------pulsaciones teclado------
def numeroPulsado(num):
       global operacion
       global reset_pantalla
       if reset_pantalla!=False:
              numeroPantalla.set(num)
              reset pantalla=False
       else:
              numeroPantalla.set(numeroPantalla.get() + num)
#-----funcion suma-----
def suma(num):
       global operacion
       global resultado
       global reset_pantalla
       resultado+=int(num) #resultado=resultado+int(num)
       operacion="suma"
       reset_pantalla=True
       numeroPantalla.set(resultado)
#-----funcion resta-----
num1=0
contador_resta=0
def resta(num):
       global operacion
       global resultado
       global num1
       global contador_resta
```

```
global reset_pantalla
       if contador_resta==0:
              num1=int(num)
              resultado=num1
       else:
              if contador_resta==1:
                     resultado=num1-int(num)
              else:
                     resultado=int(resultado)-int(num)
              numeroPantalla.set(resultado)
              resultado=numeroPantalla.get()
       contador_resta=contador_resta+1
       operacion="resta"
       reset_pantalla=True
#-----funcion multiplicacion-----
contador_multi=0
def multiplica(num):
       global operacion
       global resultado
       global num1
       global contador_multi
       global reset_pantalla
       if contador_multi==0:
              num1=int(num)
              resultado=num1
       else:
              if contador_multi==1:
                     resultado=num1*int(num)
              else:
                     resultado=int(resultado)*int(num)
              numeroPantalla.set(resultado)
              resultado=numeroPantalla.get()
       contador multi=contador multi+1
       operacion="multiplicacion"
       reset_pantalla=True
#-----funcion division-----
contador_divi=0
def divide(num):
       global operacion
       global resultado
       global num1
       global contador_divi
```

global reset pantalla

```
if contador_divi==0:
```

```
num1=float(num)
              resultado=num1
       else:
              if contador_divi==1:
                     resultado=num1/float(num)
              else:
                     resultado=float(resultado)/float(num)
              numeroPantalla.set(resultado)
              resultado=numeroPantalla.get()
       contador_divi=contador_divi+1
       operacion="division"
       reset_pantalla=True
#-----funcion el_resultado------
def el resultado():
       global resultado
       global operacion
       global contador_resta
       global contador multi
       global contador_divi
       if operacion=="suma":
              numeroPantalla.set(resultado+int(numeroPantalla.get()))
              resultado=0
       elif operacion=="resta":
              numeroPantalla.set(int(resultado)-int(numeroPantalla.get()))
              resultado=0
              contador_resta=0
       elif operacion=="multiplicacion":
              numeroPantalla.set(int(resultado)*int(numeroPantalla.get()))
              resultado=0
              contador multi=0
       elif operacion=="division":
              numeroPantalla.set(int(resultado)/int(numeroPantalla.get()))
              resultado=0
              contador_divi=0
#-----fila 1-----
boton7=Button(miFrame, text="7", width=3, command=lambda:numeroPulsado("7"))
```

boton7.grid(row=2, column=1) boton8=Button(miFrame, text="8", width=3, command=lambda:numeroPulsado("8")) boton8.grid(row=2, column=2) boton9=Button(miFrame, text="9", width=3, command=lambda:numeroPulsado("9")) boton9.grid(row=2, column=3) botonDiv=Button(miFrame, text="/", width=3, command=lambda:divide(numeroPantalla.get())) botonDiv.grid(row=2, column=4) #-----fila 2-----

boton4=Button(miFrame, text="4", width=3, command=lambda:numeroPulsado("4"))
boton4.grid(row=3, column=1)
boton5=Button(miFrame, text="5", width=3, command=lambda:numeroPulsado("5"))
boton5.grid(row=3, column=2)
boton6=Button(miFrame, text="6", width=3, command=lambda:numeroPulsado("6"))
boton6.grid(row=3, column=3)
botonMult=Button(miFrame, text="x", width=3,
command=lambda:multiplica(numeroPantalla.get()))
botonMult.grid(row=3, column=4)

#-----fila 3-----

boton1=Button(miFrame, text="1", width=3, command=lambda:numeroPulsado("1"))
boton1.grid(row=4, column=1)
boton2=Button(miFrame, text="2", width=3, command=lambda:numeroPulsado("2"))
boton2.grid(row=4, column=2)
boton3=Button(miFrame, text="3", width=3, command=lambda:numeroPulsado("3"))
boton3.grid(row=4, column=3)
botonRest=Button(miFrame, text="-", width=3,
command=lambda:resta(numeroPantalla.get()))
botonRest.grid(row=4, column=4)

#------fila 4-----

boton0=Button(miFrame, text="0", width=3, command=lambda:numeroPulsado("0"))
boton0.grid(row=5, column=1)
botonComa=Button(miFrame, text=",", width=3, command=lambda:numeroPulsado("."))
botonComa.grid(row=5, column=2)
botonIgual=Button(miFrame, text="=", width=3, command=lambda:el_resultado())
botonIgual.grid(row=5, column=3)
botonSum=Button(miFrame, text="+", width=3,
command=lambda:suma(numeroPantalla.get()))
botonSum.grid(row=5, column=4)

raiz.mainloop()

- Creación de interfaces gráficas
 - o Botones de radio. Radiobuttons
- Botones de selección para preguntas de respuesta única.



Vamos a crear un nuevo proyecto llamado practicaRadioButton.py



Este será el resultado:



Como podréis comprobara aparecen los dos seleccionados por defecto y no me los deja seleccionar.



Si comprobamos el resultado podrás observar que este funciona.

-ø 🗆 🗙	-ø 🗆 ×	-ø 🗆 🗙
C Masculino	Masculino	C Masculino
C Femenino	O Femenino	Femenino

¿Cómo poder rescatar los valores?



Este será el resultado después de agregar una etiqueta:



D:\Curso Python\graficos\practicaRadioButton.py (Curso Python) - Sublime Text (UNREGISTERED)	_		×
<u>File Edit Selection Find View Goto Tools Project Preferences Help</u>			
✓ ► practicaRadioButton.py ×			▼
1 from tkinter import * 2		NA BUR AND	
3 root=Tk() 4			
5 varOpcion=IntVar()			
<pre>6 7 def imprimir(): 8 9 print(varOpcion.get()) 10</pre>			
11 Label(root, <i>text</i> ="Género:").pack() 12			
<pre>13 Radiobutton(root, text="Masculino", variable=varOpcion, value=1, 14 command=imprimir).pack() < 15</pre>			
<pre>16 Radiobutton(root, text="Femenino", variable=varOpcion, value=2, 17 command=imprimir).pack() 18</pre>	J		
19 root.mainloop <u>()</u>			
Line 19, Column 16 Tab Size: 4		Python	

Vamos a ver que cuando ejecutemos el proyecto va a imprimir en consola el valor de la opción que activemos.



Vamos a crear una nueva etiqueta para poder ver los valores en la propia ventana.



Este será el resultado:



Si queremos agregar más opciones:



Este será el resultado y seleccionamos la tercera opción:

q —		×
	Género:	
0	Masculir	10
0	Femenin	ю
• o	tras opcio	ones
Has elegi	do otras (opciones





INTERFACES GRÁFICAS X (VIDEO 51)

- Creación de interfaces gráficas
 - o Botones de selección. CheckButtons

La diferencia de estos botones casillas de selección con respecto a los botones del capitulo anterior es que puede haber más de uno seleccionado o ninguno, cosa que el anterior no lo permitía.



Vamos a partir de un nuevo proyecto llamado PruebaChecksButtons.py







Vamos a agregar una imagen para recordar capítulos anteriores.



Este será el resultado:



Antes de los botones vamos a poner una etiqueta que diga "Elige destinos:"



Creamos un objeto de imagen, creamos un frame que depende de root y la nueva etiqueta que contiene la imagen la vinculamos con frame en lugar de root, así como los Checkbuttons. Este será el resultado:



Vamos a agregar funcionalidad a los Checkbuttons.



Definimos tres variables.

Creamos una función llamada opcionesViaje(), que controlará las casillas que hemos activado y según lo activado le asignaremos un valor a la variable textoFinal.

A los botones le asignamos las variables, los valores que adquieren cuando están actrivos o no y por último llamamos a la función opcionesViaje.

Este será el resultado si activamos las tres opciones:







INTERFACES GRÁFICAS XI (VIDEO 52)

- Creación de interfaces gráficas

 Menús
- Barra en la parte superior con opciones de sub-menú.



Partiendo de un proyecto nuevo llamado practicaMenu.py



Este será el resultado:



Podemos modificar el ancho de la ventana colocándonos en los extremos y arrastrar o modificar el ancho de la ventana.



Este será el resultado:

🦸 tk		_		×
Archivo	Edición	Herramientas	Ayuda	

Ahora vamos a agregar los elementos de submenú.



Vamos a crear los submenús de Archivo.



Nos colocaremos debajo de archivoMenu y agregaremos los siguientes 5 submenus. Este será el resultado:



Vamos a quitar la línea discontinua que aparece en el submenú.



Tearoff=0, cuando ejecutemos este será el resultado:

🧳 tk		-		x c
Archivo	Edición	Herramientas	Ayuda	
Nuevo Guard Guard	o Iar Iar Como			
Cerraı Salir	r			

Si queremos separar grupo de opciones en el submenú se hace con una barra horizontal continua.

Queremos separar el submenú Guardar como... con Cerrar.



Nos situaremos donde queremos insertar la barra separadora y agregaremos .add_separator(). Este será el resultado:



Vamos a crear los submenús de Edición y Ayuda



Este será el resultado:



En el próximo capítulo ya le daremos inactividad.





INTERFACES GRÁFICAS XII (VIDEO 53)

- Creación de interfaces gráficas
 - Ventanas emergentes



Vamos a continuar con el proyecto anterior.



Tenemos que importar messagebox con la librería tkinder.

Creamos una función llamada infoAdicional().



Que llamaremos desde el submenú "Acerca de..." y llamamos a la función.

Este será el resultado:



Ahora vamos a cambiar el icono que aparecen en la ventana emergente, así como su sonido.

Ahora vamos a crear una ventana de aviso con respeto al submenú licencia.



Creamos una nueva función en este caso messagebox.showwarnig que es un mensaje de aviso.



Llamamos a la función, este será el resultado:



Con la opción del submenú Salir vamos a crear una ventana en la que el usuario podrá confirmar con más de un botón.



Este tipo de messagebox.askquestion le podemos pasar un dato en este caso a la variable valor si seleccionamos el botón si retorna un yes y si seleccionamos el botón no retorna un no.

Por este motivo comparamos si valor es igual a "yes" si es así saldremos de la aplicación, en cambio si seleccionamos el botón se mantendrá en ella.



Llamamos a la función salirAplicación desde el submenú "Salir"

Este será el resultado, primero pulsaremos el botón No y luego al botón Sí.





Si lo cambiamos por messagebox.askokcancel funciona igual lo único que retorna un valor de True o False.

Este será el resultado:



Para el submenú "Cerrar" vamos a crear una ventana que diga Cerrar o Intentar.

Vamos a crear la función.



Desde el submenú correspondiente vamos a llamar a la función.



Ahora cuando ejecutemos la aplicación del submenú Cerrar.

🦸 tk			_		\times
Archivo	Edición	Herramient	tas A	yuda	
	Reinte	entar			×
		No es posibl Documento	le cerra bloque	r. eado.	
	F	Reintentar	Car	ncelar	

Si le damos a Reintentar no saldremos de la aplicación, solo se cerrará la ventana emergente, pero si le damos al botón Cancelar, saldremos de la aplicación.





INTERFACES GRÁFICAS XIII (VIDEO 54)

Creación de interfaces gráficas

 Ventanas emergentes. Abrir archivos.

Vamos a realizar un nuevo proyecto con el nombre Prueba_File_Chooser.py



Si ejecutamos el proyecto este será el resultado:



Seleccionamos el botón Abrir fichero.

Abrir						×
🕆 🗎 > Est	e equipo → Disco local (D:) → pmver → Docum	entos >	`	۹ 5	Buscar en Documentos	
Organizar 🔻 🛛 Nueva ca	rpeta					?
graficos ^	Nombre	Fecha de modifica	Тіро	Tamaño		
modulos	👜 Curso de Python basados en los vídeos d	16/08/2020 13:23	Documento de Mi	24.416 KB		
OneDrive - Persor	😰 Recibos Alquiler.xlsx	06/03/2020 10:36	Hoja de cálculo d	26 KB		
	📧 Copia de Grafica 3d_Pere.xlsx	03/02/2020 19:22	Hoja de cálculo d	241 KB		
💻 Este equipo	📵 Radio.docx	18/10/2019 11:35	Documento de Mi	40 KB		
👆 Descargas	🔁 Problema.docx	15/10/2019 10:36	Documento de Mi	13 KB		
🗄 Documentos	🧰 Diagrama de Flujo.pdf	01/10/2019 19:26	Archivo PDF	73 KB		
Escritorio	🔯 Diagrama de Flujo.vsdx	01/10/2019 19:25	Microsoft Visio Dr	35 KB		
📰 Imágenes	🔯 Prueba Visio.vsdx	01/10/2019 17:06	Microsoft Visio Dr	314 KB		
h Música	📴 Prueba Visio.pdf	01/10/2019 17:06	Archivo PDF	66 KB		
	👜 Verbos.docx	25/06/2019 15:34	Documento de Mi	26 KB		
Objetos 3D	🗐 Summary.docx	17/06/2019 16:13	Documento de Mi	14 KB		
Yídeos	👜 Coordinació a Serveis Socials de Sentme	14/06/2019 16:24	Documento de Mi	14 KB		
🏪 Windows (C:)	🗐 Exportar una hoja de Excel a XML.docx	28/05/2019 16:12	Documento de Mi	741 KB		
🔜 Disco local (D:)	📳 Metadates registres Entrada.xIsx	23/05/2019 20:56	Hoja de cálculo d	20 KB		
A Red V	🔊 Visitas.xlsx	20/05/2019 21:03	Hoja de cálculo d	19 KB		
ineu						_
No <u>m</u> k	ore:					~
					<u>A</u> brir Cancela	r
El print nos devuelve la ruta:

REPL [python] (Curso Python) - Sublime Text (UNREGISTERED)	_	
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp		
▲ ▶ Prueba_File_Chooser.py × *REPL* [python] ×		•
D:/pmver/Documents/Curso de Python basados en los vídeos de PILDORASINFORMATICAS.docx		
Line 2, Column 1	Tab Size: 4	Python

Cuando abrimos la ventana de abrir archivos, este por defecto abre la carpeta Documentos, si queremos especificar un origen diferente:



Al ejecutar el programa y seleccionar el botón este abrirá la ruta especificada.



🖉 Abrir								×
\leftarrow \rightarrow \checkmark \uparrow \checkmark \rightarrow Este eq	quipo 🔸 Disco loca	al (D:) → Curso Pytł	non → graficos		ٽ ~	,	en graficos	
Organizar 👻 Nueva carpe	ta						• •	?
graficos ^ modulos OneDrive - Persor Este equipo Descargas	avion.png	calculadoralCO.i co	calculadoralCO.j pg	EjemploEntry.py	mano.png	manzana.ico	J ok.png	
Documentos		\square	\square	\square	\square	\square	\square	~
No <u>m</u> bre:	[]					<u>A</u> brir	Cancelar	×

Podemos especificar que tipos de archivos nos tiene que mostrar esta ventana.

D:\Curso Python\graficos\Prueba_File_Chooser.py (Curso Python) - Sublime Text (UNREGISTERED)	_		Х
<u>F</u> ile <u>E</u> dit <u>S</u> election Find <u>V</u> iew <u>G</u> oto <u>T</u> ools <u>P</u> roject Prefere <u>n</u> ces <u>H</u> elp			
Prueba_File_Chooser.py ×			▼
<pre>1 from tkinter import * 2 from tkinter import filedialog 3 4</pre>			
5 root=Tk() 6			
7 <i>def</i> abreFichero():			
<pre>8 fichero=filedialog.askopenfilename(title="Abrir",</pre>			
9 initial dir="D: Murso Python raficos",			
10 <i>filetypes=</i> (("Ficheros de Excel","*.xlsl"),("Ficheros de texto","*.txt"))			
11 12 print(fichero) 13			
<pre>14 Button (root, text="Abir fichero", command=abreFichero).pack()</pre>			
15			
16 root.mainloop()			
1/		Duthor	
Tab Size: 4		Fythor	

Al abrir la ventana de búsqueda de ficheros le hemos dicho que solo busque archivos de Excel.

🖉 Abrir					×
← → · · ↑ 🗄 > Este e	equipo > Documentos >		`	ڻ v	
Organizar 🔻 Nueva carp	eta				= • 🔳 😯
🕂 Descargas 🛛 🖈 ^	Nombre	Fecha de modifica	Тіро	Tamaño	
🔮 Documentos 🖈	hp.applications.package.appdata	06/03/2017 15:01	Carpeta de archivos		
📰 Imágenes 🛛 🖈	Grabaciones de sonido	06/03/2017 17:55	Carpeta de archivos		
Curso Python	CyberLink	06/03/2017 20:30	Carpeta de archivos		
Documentos	InstantCDDVD	07/03/2017 10:43	Carpeta de archivos		
graficos	Pinnacle Studio Screen Capture	07/03/2017 11:11	Carpeta de archivos		
modulos	📴 Mis archivos de origen de datos	15/03/2017 14:07	Carpeta de archivos		
	Presentacion access	24/03/2017 11:24	Carpeta de archivos		•
> 🦱 OneDrive - Persor	Microsoft office	02/04/2017 18:41	Carpeta de archivos		
V Erte equipo	📙 Plantillas personalizadas de Office	27/08/2017 19:19	Carpeta de archivos		
	MAGIX_MusicEditor	16/12/2017 17:06	Carpeta de archivos		
> Uescargas	MAGIX Descargas	16/12/2017 17:16	Carpeta de archivos		
Documentos *	— •••••••	17/05/0040 45 07	A 1 1 11		
No <u>m</u> bre	s			~	Ficheros de Excel (*.xlsl) ~
				[<u>A</u> brir Cancelar

Si abrimos el cuadro combinado podremos decirle que busque archivos de texto.

Abrir						×
→ < ↑ ▲ State	e equipo > Documentos >			٩ ٥	Buscar en Documento	IS
ganizar 🔻 🛛 Nueva ca	rpeta				EE 🔻 🔲	. ?
🕹 Descargas 🛛 🖈 🔨	Nombre	Fecha de modifica	Тіро	Tamaño		
🔮 Documentos 🖈	hp.applications.package.appdata	06/03/2017 15:01	Carpeta de archivos			
📰 Imágenes 🛛 🖈	Grabaciones de sonido	06/03/2017 17:55	Carpeta de archivos			
Curso Python	CyberLink	06/03/2017 20:30	Carpeta de archivos			
Documentos	InstantCDDVD	07/03/2017 10:43	Carpeta de archivos			
graficos		07/03/2017 11:11	Carpeta de archivos			
modulos	📴 Mis archivos de origen de datos	15/03/2017 14:07	Carpeta de archivos			
modulos	Presentacion access	24/03/2017 11:24	Carpeta de archivos			
OneDrive - Persor	Microsoft office	02/04/2017 18:41	Carpeta de archivos			
Este equipo	📙 Plantillas personalizadas de Office	27/08/2017 19:19	Carpeta de archivos			
Liste equipo	MAGIX_MusicEditor	16/12/2017 17:06	Carpeta de archivos			
- Descargas	MAGIX Descargas	16/12/2017 17:16	Carpeta de archivos			
Documentos *	· · · · · ·	17/05/0040 45 07	A 11 11			
No <u>m</u> l	ore:			~ Fich	neros de texto (*.txt)	~
					Abrir Canc	elar

D:\Curso Python\graficos\Prueba_File_Chooser.py (Curso Python) - Sublime Text (UNREGISTERED)	_		×
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
Prueba_File_Chooser.py ×			▼
1 from tkinter import *		-	
2 from tkinter import filedialog		Print School of Print	
3			
4 5 reat-Tk()			
6			
7 <i>def</i> abreFichero():			
<pre>8 fichero=filedialog.askopenfilename(title="Abrir",</pre>			
9 initialdir="D:\Curso Python\graficos",			
10 <i>filetypes=</i> (("Ficheros de Excel", "*.xlsl"),("Ficheros de texto", "*.txt")	Þ		
11 ("lodos los ficheros","*.*")))			
13 print(fichero)			
14			
<pre>15 Button (root, text="Abir fichero", command=abreFichero).pack()</pre>			
16			
17 root.mainloop()			
18			
Line 17, Column 16 Tab Size: 4	•	Pytho	n

Ahora queremos que además busque todos los archivos.







BBDD I (VIDEO 55)

- Creación BBDD
 - o Conexión con BBDD
 - o Inserción de registros en BBDD

SGDB y Python

- SQL Server
- Oracle
- MySql
- SQLite
- PostgresSQL
- Etc.



Vamos a ver MySql y SQLite.

¿Qué es SQLite?

- Sistema de gestión de BBDD relacional.
- Escrito en C, es de código abierto.
- Forma parte integral del programa.
- Se guarda como un único fichero en host.



Ventajas

- Ocupa muy poco espacio en disco y memoria.
- Muy eficiente y rápido.
- Multiplataforma.
- Sin administración / Configuración
- Dominio público. Sin costo

Inconvenientes

- Mo admite clausulas anidadas (where)
- No existen usuarios (no acceso simultáneo por parte de varios usuarios a la vez)
- Falta de clave foránea cuando se crea en modo consola.

Pasos a seguir para conectar con BBDD

- 1. Abrir Crear conexión
- 2. Crear puntero
- 3. Ejecutar query (consulta) SQL
- 4. Manejar los resultados de la query (consulta)
 - a. Insertar, Leer, Actualizar, Borrar (Create, Read, Update, Delete)
- 5. Cerrar puntero
- 6. Cerrar conexión

Vamos a empezar un nuevo proyecto llamado Ejemplo1BBDD.py en una nueva carpeta llamada BBDD.



Si ahora ejecutamos el proyecto observaremos que nos ha creado la base de datos.

Ar Fvo Inicio Compartir	Vista				× ^ 😭
Anclar al Acceso rápido	W Mover a ✓ X Eliminar ✓ Copiar a ✓ I Cambiar nombre	Nueva carpeta	Propiedades	Seleccionar te	odo ar nada ción
Portapapeles	Organizar	Nuevo	Abrir	Selecciona	ir
← → × ↑ 📙 > Este equ	iipo → Disco local (D:) → Curso Python	> BBDD		5 V	🔎 Buscar en BBDD
Escritorio	lombre	Fecha de	modificación	Тіро	Tamaño
Imágenes	Ejemplo1BBDD.py	16/08/202	20 18:54	Python File	1 KB
👌 Música	PrimeraBase	16/08/202	20 18:54	Archivo	0 KB
🧊 Objetos 3D 🖕					
2 elementos					

Ahora está completamente vacía.

Vamos a crear nuestra primera tabla.



Guardamos y ejecutamos.

📙 🛃 📕 🖛 BB	DD									_		×
Archivo Inicio	Compartir	Vista										^ ?
Anclar al Acceso rápido	Pegar	🙀 Mover a ▾	X Eliminar ▼ ➡ Cambiar nombre	Nueva carpeta	Propiedades		Select No se	cionar to lecciona ir selecc	odo ir nada ión			
Portapapele	25	0	rganizar	Nuevo	Abrir		Sel	ecciona	r			
← → * ↑ <mark> </mark>	> Este equip	po → Disco loca	II (D:) → Curso Pythor	n ≽ BBDD			~	Ō	<u>,</u>	Busca	r en BBE	D
🔶 Descargas	^ No	ombre	^	Fech	a de modificació	n Ti	ро		Tama	ño		
Documentos	F	Ejemplo1BBDD	.ру	16/08	/2020 19:02	P	ython File			1 KE	3	
Escritorio	~] PrimeraBase		16/08	/2020 19:02	A	rchivo			8 KE	3	
2 elementos								/				

Podemos observar que ahora el fichero ya no está vacío.

Hay una tabla aun que esta está vacía.

Desde la siguiente url: <u>https://sqlitebrowser.org/</u> podremos descargar un visor SQLite.

Desde la pestaña Dowload la podremos descaragar.

Windows

Our latest release (3.12.0) for Windows:

- DB Browser for SQLite Standard installer for 32-bit Windows
- DB Browser for SQLite .zip (no installer) for 32-bit Windows
- DB Browser for SQLite Standard installer for 64-bit Windows
- DB Browser for SQLite .zip (no installer) for 64-bit Windows
- DB Browser for SQLite PortableApp 🛛 🗲

Yo me he descargado el PortableApp.

Una vez descargada la aplicación la guardas en una carpeta y desde la carpeta la instalas.

📙 🛃 📑 🖛 I SC	QLiteDatabase						_		\times
Archivo Inicio	Compartir	Vista							~ ?
Anclar al Acceso rápido	Pegar 🔏	Mover a × Eliminar ×	Nueva carpeta	Propiedade	s 🐼	Seleccionar to	odo ar nada ión		
Portapapel	es	Organizar	Nuevo	Abrii	r	Selecciona	r		
\leftrightarrow \rightarrow \checkmark	> SQLiteDa	atabase >				ٽ ~	,⊂ Bus	car en SQl	.iteD
📌 Acceso rápido	^ N	ombre			Fecha d	e modificación	Тіро		Tamañ
Escritorio	*	SQLiteDatabaseBrowserPortable			16/08/2	020 19:10	Carpeta de	archivos	
🖊 Descargas	*	SQLiteDatabaseBrowserPortable_3.12.)_English.paf.exe	e	16/08/20	020 19:07	Aplicación		24.1
Documentos	×								
📰 Imágenes	* v <								>
2 elementos									:==

Verás que ha creado una carpeta, entramos en ella.

📕 🛃 📕 🖛 SQLiteDatabase	BrowserPortable				– 🗆 ×
Archivo Inicio Compartir	Vista				~ 🕐
Anclar al Acceso rápido	Mover a - Eliminar -	Nueva carpeta	Propiedades	Seleccionar to	odo ir nada ión
Portapapeles	Organizar	Nuevo	Abrir	Selecciona	r
← → × ↑ 🔤 → SQLiteDa	tabase > SQLiteDatabaseBrowserPorta	ble >		5 V	
▲ No	ombre	Fecha d	e modificación	Тіро	Tamaño
Escritorio	Арр	16/08/20	020 19:10	Carpeta de archivos	
	Data	16/08/20	020 19:11	Carpeta de archivos	
🔶 Descargas 🚿	Other	16/08/20	020 19:10	Carpeta de archivos	
🖆 Documentos 🖈 🛛 📄	help.html	18/06/20	020 4:17	Archivo HTML	5 KB
📰 Imágenes 🖈 🛛	SQLiteDatabaseBrowserPortable.exe	18/06/20	020 4:18	Aplicación	173 KB
Curso Python V 5 elementos	-			-	

Ejecutamos SQLiteDatabaseBrowserPortable.exe.

DB Browser for SQLite - D:\Curso Python\BBDD\PrimeraBase Archivo Editar Ver Herramientas Ayuda		-	- 🗆 X
Nueva base de datos 🛛 🔒 Abrir base de datos 🖕 🕞 Guardar camb	pios 🔯 Deshacer cambios	s 🎲 Abrir proyecto 😭 Guardar proyecto 🎧 Anexar base de datos	s »
Estructura Hoja de datos Editar pragmas Ejecutar SQL		Editar celda	ā ×
🕞 Crear tabla 🛛 🖓 Crear índice 🚔 Imprimir		Modo: Texto 🗸 🎲	
Nombre Tipo	Esquema	1	
✓ ■ PRODUCTOS	CREATE TABLE PRO		
NOMBRE_ARTICULO VARCHAR(50) "NOMBRE_ARTICUL		
PRECIO INTEGER	"PRECIO" INTEGER		
SECCION VARCHAR(20) "SECCION" VARCH	lipo de datos actualmente en la celda Tamaño de los datos actualmente en la tabla	Aplicar
Índices (0)		Tamano de los datos actualmente en la tabla	
Vistas (0)		Remoto	ē ×
L Disparadores (0)		Identidad Pública	B
			-342
		Nombre Consolidación Ultima modificac Iamaño	
<	>	Historial de SQL Gráfica Esquema Remoto	
			UTF-8

Si abrimos una base de datos podremos ver la estructura.

También podemos arrastras la base de datos al visor, esta te la mostrará automáticamente.

Con la	pestaña	Hoja de	e datos	podremos	ver los	registros.

DB Browser for SQLite - D:\Curso Python\BBDD\PrimeraBase	- 🗆 X
<u>A</u> rchivo <u>E</u> ditar <u>V</u> er <u>H</u> erramientas Ay <u>u</u> da	
Rueva base de datos Abrir base de datos Regular da cambios Deshacer cambios	s 🍘 Abrir proyecto 🕼 Guardar proyecto 🖓 Anexar base de datos »
Estructura Hoja de datos Editar pragmas Ejecutar SQL	Editar celda 🗗 🛪
Iabla: IPRODUCTOS V 😵 🗞 🗞 🗮 🖨 🗟 » Fiterina	Modo: Texto 🗸 🎯
NOMBRE_ARTICULO PRECIO SECCION Filtro Filtro Filtro	1 Tipo de datos actualmente en la celda Tamaño de los datos actualmente en la tabla Remoto 6" ×
I 4 0 -0 de 0 ▶ ▶ Ir a: 1	Identidad Pública V 🛞 🕼 Nombre Consolidación Última modificac Tamaño
	UTF-8 ":

¿Cómo insertamos información en la tabla?

Lo primero que tenemos que realizar es eliminar o comentar la línea de crear tabla, ya que si no lo hiciéramos, nos daría error diciendo que dicha tabla ya existe.



En miConexion.execute insertamos los artículos, pero para que esto tenga efecto hemos de confirmar con miConexion.commit().

Guardamos y ejecutamos.

REPL [python] (Curso Python) - Sublime Text (UNREGISTERED)										×				
<u>F</u> ile	<u>E</u> dit	<u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ce	s <u>H</u> elp					
••					*RE	EPL* (pyt	hon]	×						▼
***	Repl	Closed*	**											
	Line 3	3, Column 1									Tab	Size: 4	Pyth	on

Podrás observar que no ha pasado nada, pero si consultamos la base de datos con el visor, este será el resultado:

Arch)B Browser for SQLite - ivo Editar Ver He	D:\Curso F	ython\BBD s Avuda	D\Prime	eraBase										-			×
	Nueva base de datos	🔒 Abrir ba	ase de datos		Guardar	cambios	(i)	Deshacer cambio	s 🎲 Abrir pr	oyecto	😭 Guardar pro	oyecto	🗟 Ane	exar bas	e de datos			>>
Estr	uctura Hoja de datos	Editar	pragmas	Ejecuta	r SQL				Editar celda									8×
<u>T</u> abla	PRODUCTOS	~	3 🔏	•	L 🖨	B		» Filter in a	Modo: Texto	\sim				Ē,	🗈 🗔	æ	•	
E 1 B	NOMBRE_ARTICULO Filtro	PRECIO Filtro 15	SECCI Filtro DEPORTES	ON 5					1									
									Tipo de datos a 1 carácter	tualmen	te en la celda: Tex	to / Numério	:0				Aplici	ar
									Remoto									8×
									Identidad Pú	blica	× 🙀							
									Nombre	(Consolidación	Última m	odificad	i Tama	nõ			
M	🖣 1 - 1 de 1 🕨 🕅				Ir a:	1			Historial de SQL	Grá	ifica Esquema	Remoto	,					
												_					U	IF-8

Nos confirma que hemos agregado un registro.





BBDD II (VIDEO 56)

- Inserción de varios registros.
- Recuperación de varios registros.

Vamos a reutilizar el código del proyecto anterior.



Para insertar varios registros a la vez, lo primero que vamos a realizar es una variable de tipo tupla, donde especificaremos tres registros.

Para añadir varios registros utilizaremos miCursor.executemany con la sintaxis que se muestra en el código, (?,?,?) cada interrogante por campo que tiene la tabla, y por último el nombre de la variable de tipo tupla.

Guardamos y ejecutamos, veremos que no ha realizado nada pero ahora con el visor SQLite vamos a comprobar si se han agregado dichos registros.

BB Browser for SQLite - D:\Curso Python\BBDD\PrimeraBase	-	
<u>A</u> rchivo <u>E</u> ditar <u>V</u> er <u>H</u> erramientas Ay <u>u</u> da		
Reverse asse de datos 🔹 Abrir base de datos 🚬 🕞 Guardar cambios 🕼 Deshacer cambios 🚱 Abrir proyecto 😭 Guardar proyecto	xar base de datos	»
Estructura Hoja de datos Editar pragmas Ejecutar SQL Editar celda		₽×
Iabla: 📑 PRODUCTOS 🗸 😵 💊 🔩 🖨 📾 💀 Fitter in a Modo: Texto 🗸 🎯	🖹 🖬 📮	a 🕞 🖨
NOMBRE_ARTICULO PRECIO SECCION		
Filtro Filtro		
1 BALÓN 15 DEPORTES		
2 Camiseta 10 Deportes		
3 Jarón 40 Cerámica Tipo de datos actualmente en la celda: Texto / Numérico		Aplicar
4 Camión 20 Juguetería		
Remoto		Ð ×
Identidad Pública 🗸 🔯		
Nombre Consolidación Última modificac	Tamaño	
I - 4 de 4 I		
Historial de SQL Grafica Esquema Remoto		

Observamos que así ha sido.

Ahora la pregunta es como accedemos a esta información.



Recuerda Ctrol + S guardar y Ctrol + B ejecutar en la consola.



En este ejemplo se muestra cada registro en una fila distinta.

Ahora solo queremos consultar por el primer campo:



Para poner el nombre del campo junto con el artículo:



En el próximo capitulo trataremos las claves principales.





BBDD III (VIDEO 57)

• Claves principales en BBDD



La primera columna los valores no pueden ser repetido ya que es un campo clave.

DB Browser for SQLite - D:\Curso Python\BBDD\Gest	ionProductos		-		×
<u>Archivo</u> <u>Editar</u> <u>V</u> er <u>H</u> erramientas Ayuda					
Rueva base de datos Abrir base de datos 🖉 🕼	Guardar cambios	Deshacer cambios	Abrir proyecto		>>
Estructura Hoja de datos Editar pragmas Ejecut	ar SQL		Editar celda Carga una sesión de trabajo de un archivo		ð ×
Crear tabla 💊 Crear índice 🔝 Modificar tabla	Borrar tabla	Imprimir	Modo: Texto 🗸 🔯	0	
Nombre	Тіро	Esquema			_
✓ ☐ Tablas (1)					
✓		CREATE TABLE PROI			
Digo_Articulo	VARCHAR(4)	"CODIGO_ARTICULO			
NOMBRE_ARTICULO	VARCHAR(50)	"NOMBRE_ARTICUL			
PRECIO	INTEGER	"PRECIO" INTEGER	Tipo de datos actualmente en la celda: Texto / Numérico	Aplica	ar
SECCION	VARCHAR(20)	"SECCION" VARCHA	1 carácter		
📎 Índices (0)			Remoto		đΧ
Vistas (0)					
Disparadores (0)			Identidad Pública V		3
			Nombre Consolidación Última modificac Tamaño		
<		>	Historial de SQL Gráfica Esquema Remoto		
Carga una sesión de trabajo de un archivo				UT	F-8

Observarás una pequeña llave con CODIGO_ARTICULO, nos indica que es un campo clave.

Si vamos a Hoja de datos veremos los artículos:

Nueva base de d	atos 🛛 🙀 Abrir base de datos	Gua	rdar cambios	Deshacer cambios Abrir proyecto	
Estructura Hoja	de datos Editar pragmas E	ijecutar SQ	L	Editar celda	é
abla: PRODUCT	os 🗸 🔁 🔏 😓	E i		📑 🔋 Filter in a 🛛 Modo: Texto 🗸 🎯 📄 🚍 🕞 🖓	
CODIGO_ARTI	CULO NOMBRE_ARTICULO	PRECIO	SECCION		
Filtro	Filtro	Filtro	Filtro		
AR01	Pelota	20	Juguetería		
AR02	Pantalón	15	Confección		
AR03	Destornillador	25	Ferretería	Tipo de datos actualmente en la celda: Texto / Numérico	Aplicar
AR04	Jarrón	45	Cerámica	1 caracter	
				Remoto	é
				Identidad Pública 🗸 🎲	E.
				Nombre Consolidación Última modificaci Tamaño	_
		Te			

Veremos los artículos que hemos añadido.

Ahora vamos a agregar un nuevo registro.



Vamos a comprobar con el visor si hemos agregado este registro.

Nueva base de datos	Abrir base de datos	Gua	rdar cambios	Deshacer cambios	Abrir proyecto	😭 Guardar pro	yecto 🔒 Ane:	xar base de datos			
Estructura Hoja de dato	s Editar pragmas E	jecutar SQI	-		Editar celda					Ð	
abla: 🔲 PRODUCTOS	- 🛿 🔏 🔹	E a		» Filter in a	Modo: Texto 🗸	**		🖹 🖪 🗔		•	
CODIGO_ARTICULO	NOMBRE_ARTICULO	PRECIO	SECCION		1						
Filtro	Filtro	Filtro	Filtro		± •						
AR01	Pelota	20	Juguetería								
AR02	Pantalón	15	Confección								
3 AR03 Destornillador 25 Ferretería					Tipo de datos actualmente en la celda: Texto / Numérico					Aplicar	
AR04	Jarrón	45	Cerámica		1 carácter						
5 AR05	Tren	15	Jugueteria		Remoto					Ð	
					Identidad Pública	~ 🐞				3	
					Nombre (Consolidación	Última modificad	Tamaño			
					Nonibie (consolidación	olama modificad	amano			
	1	-									
N 1 - 5 de 5 🕨 🕅		Ir	a: 1		Historial de SQL Grá	ifica Esquema	Remoto				

Así es hemos agregado el artículo con código AR05.

Si por error repetimos el mismo código esto nos generará un error.



sqlite3.IntegrityError: UNIQUE constraint failed: PRODUCTOS.CODIGO_ARTICULO

Porque ya existe un registro con este campo clave, no se puede repetir.

Este registro no ha sido agregado a la BBDD.

Vamos a borrar la base de datos para realizar otro proyecto y que el campo clave se inserte automáticamente sin la necesidad de ponerlo nosotros y evitar campos clave repetidos.



Para que se ingrese el campo clave automáticamente al crear la tabla escribiemos: ID *INTEGER* PRIMARY KEY AUTOINCREMENT

En la línea 22 cambiamos el primer ? por NUL de esta manera le estamos diciendo que lo introduzca automáticamente.

	🗊 DB Browser for SQLite - D:\Curso Python\BBDD\GestionProductos — 🗆 X											×
	Nueva	base de datos 🙀 Abri	r base de d	latos 🛓 庙 🤅	Guardar cambios	Deshacer cambios	Abrir proyecto	Guardar proyecto	Anexa	ar base de datos		>>
E	tructura	Hoja de datos Edi	tar pragma	s Ejecutar	SQL		Editar celda					đΧ
Ta	la: 🔲	PRODUCTOS ~	2 7	. 🕹 🖳		» Filter in a	Modo: Texto 🗸	1		🖹 🖪 🗔		
	ID	NOMBRE_ARTICULO	PRECIO	SECCION			1 1					
	Filtro	Filtro	Filtro	Filtro			± •					
1	1	Pelota	20	Juguetería								
2	2	Pantalón	15	Confección								
3	3	Destornillador	25	Ferretería			Tipo de datos actualmen	te en la celda: Texto / Nu	mérico		Ap	licar
4	4	Jarrón	45	Cerámica			1 caracter					
							Remoto					₽×
							Identidad Pública	~ 🙀				3
							Nombre (Consolidación Últir	na modificaci	Tamaño		
k	1	- 4 de 4 🕨 🕅			Ira: 1							
							Historial de SQL Grá	ifica Esquema Re	moto			
												UTE-8

Observamos que ha creado el campo ID y ha número automáticamente.

DR Prowner for SOLite - Di) Curro		anBraductor		_		×
b browser for SQLife - D:\Curson	- yulon(bbDD/desu	onrioductos			-	~
<u>Archivo</u> <u>Editar</u> <u>V</u> er <u>H</u> erramienta	is Ay <u>u</u> da					
🐻 Nueva base de datos 🛛 🗟 Abrir b	ase de datos 🖕 🕼	Guardar cambios 🕼 Deshacer cambios	Abrir proyecto 😭 Guardar proyecto			>>
Estructura Hoja de datos Editar	pragmas Ejecuta	r SQL	Editar celda			ē ×
🕞 Crear tabla 🛛 🗞 Crear índice	Modificar tabla	Borrar tabla	Modo: Texto 🗸 🔯	a	6	
Nombre V III Tablas (2)	Тіро	Esquema	1			
✓ ■ PRODUCTOS		CREATE TABLE PRODUCTOS (ID INTEGER PR				
🚽 🔜 ID	INTEGER	"ID" INTEGER				
	VARCHAR(50)	"NOMBRE_ARTICULO" VARCHAR(50)				
PRECIO	INTEGER	"PRECIO" INTEGER	Tipo de datos actualmente en la celda: Texto / Numerico		Aplic	ar
SECCION	VARCHAR(20)	"SECCION" VARCHAR(20)	1 caracter			
> sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)	Remoto			đΧ
📎 Índices (0)						
Vistas (0)			Identidad Pública 🗸 🎲			3
Disparadores (0)						
¢		,	Hotelido 501 Crófen Engune Regult			
			riistonai ue sou granca Esquerra Remoto			-
					U	IF-8 .

Si observamos la estructura veremos que es campo clave ID INTEGER.



CURSO PYTHON	57
BBDD III	Con la

BBDD IV (VIDEO 58)

- Cláusula UNIQUE
- Operaciones CRUD

La cláusula UNIQUE nos va a impedir repetir la información en un campo en toda la tabla.

Supongamos que no podemos tener dos artículos con el mismo nombre.

Borramos la base de datos, para que no nos diga que dicha base de datos existe.



Le estamos diciendo que no duplique ningún registro con el mismo NOMBRE_ARTICULO.

Ejecutamos el proyecto, verás que no ha hecho nada, pero hemos creado de nuevo el fichero "GestionProductos".

Lo vamos a leer con el lector SQLite.

Nombre	Тіро	Esquema
✓ III Tablas (2)		
V III PRODUCTOS		CREATE TABLE PRODUCTOS (ID INTEGER PRIMARY
D ID	INTEGER	"ID" INTEGER
NOMBRE_ARTICULO	VARCHAR(50)	"NOMBRE_ARTICULO" VARCHAR(50) UNIQUE
PRECIO	INTEGER	"PRECIO" INTEGER
SECCION	VARCHAR(20)	"SECCION" VARCHAR(20)
> 📰 sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)
📎 Índices (0)		
📕 Vistas (0)		•
🔲 Disparadores (0)		

Observamos como esta campo es UNIQUE y ya no lo podremos duplicar.

Vamos a borrar de nuevo la base de datos.



Hemos duplicado "Pantalón" en principio esto tiene que dar error.



Nos está diciendo que el campo NOMBRE_ARTICULO es UNIQUE.

Ha creado la base de datos pero está vacía.

DB Browser for SQLite - D:\Curso Python\BBDD\GestionProductos	- 🗆 X
<u>Archivo E</u> ditar <u>V</u> er <u>H</u> erramientas Ay <u>u</u> da	
Rueva base de datos Abrir base de datos Gardar cambios	🍘 Abrir proyecto 🕼 Guardar proyecto 🖓 Anexar base de datos 🗙 Cerrar base de datos
Estructura Hoja de datos Editar pragmas Ejecutar SQL	Editar celda & X
Tabla: 🔲 PRODUCTOS 🗸 💈 🔏 🔩 🖳 🔒 🦉 » Filter in a	ny column Modo: Texto 🗸 🎇
ID NOMBRE_ARTICULO PRECI0 SECCION Filtro Filtro Filtro Filtro	1
	Tipo de datos actualmente en la celda Aplicar
	Tamano de los datos actualmente en la tabla
	Remoto 🗗 🛪
	Identidad Pública 🗸 🔯
	Nombre Consolidación Última modificac Tamaño
Image: A state of the	Historial de SOL Gráfica Esquema Remoto
	UTF-8

Borramos de nuevo la base de datos.



Si un artículo se llama Pantalón y otro Pantalones ya son datos diferentes.

Ejecutamos el proyecto y vemos el contenido de la tabla.

_																				
	DB Br	owser for SQLite - D:\Cur	so Python	BBDD\Gest	tionProd	luctos												—		\times
A	chivo	<u>E</u> ditar <u>V</u> er <u>H</u> erramie	ntas Ay	<u>u</u> da																
(Nueva	a base de datos 🛛 🔒 Abr	ir base de d	latos 🖕 🖟	Guarda	ar cambios	Besha	acer cambios	ᢙ Abri	r proyecto	Guar	rdar proyecto	Q	Anexar	oase de d	latos	🗙 Ce	errar bas	e de dato	s
E	structur	a Hoja de datos Ed	itar pragma	s Ejecut	ar SQL					Editar ce	elda									đΧ
Ī	bla:	PRODUCTOS ~	2 7	-	ą a		3	» Filter in a	any column	Modo:	Texto	~								
Г	ID	NOMBRE_ARTICULO	PRECIO	SECCION	N															
	Filtro	Filtro	Filtro	Filtro						1	1									
1	1	Pelota	20	Jugueterí	a															
2	2	Pantalón	15	Confecció	ón															
3	3	B Destornillador	25	Ferretería	а					Tipo de	datos actua	almente en la	celda: Tex	to / Num	érico				Ap	licar
4	4	Jarrón	45	Cerámica	1					1 caráct	er									
5	5	6 Pantalones	35	Confecció	ón					Remoto										đΧ
										Identi	dad Públic	:a 🗸 🎲	1							
										Nom	nbre	Consoli	dación	Última	modific	aci Tar	maño			
		1 - 5 de 5 🕨 🕅				Ir a:	1			Historia	l de SOI	Gráfica	Ecquema	Dom	oto					
-										ristoria	TUE SQL	Granta	Laquenia	Rem	010					

Ha agregado los 5 registros.

Operaciones CRUD.

Queremos consultar aquellos registros cuya SECCION='Confección' usaremos el parámetro WHERE, cuando ejecutemos verás que hay dos registros.



UPDATE= Actualización de registros.

Queremos actualizar el precio de la Pelota que tiene un precio de 20 € y queremos que tenga un precio de 35 €.



Ejecutamos y consultamos la base de datos.

Arc	DB Bro hivo	owser for SQLite - D:\Curs Editar <u>V</u> er <u>H</u> erramie	o Python	.BBDD\Gestion	nProductos						_		×
6	Nueva	base de datos 🛛 🔒 Abri	r base de d	latos 🕌 🗎	Guardar cambios	Desha Desh	cer cambios	r provecto 🕋 Gua	rdar proyecto	Anexar base de datos	G 🗙 Cerrar bas	e de datos	
Es	tructura	Hoja de datos Edi	tar pragma	s Ejecutar	SQL			Editar celda					đΧ
Tal	la:	PRODUCTOS ~	2 7	•		3 4	» Filter in any column	Modo: Texto	~		🖹 🖬 📮		
	ID	NOMBRE_ARTICULO	PRECIO	SECCION				1 1					_
	Filtro	Filtro	Filtro	Filtro				± 🖬					
1	1	Pelota	35	Juguetería	-								
2	2	Pantalón	15	Confección									
3	3	Destornillador	25	Ferretería				Tipo de datos actu	almente en la celda: 1	Fexto / Numérico		Aplic	tar
4	4	Jarrón	45	Cerámica				1 carácter					
5	5	Pantalones	35	Confección				Remoto					đΧ
								Identidad Públic	ca 🗸 🎲				
								Nombre	Consolidación	Última modificac	Tamaño		
		L - 5 de 5 🕨 🕅			Ir a:	1							
								Historial de SQL	Gráfica Esquer	ma Remoto			
Des	hace lo	os cambios al último estac	lo guarda	do								U	TF-8

El precio de la Pelota ya es de 35 €.

Ahora vamos a borrar registros utilizaremos DELETE.



Guardamos, ejecutamos y consultamos la base de datos con el visor.

	DB Br	owser for SQLite - D:\Curs	o Python	BBDD\Gestion	Productos										-		×
An	hivo	<u>E</u> ditar <u>V</u> er <u>H</u> erramie	ntas Ay	<u>u</u> da													
	Nueva	base de datos 🛛 🔒 Abri	r base de d	latos 🖕 🕞 🕞	Guardar cambios	🖗 Deshao	er cambios	ᢙ Abrir	proyecto	🏫 Guardar p	royecto	Anexar base	e de datos	🗙 Ce	rrar base	de datos	
E	tructura	Hoja de datos Edi	tar pragma	s Ejecutar S	SQL				Editar cel	lda							đΧ
Īa	ola:	PRODUCTOS ~	2 7	🔹 🖳		3 4 4	» Filter in a	ny column	Modo: T	Texto 🗸	```		· ·	L 🖪	a		
	ID	NOMBRE_ARTICULO	PRECIO	SECCION													
	Filtro	Filtro	Filtro	Filtro					1 1	1							
1	1	Pelota	35	Juguetería													
2	2	Pantalón	15	Confección													
3	3	Destornillador	25	Ferretería					Tipo de da	latos actualmen	te en la celda: Te	exto / Numéric	0			Aplic	car
4	4	Jarrón	45	Cerámica					1 caracte	er .							
									Remoto								đΧ
									Identida	ad Pública	~						
									Nomb	bre (Consolidación	Última m	odificaci Ta	maño			
ŀ	◀	l - 4 de 4 🕨 🕅			Ir a:	1			Historial	de SOL Grá	ifica Esquem	a Pemoto					
									- instorial t	00040 010	Laquem	Remoto				U	TE-8

El registro número 5 ya se ha eliminado.





PRACTICA GUIADA I (VIDEO 59)

• Aplicación gráfica CRUD

Vamos a realizar el siguiente proyecto paso a paso.

🖉 tk		-		\times
BBDD Borrar	CRUD	Ayuda		
ld:				
Nombre				
Password:				
Apellido:				
Dirección:				
Comentarios:				- ^
Create R	ead	Update	De	slete

El video explica como tiene que ser el proyecto.





PRÁCTICA GUIADA II (VIDEO 60)

Vamos a crear una nueva carpeta a la que le llamaremos Práctica guiada.

Vamos a crear un nuevo proyecto llamado CRUD guiado.py

ا 🗾	D:\Curso Python\Practica guiada\CRUD guiado.py (Curso Python) - Sublime Text (UNRE –											
<u>F</u> ile	<u>E</u> dit	<u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp			
▲ ►		CRUD guiade	o.py	×								▼
1		from tki	nter	impo								
2		from tki	nter	impor	rt m	essag	ebox					
3		import s	qlite	23								
4												
5		root=Tk()									
6												
7												
8												
9												
10												
11		root.mai	nloop	D()								
	Line	11, Column 1	6						Tab Size: 4	4	Python	

Importamos las librerías y definimos la ventana gráfica.

—		\times
	_	

Vamos a empezar por el menú de la parte superior.



Vamos a crear las cajas de textos:



Este será el resultado:

Ø +	_		×
BBDD	Borrar	CRUD	
Ayuda			
			-
			-
			-
			-
			-

El siguiente paso será agregar los Label.





PRACTICA GUIADA III (VIDEO 61)

Vamos a crear las etiquetas:

🗾 D	D:\Curso Python\Practica guiada\CRUD guiado.py (Curso Python) - Sublime Text (UNREGISTERED)	_		\times
<u>F</u> ile	<u>E</u> dit <u>S</u> election F <u>i</u> nd <u>V</u> iew <u>G</u> oto <u>T</u> ools <u>P</u> roject Prefere <u>n</u> ces <u>H</u> elp			
••	CRUD guiado.py ×			▼
54 55	<pre>4 textoComentario=Text(miFrame, width=16, height=5) 5 textoComentario_grid(row=5, column=1, nady=10, nady=10)</pre>			
56	6 scrollVert=Scrollbar(miFrame, <i>command</i> =textoComentario.yview)			
57	7 scrollVert.grid(<i>row=</i> 5, <i>column=2</i> , <i>sticky=</i> "nsew")			
58	<pre>8 textoComentario.config(yscrollcommand=scrollVert.set)</pre>		III Ginarar	
59	9			_
60	ð #Aquí comienzan las etiquetas			
61 62	μ) idlaha]=laha]/mi[namo_tayt="Id:")			
63	idlabel grid(row=0 column=0 sticky="e" nady=10 nady=10)			
64	4			
65	5 nombreLabel=Label(miFrame, <i>text</i> ="Nombre:")			
66	6 nombreLabel.grid(row=1, column=0, sticky="e", padx=10, pady=10)			
67	7			
68	<pre>gassLabel=Label(miFrame, text="Password:")</pre>			
69	9 passLabel.grid(row=2, column=0, sticky="e", padx=10, pady=10)			
70 71	/ 1 apollidalahol-lahol(miEnamo taxt="Apollida:")			
71	2 apellidolabel_cabel(mirrame, text- Apellido;)			
73	3			
74	4 direccionLabel=Label(miFrame, <i>text</i> ="Dirección:")			
75	5 direccionLabel.grid(row=4, column=0, sticky="e", padx=10, pady=10))		
76	5			
77	<pre>7 comentariosLabel=Label(miFrame, text="Comentarios:")</pre>	•		
78	8 comentariosLabel.grid(<i>row</i> =5, <i>column=</i> 0, <i>sticky</i> ="ne", <i>padx=</i> 10, <i>pady=</i>	=10)		
/9	post mainloon()		J .	
00 81	1			
01				
_				
	Line 80, Column 16 Tab Size: 4		Python	

Este será el resultado:

🧳 tk			_	×
BBDD	Borrar	CRUD	Ayuda	
	ld:			
N	lombre:			
Pa	ssword:			
۵	pellido:			 -
Di	rección:			
Come	entarios:			^
				~

Ahora vamos a crear los botones inferiores:

D:\Curso Python\Practica guiada\CRUD guiado.py • (Curso Python) - Sublime Text (UNREGISTERED)	_		×
<u>F</u> ile <u>E</u> dit <u>S</u> election Find <u>V</u> iew <u>G</u> oto <u>T</u> ools <u>P</u> roject Prefere <u>n</u> ces <u>H</u> elp			
CRUD guiado.py			▼
76		1.0	
<pre>77 comentariosLabel=Label(miFrame, text="Comentarios:")</pre>			
78 comentariosLabel.grid(row=5, column=0, sticky="ne", padx=10, pady=	10)		
79			
80 #Aquí los botones		SEL COnser	
82 miFrame2=Frame(root)			
83 mlFrame2.pack()			
04 25 hoton(noon_Dutton(mi[nomo] tout-"(nooto")			
86 boton(nean_gaid(now=1_column=0_sticky="e" nady=10_nady=10)		BELLEVILLE ACTION	
87			
88 botonleer=Button(miFrame2. text="Read")			
89 botonLeer.grid(row=1, column=1, sticky="e", padx=10, pady=10)		_	
90			
91 botonActualizar=Button(miFrame2, <i>text</i> ="Update")			
92 botonActualizar.grid(row=1, column=2, sticky="e", padx=10, pady=10)		
93			
94 botonBorrar=Button(miFrame2, <i>text</i> ="Delete")			
<pre>95 botonBorrar.grid(row=1, column=3, sticky="e", padx=10, pady=10)</pre>			
96			
9/			
98 root.mainicop()			
99			
Line 98, Column 16 Tab Siz	e: 4	Python	

Este será el resultado:

🧳 tk			_		×
BBDD	Borrar	CRUD	Ayuda		
	ld:				-
N	lombre:				
Pa	ssword:				
4	pellido:				
Di	rección:				
Come	entarios:				
Crea	te F	Read	Update	Dele	ete

En el próximo capítulo de daremos funcionalidad al submenú Conectar de la barra superior apartado BBD.





PRÁCTICA GUIADA (VIDEO 62)

- Conexión a BBDD
- Salir de la aplicación

Vamos a crear la siguiente función:



Ahora tenemos que llamar a la función:



Ahora vamos a ejecutar el programa.

🧳 tk			_	Х
BBDD	Borrar	CRUD	Ayuda	
Con	ectar			
Salir	r			
N	lombre:			
Pa	ssword			

Seleccionaremos Conectar.

Ø BBDD	Х
BBDD creada con éxito	
Aceptar]

Si le danos de nuevo nos tiene que decir que esta base de datos ya existe.



Ahora con el visor SQLite vamos a ver si ha creado la correspondiente tabla.

B Browser for SQLite - D:\Curso Python\Practica g	uiada\Usuarios		- D X	
<u>Archivo</u> <u>Editar</u> <u>Ver</u> <u>H</u> erramientas Ayuda				
Rueva base de datos → Abrir base de datos	Guardar cambios	Deshacer cambios	ir proyecto 🕼 Guardar proyecto 😹 Anexar base de datos 🗙 Cerrar base de datos	
Estructura Hoja de datos Editar pragmas Ejecu	tar SQL		Editar celda 🖉 :	×
Crear tabla Crear índice Modificar tabla	Borrar tabla	Imprimir	Modo: Texto 🗸 🎯	
Nombre V Tablas (2)	Тіро	Esquema	1	
V DATOSUSUARIOS		CREATE TABLE DATOSUSUARIOS(II		
D ID	INTEGER	"ID" INTEGER		
	VARCHAR(50)	"NOMBRE_USURIO" VARCHAR(50)		
PASSWORD	VARCHAR(50)	"PASSWORD" VARCHAR(50)	Tipo de datos actualmente en la celda Aplicar	
APELLIDO	VARCHAR(10)	"APELLIDO" VARCHAR(10)	Tamaño de los datos actualmente en la tabla	
DIRECCION	VARCHAR(50)	"DIRECCION" VARCHAR(50)	Remoto 🗗 :	×
COMENTARIOS	VARCHAR(100)	"COMENTARIOS" VARCHAR(100)		
> sqlite_sequence		CREATE TABLE sqlite_sequence(na	Identidad Pública 🗸 🙀	
📎 Índices (0)			Nambra Consellation últime un défend Terre és	
🔳 Vistas (0)			Nombre Consolidación Ultima modificac lamano	
Disparadores (0)				
<		>	Historial de SQL Gráfica Esquema Remoto	-
			LITE O	
			011-0	
Se a creado la tabla correctamente.

Vamos a crear la función para salir de la aplicación.



Realizamos la siguiente función salirAplicacion().



Llamamos a la función, guardamos y ejecutamos.

🧳 tk			_	×
BBDD	Borrar	CRUD	Ayuda	
Conectar				-
Salii				
N	lombre:			-
Pa	ssword:			



Le decimos que no.

Volvemos a ejecutar Salir y en este caso le contestamos que sí.





PRACTICA GUIADA V (VIDEO 63)

- Limpiar campos
- Insertar registros



Ahora vamos a crear la función limpiarCampos().





Guardamos y ejecutamos:

🧳 tk			_		×
BBDD	Borrar	CRUD	Ayuda		
	ld:	001			
N	lombre:		F	ere Ma	anel
Pa	ssword:	????	?		
4	pellido:	Verd	lugo Zar	nora	_
Di	rección:	C/ N	Aolino, 7	'	
Come	entarios:	Est el (Pyt)	oy hac curso hon	iend de	•
Crea	teR	ead	Update	•	Delete

Ahora del menú Borrar seleccionaremos Borrar campos.

🧳 tk			_		\times				
BBDD	Borrar	CRUD	Ayuda						
	Borrar campos								
N	lombre:		Pere Manel						
Pa	ssword:	????	?????						

Se borran todos los campos.

🧳 tk			_		×
BBDD	Borrar	CRUD	Ayuda		
	ld:				
N	lombre:				
Pa	ssword:				
A	pellido:				
Di	rección:				
Come	ntarios:				< < >
Crea	te F	Read	Update	Dele	te

Ahora vamos a crear otra función para agregar los registros en la base de datos que tanto tiene que funcionar por parte del menú como con su respectivo botón en la parte inferior.

🗾 D:\Cu	rso Python\Practica guiada\CRUD guiado.py (Curso Python) - Sublime Text (UNREGISTERED)	-		×
<u>F</u> ile <u>E</u> dit	<u>S</u> election F <u>i</u> nd <u>V</u> iew <u>G</u> oto <u>T</u> ools <u>P</u> roject Prefere <u>n</u> ces <u>H</u> elp			
<►	CRUD guiado.py ×			
35	mlUreccion.set("")			1
36	<pre>textoComentario.delete(1.0,END)</pre>		Property.	
37				
38	def crear():		TESE	
39	<pre>miConexion=sqlite3.connect("Usuarios")</pre>		in the second se	
40	miCursor=miConexion.cursor()		Station of the second	
41	<pre>miCursor.execute("INSERT INTO DATOSUSUARIOS VALUES(NULL, '"+ miNombre.get()+</pre>			L
42	"','" + miPass.get()+		100 S	
43	"','" + miApellido.get()+			
44	"','" + miDireccion.get()+		E.	
45	"','" + textoComentario.get("1.0",END)+ "')")			
46	<pre>miConexion.commit()</pre>			
47	<pre>messagebox.showinfo("BBDD", "Registro insertado con éxito")</pre>			
48				
49				
50	root=Tk()		SECONDECTION OF STREET	
51	barraMenu=Menu(root)		STREET, STORY, STORY, ST.	
52	<pre>root.config(menu=barraMenu, width=300, height=300)</pre>			
53				
Line	52, Column 51 Tab Size: 4		Pythor	1

Vamos a llamar la función.



Ahora lo mismo sobre el botón Create.



Ejecutamos:

🦸 tk		_		×
BBDD Borrar	CRUD	Ayuda		
ld:				_
Nombre:			Car	los
Password:	????			_
Apellido:	Ruiz			_
Dirección:	Gran	n Via		_
Comentarios:	Un :	saludo	••	_ ^
				~
Create F	Read	Update	D	elete

Del menú CRUD seleccionaremos Crear.

Ø BBDD ×
Registro insertado con éxito
Aceptar

Ahora con el visor LQLite vamos a comprobar y el registro se ha añadido.

	DB Brov	vser for SQLite - D:\(Curso Python\Pr	actica guiada	\Usuarios								-		×
Arc	hivo <u>E</u>	ditar <u>V</u> er <u>H</u> errar	mientas Ay <u>u</u> da	а											
G	Nueva b	oase de datos 🛛 🔒	Abrir base de dato	os 🔪 🕞 Gu	ardar cambios	Deshacer cam	bios 🔞 🕼 Abri	proyecto	Guardar prog	yecto 🥃	Anexar base de datos	s 🗙 Ce	errar base	de datos	
Es	tructura	Hoja de datos	Editar pragmas	Ejecutar SC	2L			Editar ce	elda						ēΧ
Tal	ola: 🔲 C	ATOSUSUARIOS	- 🕄 🔏	₅ 🖳	a 🔍	🔒 💁 » Fil	ter in any column	Modo:	Texto 🗸	§		B			
	ID Filtro	NOMBRE_USURIO Filtro	PASSWORD Filtro	APELLIDO Filtro	DIRECCION Filtro	COMENTARIOS Filtro		1	1						
1	1	Carlos	1234	Ruiz	Gran Via	Un saludo									
								Tipo de o 1 carácto	datos actualmente er	en la celda: Te:	xto / Numérico			Aplic	tar
								Remoto							₽×
								Identio	dad Pública 🗸						
								Nom	nbre Co	nsolidación	Última modificac	Tamaño			
K	1	0 de 1 🕨 🕅			Ir a:	1		Historia	l de SQL Gráfic	a Esquema	Remoto				
														U	TF-8

Ahora vamos a añadir un segundo registro desde el botón Create.

🧳 tk			_		×						
BBDD	Borrar	CRUD	Ayuda								
	ld:				_						
Ν	lombre:			l	.uis						
Pa	ssword:	????									
۵	pellido:	Pére	z		_						
Di	Dirección: Meridiana										
Come	ntarios:	Otr	o salu	do.							
Crea	te F	Read	Update		Delete						
Ø	BBDD				×						
	Registro insertado con éxito										
				Acepta	r						

BB Browser for SQLite - D\/Curso Python\Practica guidal\Usuarios Archivo Editar Ver Herramientas Ayuda Archivo Editar Ver Herramientas Ayuda Neuve base de datos Abrr base de datos Cambos Comentarios Editar celda Ano Marce Lusura Vola de datos Editar pragmas Eecutar SQL Editar celda Caracter D NOMBRE_USURIO PASSWORD APELLIDO DIRECCION COMENTARIOS Intro Filtro Filtro Filtro Filtro Filtro 1 1 Carlos 1224 Ruiz Gran Via Un saludo Z 2 Luis 5678 Pérez Meridiana Otro saludo Top de datos actualmente en la celda: Texto /Numérico Larácter International de formation of the saludo dato of the saludo Hatorial de SQL Gráfica Esquema Remoto UTER	_										
Archivo Editar Ver Herramientas Ayuda Anachivo Editar Ver Herramientas Ayuda Nueva base de datos Estructura Hoja de datos Editar regimas Editar Celda Editar celda Editar celda Editar celda Ideitar celda Ideitar celda Ideitar celda Ideitar celda Ideitar celda Ideitar celda	1	DB	Brov	vser for SQLite - D:\C	Curso Python\Pr	ractica guiada	\Usuarios			- 0	×
	A	rchive		ditar <u>V</u> er <u>H</u> erran	nientas Ay <u>u</u> d	a					
Estructura Hoja de datos Editar pragmas Ejecutar SQL Editar celda # × Igblas: DATOSUSUARIOS Image: State S		Nu	eva b	ase de datos 🛛 🔒 🗸	Abrir base de dat	os 💡 🕞 Gu	ardar cambios	Deshacer cambio	os 🎯 Abrir	proyecto 🕼 Guardar proyecto 🖓 Anexar base de datos 🗙 Cerrar base de d	atos
I Jable: D NOMBRE_USURIO PATOSUBUARIOS I D I Carlos 124 Ruiz Gran Via Unsaludo Modo: Texto Image:		Estruc	tura	Hoja de datos	Editar pragmas	Ejecutar SC	2L			Editar celda	đΧ
ID NOMBRE_USURIO PASSWORD APELLIDO DIRECCION COMENTARIOS 1 I carlos 1234 Ruiz Gran Via Un saludo 2 2 Luis 5678 Pérez Meridiana Otro saludo 2 2 Luis 5678 Pérez Meridiana Otro saludo Identidad Pública Identidad Pública Identidad Pública Identidad Mombre Consolidación Última modificac Tamaño Identidad Pública Identidad Identidad Identidad SQL Gradica Identidad Identidad SQL Gradica Identidad SQL	I	abla:		ATOSUSUARIOS	× 🕄 🔏	♣ 🖳	8	🖁 🐐 » Filter	r in any column	Modo: Texto 🗸 🎯	•
Filtro Filtro Filtro Filtro Filtro 1 Carlos 1234 Ruiz Gran Via Un saludo 2 Luis 5678 Pérez Meridiana Otro saludo 1 Carlos 1234 Ruiz Gran Via Un saludo 2 Luis 5678 Pérez Meridiana Otro saludo Icarácter Remoto Ø × Identidad Pública Identidad Pública Identidad Nombre Consolidación Útima modificac Tamaño Ir a: 1 Ir a: 1 Istorial de SQ. Gráfica Esquema UTE-8 1 Nombre Consolidación UTE-8 UTE-8		I	D	NOMBRE_USURIO	PASSWORD	APELLIDO	DIRECCION	COMENTARIOS			
1 1 Carlos 1234 Ruiz Gran Via Un saludo 2 2 Luis 5678 Pérez Meridiana Otro saludo Tipo de datos actualmente en la celda: Texto / Numérico 1 carácter Remoto 8 Identidad Pública Identidad Pública Identidad Pública Identidad J1-1 de 2 If ra: 1		Filt	ro I	Filtro	Filtro	Filtro	Filtro	Filtro			
2 2 Luis 5678 Pérez Meridiana Otro saludo Tpo de datos actualmente en la celda: Texto / Numérico Aplicar I carácter 8 8 I dentidad Pública 8 I dentidad Pública 8 Nombre Consolidación Útima modificac I 1 - 1 de 2 Ir a: 1		L	1	Carlos	1234	Ruiz	Gran Via	Un saludo			
Tipo de datos actualmente en la celda: Texto / Numérico Aplicar 1 carácter 8 Remoto 8 Identidad Pública Nombre Consolidación Última modificac Tamaño Historial de SQ. Gráfica UTF-8 UTF-8		2	2	uis	5678	Pérez	Meridiana	Otro saludo			
I Caracter Remoto I Centidad Pública I Identidad Pública I I I I I de 2 I I ra: I I I I I de 2 I I I I I de 2										Tipo de datos actualmente en la celda: Texto / Numérico	Aplicar
Remoto # x Identidad Pública Nombre Consolidación Útima modificac Tamaño Image: Square and Squ										1 caracter	
Identidad Pública Image: Consolidación Útima modificac Tamaño Image: Consolidación Útima modificac Tamaño Image: Consolidación Utima modificac Tamaño Image: Consolidación Utima modificac Tamaño Image: Consolidación Utima modificac Tamaño Image: Consolidación Image: Consolidación Utima modificac Tamaño Image: Consolidación Utima modificac Tamaño Image: Consolidación Image: Consolidación Image: Consolidación Utima modificac Tamaño Image: Consolidación										Remoto	8 ×
Nombre Consolidación Última modificac Tamaño Image: Square										Identidad Pública 🗸 🎲	
Image: Square S Square Square Squa										Nombre Consolidación Última modificac Tamaño	
Image: Squeen and Squeen an											
Image:											
Image:											
If all 1 - 1 de 2 If all 1 Historial de SQL Gráfica UTF-8											
If a: 1 Historial de SQL Gráfica Esquema Remoto	l							1			
UTF-8			1-	1 de 2 🕨 🕅			Ir a:	1		Historial de SQL Gráfica Esquema Remoto	
											UTF-8

Vamos a comprobar se hemos añadido este nuevo registro.





PRÁCTICA GUIADA VI (VIDEO 64)

- Leer registros
- Actualizar registros

Vamos a crear la función leer.



Ahora vamos ha llamar la función desde el menú superior y el botón de la parte inferior.

🗾 D:\Curso Python\Practica guiada	a\CRUD guiado.py • (Curso Python) - Sublime Text (UNREGISTERED)	_		×
<u>File Edit Selection Find View</u>	<u>G</u> oto <u>T</u> ools <u>P</u> roject Prefere <u>n</u> ces <u>H</u> elp			
CRUD guiado.py				•
71 bbddMenu.add_com	mand(<i>label=</i> "Salır", <i>command</i> =salırAplıcacıon)			
72 73 borrarMenu=Menu(barraMenu, <i>tearoff</i> =0)			
74 borrarMenu.add_c	ommand(<i>label=</i> "Borrar campos", <i>command=</i> limpiarCampos)			
75			libra area	
76 crudMenu=Menu(ba	rraMenu, <i>tearoff=</i> 0)		little and and and	
77 crudMenu.add_com	mand(<i>label=</i> "Crear", <i>command=</i> crear)		IE	
78 crudMenu.add_com	mand(<i>label=</i> "Leer", <i>command</i> =leer)		Hartson and	
79 crudMenu.add_com	mand(<i>label</i> ="Actualizar")			
80 crudMenu.add com	mand(<i>label</i> ="Borrar")			
81 –				
82 ayudaMenu=Menu(b	arraMenu, <i>tearoff</i> =0)			
83 ayudaMenu.add co	mmand(<i>label</i> ="Licencia")			
84 ayudaMenu.add co	mmand(<i>label</i> ="Acerca de")			
85				
86 barraMenu.add_ca	scade(<i>label=</i> "BBDD", <i>menu=</i> bbddMenu)			
87 barraMenu.add_ca	scade(<i>label</i> ="Borrar", <i>menu</i> =borrarMenu)			
88 barraMenu.add_ca	scade(<i>label=</i> "CRUD", <i>menu=</i> crudMenu)			
89 hannaMenu add ca	scade(Label="Avuda" menu=avudaMenu)		1993 (1977) (1977)	
Line 85, Column 1	Tab Size	e: 4	Python	



Vamos a guardar y a ejecutar.

🖗 tk		_		×	Ø	tk			_		Х
BBDD Borrar	CRUD	Ayuda			BBD	D Borr	rar	CRUD	Ayuda		
ld:	1						ld:	Crea	ar		-
Nombre:						Nomb	re:	Acto Borr	ualizar rar	-	
Password:						Passwo	rd:				
Apellido:						Apellic	io:				
Dirección:						Direcció	ón:				-
Comentarios:				- ^	Co	mentario	05:				- ^
Create	Read	Update	Dele	te	C	reate	R	ead	Update	De	lete

Este será el resultado:

🦸 tk	– 🗆 X
BBDD Borrar	CRUD Ayuda
ld:	1
Nombre:	Carlos
Password:	????
Apellido:	Ruiz
Dirección:	Gran Via
Comentarios:	Un saludo.
Create	lead Update Delete

Ahora del menú seleccionaremos Borrar y de este borrar campos.

🧳 tk		_		×	Ø	tk			_		×
BBDD Borrar	CRUD	Ayuda			BBD	D Borra	ar C	RUD	Ayuda		
ld:	2					l	d:	2			_
Nombre:						Nombr	e:			L	uis
Password:						Passwor	d:	????			
Apellido:						Apellid	0:	Pére	z		
Dirección:						Direcció	n:	Meri	diana		
Comentarios:				~	Co	mentario	s:	Otro	o salu	do.	
Create	Read	Update	Dele	ete	C	reate	Read	4	Update	D	elete

Desde el botón ejecutamos la misma función.

Ahora vamos a crear la función actualizar().



Ahora vamos a llamar la función desde la barra menú y del correspondiente botón.



Guardaos y ejecutamos.

Primero introducimos el Id del usuario, en nuestro ejemplo tenemos 2, pondremos el 1.

∉tk — □ ×	🖉 tk — 🗆 🗙
BBDD Borrar CRUD Ayuda	BBDD Borrar CRUD Ayuda
ld: 1	ld: 1
Nombre:	Nombre: Carlos
Password:	Password: ????
Apellido:	Apellido: Ruiz
Dirección:	Dirección: Gran Via
Comentarios:	Comentarios: Un saludo.
Curta Band Undeta Deleta	Create Band Undeta Delete
Ahora vamos a cambiar la dirección.	
🖉 tk — 🗆 🗙	
BBDD Borrar CRUD Ayuda	
ld: 1	
Nombre: Carlos	
Password: ????	Ø BBDD ×
Apellido: Ruiz	Registro Actualizado con éxito
Dirección: C/ Arenys de Mar	Aceptar
Comentarios: Un saludo.	
Create Read Update Delete	

Ahora tu lo puedes probar con los menús.





PRÁCTICA GUIADA VII (VIDEO 65)

- Borrar registros
- Consultas parametrizadas

Vamos a crear la función eliminar().







Guardamos y ejecutamos, vamos a eliminar el registro 1, lo primero habrá que leerlo para a posteriori eliminarlo.

.

🦸 tk —		×	🧳 tk		-		×
BBDD Borrar CRUD Ayu	uda		BBDD Borrar	CRUD	Ayuda		
ld: 1			ld:	1			
Nombre:			Nombre:			Carlos	
Password:			Password:	????			
Apellido:			Apellido:	Ruiz	z		
Dirección:			Dirección:	C/ /	Arenyas de	e Mar	
Comentarios:		- ^	Comentarios:	Un	saludo		- ^
Create Read Up	date	te	Create	Read	Update	Dele	te
	BBDD			×			
	() Re	gistro bo	rrado con éxito				
			Aceptar]			

Vamos al visor SQLite para comprobar que dicho registro se ha eliminado.

	DB Bro	owser for SQLite - D:\(Curso Python\Pr	actica guiada	a\Usuarios					-		×
	Nueva	base de datos 🔒	Abrir base de dato	os 🚽 🕞 Gu	uardar cambios	Deshacer cambios	Abrir proyecto	Guardar proyecto	Anexar base de datos	🗙 Cerrar base	de datos	
Es	tructura	a Hoja de datos	Editar pragmas	Ejecutar S	QL		Editar c	elda				đΧ
Tab	ola:	DATOSUSUARIOS	- 🕄 🔏	₅ 🖪	a 🕫	🖁 🀐 » Filter in ar	ny column Modo:	Texto 🗸				
1	ID Filtro 2	NOMBRE_USURIO Filtro Luis	PASSWORD Filtro 5678	APELLIDO Filtro Pérez	DIRECCIO Filtro Meridiana	ON COMENTARIOS Filtro Otro saludo	1	2				
							Tipo de 1 carác	datos actualmente en la ce ter	lda: Texto / Numérico		Aplic	ar
							Remoto)				∂×
							Identi	idad Pública 🗸 🎲				B
							Non	nbre Consolida	ación Última modificac Tar	naño		
K		l - 0 de 1 🕨 🕅			Ir a:	1	Historia	al de SQL Gráfica E	squema Remoto			
											U	TF-8

Observamos que el registro 1 ya no está.

Vamos a crear las dos ventanas emergentes del apartado Ayuda.

🗾 D:\Cu	rso Python\Practica guiada\CRUD guiado.py • (Curso Python) - Sublime Text (UNREGISTERED)		– 🗆 X
<u>F</u> ile <u>E</u> dit	<u>S</u> election F <u>i</u> nd <u>V</u> iew <u>G</u> oto <u>T</u> ools <u>P</u> roject Prefere <u>n</u> ces <u>H</u> elp		
 	CRUD guiado.py		▼
79	micursor.execute(DELETE FROM DATOSUSUARIUS WHERE ID= + mild.get())		
81	messagebox.showinfo("BBDD", "Registro borrado con éxito")		
82 83			
84 85 86	<pre>def licencia(): messagebox.showinfo("LICENCIA", "Esta licenccia es de\n Pere Manel")</pre>		
87 88 89	<pre>def acercade(): messagebox.showinfo("Acerca de", "Base de datos para el\n control de contactos")</pre>		
90 91 92 93 94	root=Tk() barraMenu-Menu(root) root.config <u>(menu</u> =barraMenu, width=300, height=300 <u>)</u>		
95 96			Martin Control Martin Control
Line	93, Column 51	Tab Size: 4	Python

Ahora las vamos a llamar.

D:\Curso Python\Practica guiada\CRUD guiado.py • (Curso Python) - Sublime Text (UNREGISTERED)		_		\times
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp				
CRUD guiado.py •				▼
106 cruamenu=menu(barrāmenu, cearojj =0)		-		
107 crudMenu.add_command(<i>label</i> ="Crear", <i>command</i> =crear)		1	<u> </u>	
108 crudMenu.add_command(<i>label</i> ="Leer", <i>command</i> =leer)		1		
109 crudMenu.add_command(<i>label</i> ="Actualizar", <i>command</i> =actualizar)		1	And the other states are a second states and the second states are a second states are a second states are a s	
110 crudMenu.add_command(<i>label</i> ="Borrar", <i>command</i> =eliminar)		_		
111				
112 ayudaMenu=Menu(barraMenu, tearoff=0)		1		
113 ayudaMenu.add_command(<i>Label</i> ="Licencia", <i>command</i> =licencia)		1	and the second s	
114 ayudaMenu.add_command(<i>label</i> ="Acerca de", <i>command</i> =acercade)		E5		
115				
116 barraMenu.add_cascade(<i>label</i> ="BBDD", <i>menu</i> =bbddMenu)			i Diker	
117 barraMenu.add_cascade(<i>Label</i> ="Borrar", <i>menu</i> =borrarMenu)				
118 barraMenu.add_cascade(<i>label</i> ="CRUD", <i>menu</i> =crudMenu)		10		
119 barraMenu.add cascade(<i>Label</i> ="Ayuda", <i>menu</i> =ayudaMenu)		1		
120		10		
121 #Comienzo de campos				
122				
123 miFrame=Frame(root)				
Line 119, Column 53	Tab Size: 4		Pythor	1



Aquí está todo el código: from tkinter import * from tkinter import messagebox import sqlite3

#-----Funciones----

def conexionBBDD():

```
miConexion=sqlite3.connect("Usuarios")
miCursor=miConexion.cursor()
try:
miCursor.execute('''
```

```
CREATE TABLE DATOSUSUARIOS(
ID INTEGER PRIMARY KEY AUTOINCREMENT,
NOMBRE_USURIO VARCHAR(50),
PASSWORD VARCHAR(50),
APELLIDO VARCHAR(10),
DIRECCION VARCHAR(50),
COMENTARIOS VARCHAR(100))
''')
```

messagebox.showinfo("BBDD", "BBDD creada con éxito")

except:

```
messagebox.showwarning("¡Atención!", "La BBDD ya existe")
```

```
def salirAplicacion():
```

```
valor=messagebox.askquestion("Salir", "¿Deseas salir de la aplicación?")
if valor=="yes":
root.destroy()
```

def limpiarCampos():

```
mild.set("")
miNombre.set("")
miPass.set("")
miApellido.set("")
miDireccion.set("")
textoComentario.delete(1.0,END)
```

```
def crear():
```

```
miConexion=sqlite3.connect("Usuarios")
       miCursor=miConexion.cursor()
       miCursor.execute("INSERT INTO DATOSUSUARIOS VALUES(NULL, "+ miNombre.get()+
               "','" + miPass.get()+
              "','" + miApellido.get()+
               "','" + miDireccion.get()+
               "'," + textoComentario.get("1.0",END)+ "')")
       miConexion.commit()
       messagebox.showinfo("BBDD", "Registro insertado con éxito")
def leer():
       miConexion=sqlite3.connect("Usuarios")
       miCursor=miConexion.cursor()
       miCursor.execute("SELECT * FROM DATOSUSUARIOS WHERE ID=" + mild.get())
       elUsuario=miCursor.fetchall()
       for usuario in elUsuario:
               mild.set(usuario[0])
              miNombre.set(usuario[1])
               miPass.set(usuario[2])
               miApellido.set(usuario[3])
              miDireccion.set(usuario[4])
              textoComentario.insert(1.0, usuario[5])
       miConexion.commit()
def actualizar():
       miConexion=sqlite3.connect("Usuarios")
       miCursor=miConexion.cursor()
       miCursor.execute("UPDATE DATOSUSUARIOS SET NOMBRE USURIO='" +
miNombre.get() +
               "', PASSWORD='" + miPass.get() +
               "', APELLIDO="" + miApellido.get() +
               "', DIRECCION="" + miDireccion.get() +
              "', COMENTARIOS="" + textoComentario.get("1.0",END) +
               "'WHERE ID="+ mild.get())
       miConexion.commit()
       messagebox.showinfo("BBDD", "Registro Actualizado con éxito")
```

```
def eliminar():
```

```
miConexion=sqlite3.connect("Usuarios")
miCursor=miConexion.cursor()
miCursor.execute("DELETE FROM DATOSUSUARIOS WHERE ID="+ mild.get())
miConexion.commit()
messagebox.showinfo("BBDD", "Registro borrado con éxito")
```

def licencia():

messagebox.showinfo("LICENCIA", "Esta licenccia es de\n Pere Manel")

def acercade():

messagebox.showinfo("Acerca de...", "Base de datos para el\n control de contactos")

root=Tk()

barraMenu=Menu(root)

root.config(menu=barraMenu, width=300, height=300)

-----Barra de menú-----

bbddMenu=Menu(barraMenu, tearoff=0) bbddMenu.add_command(label="Conectar", command=conexionBBDD) bbddMenu.add_command(label="Salir", command=salirAplicacion)

borrarMenu=Menu(barraMenu, tearoff=0) borrarMenu.add_command(label="Borrar campos", command=limpiarCampos)

crudMenu=Menu(barraMenu, tearoff=0) crudMenu.add_command(label="Crear", command=crear) crudMenu.add_command(label="Leer", command=leer) crudMenu.add_command(label="Actualizar", command=actualizar) crudMenu.add_command(label="Borrar", command=eliminar)

ayudaMenu=Menu(barraMenu, tearoff=0) ayudaMenu.add_command(label="Licencia", command=licencia) ayudaMenu.add_command(label="Acerca de...", command=acercade)

barraMenu.add_cascade(label="BBDD", menu=bbddMenu) barraMenu.add_cascade(label="Borrar", menu=borrarMenu) barraMenu.add_cascade(label="CRUD", menu=crudMenu) barraMenu.add_cascade(label="Ayuda", menu=ayudaMenu)

#-----Comienzo de campos-----

miFrame=Frame(root) miFrame.pack()

mild=StringVar() miNombre=StringVar() miApellido=StringVar() miPass=StringVar() miDireccion=StringVar() cuadroID=Entry(miFrame, textvariable=mild)
cuadroID.grid(row=0, column=1, padx=10, pady=10)

cuadroNombre=Entry(miFrame, textvariable=miNombre)
cuadroNombre.grid(row=1, column=1, padx=10, pady=10)
cuadroNombre.config(fg="red", justify="right")

cuadroPass=Entry(miFrame, textvariable=miPass)
cuadroPass.grid(row=2, column=1, padx=10, pady=10)
cuadroPass.config(show="?")

cuadroApellido=Entry(miFrame, textvariable=miApellido)
cuadroApellido.grid(row=3, column=1, padx=10, pady=10)

cuadroDireccion=Entry(miFrame, textvariable=miDireccion)
cuadroDireccion.grid(row=4, column=1, padx=10, pady=10)

textoComentario=Text(miFrame, width=16, height=5)
textoComentario.grid(row=5, column=1, padx=10, pady=10)
scrollVert=Scrollbar(miFrame, command=textoComentario.yview)
scrollVert.grid(row=5, column=2, sticky="nsew")
textoComentario.config(yscrollcommand=scrollVert.set)

#----Aquí comienzan las etiquetas-----

idLabel=Label(miFrame, text="Id:")
idLabel.grid(row=0, column=0, sticky="e", padx=10, pady=10)

nombreLabel=Label(miFrame, text="Nombre:")
nombreLabel.grid(row=1, column=0, sticky="e", padx=10, pady=10)

passLabel=Label(miFrame, text="Password:")
passLabel.grid(row=2, column=0, sticky="e", padx=10, pady=10)

apellidoLabel=Label(miFrame, text="Apellido:")
apellidoLabel.grid(row=3, column=0, sticky="e", padx=10, pady=10)

direccionLabel=Label(miFrame, text="Dirección:")
direccionLabel.grid(row=4, column=0, sticky="e", padx=10, pady=10)

```
comentariosLabel=Label(miFrame, text="Comentarios:")
comentariosLabel.grid(row=5, column=0, sticky="ne", padx=10, pady=10)
```

#-----Aquí los botones----

```
miFrame2=Frame(root)
miFrame2.pack()
```

botonCrear=Button(miFrame2, text="Create", command=crear)
botonCrear.grid(row=1, column=0, sticky="e", padx=10, pady=10)

botonLeer=Button(miFrame2, text="Read", command=leer)
botonLeer.grid(row=1, column=1, sticky="e", padx=10, pady=10)

botonActualizar=Button(miFrame2, text="Update", command=actualizar) botonActualizar.grid(row=1, column=2, sticky="e", padx=10, pady=10)

botonBorrar=Button(miFrame2, text="Delete", command=eliminar)
botonBorrar.grid(row=1, column=3, sticky="e", padx=10, pady=10)

root.mainloop()

Ahora vamos a comentar que son consultas parametrizadas.



Con tres comillas al principio y luego al final pasamos una parte de texto a comentario, es decir a la hora de ejecutarlo lo ignorará y la podré recuperar cuando quiera.



Guardamos y ejecutamos.

🧳 tk	_		×	
BBDD Borrar	CRUD Ayuda			
ld:				
Nombre:		Carlos		Ø BBDD
Password:	?????			Registro insertado con éxito
Apellido:	Martínez			
Dirección:	Can Palau			Aceptar
Comentarios:	Buen curso de Python		- ^	
			\sim	
Create	Read Update	Dele	te	

Este es el código:

def crear():

miConexion=sqlite3.connect("Usuarios")
miCursor=miConexion.cursor()
datos=miNombre.get(), miPass.get(),miApellido.get(), miDireccion.get(),
 textoComentario.get("1.0",END)
miCursor.execute("INSERT INTO DATOSUSUARIOS VALUES(NULL,?,?,?,?,?)",(datos))

El recuadro es una línea.

Comprobamos si se añadió el registro.

	DB Bro	wser for SQLite - D:\C	Curso Python\Pr	actica guiada	\Usuarios											-		×
Arc	thivo	<u>E</u> ditar <u>V</u> er <u>H</u> errar	nientas Ay <u>u</u> d	a														
G	Nueva	base de datos 🛛 🔒	Abrir base de dati	os 🖕 🕞 Gu	ardar cambios	Carl Deshacer cam	bios 🔞 Abri	ir proyecto	Guard	dar proyecto	🔒 An	nexar base	de dato	os 🔰	🗙 Cerr	rar base	de datos	
Es	structura	Hoja de datos	Editar pragmas	Ejecutar SQ	2L			Editar c	elda									8×
Tal	bla:	DATOSUSUARIOS	~ 😂 🔏	₅ 🖳	a 🔍 I	B 🐴 » Fi	Iter in any column	Modo:	Texto	~			9			-		
	ID	NOMBRE_USURIO	PASSWORD	APELLIDO	DIRECCION	COMENTARIOS		1	E									
	Filtro	Filtro	Filtro	Filtro	Filtro	Filtro]	1	2									
1	5	Carlos	12345	Martínez	can Palau	Buen curso d												
								Tipo de 1 carác	datos actual	lmente en la cel	da: Texto	/ Numérico					Apli	car
								Parmate	-									
								nemote	0									
								Ident	tidad Pública	a 🗸 🎲								B
								Nor	mbre	Consolida	ción Ú	Última mo	dificac	Tama	ño			
k	1 1	- 0 de 1 🕨 🕅			Ir a:	1		1. Cabardo	-1 -1 - 5 -0 -	0.46-0 5								
								historia	ai de SQL	Granca Es	squema	Remoto						

También abría que hacerlo con el actualizar.



Este es el código:

def actualizar():

miConexion=sqlite3.connect("Usuarios")
miCursor=miConexion.cursor()
datos=miNombre.get(), miPass.get(), miApellido.get(), miDireccion.get(),
 textoComentario.get("1.0",END)
miCursor.execute("UPDATE DATOSUSUARIOS SET NOMBRE_USURIO=?, PASSWORD=?,
 APELLIDO=?,DIRECCION=?, COMENTARIOS=? "+ "WHERE ID=" +
 mild.get(),(datos))

Cada recuadro es una línea.





FUNCIONES LAMBDA (VIDEO 66)

• Qué son las funciones lambda y cuál es su utilidad

Vamos a crear un nuevo proyecto llamado practica_lambda.py. Sirve para resumir una función normal expresión lambda en Python.



Hemos creado una función que calcula el área de un triángulo que después con un print recuperamos la información.

Si queremos repetir varias veces esta función le vamos a asignar a esta función dos variables.



Este será el resultado:



¿Cómo podemos resumir más con una función lambda?





Para recuperar la función lambda se muestra en las siguientes líneas. También es correcto:



<u>T</u>	East Delection Find	<u>1</u> .cm <u>0</u> .cu	<u>1</u> 0015 <u>1</u> 10Jeee	Therefore <u>in</u> e	<u>n</u> ap		
<۲	practica_lambda.py	×	REPL* [python]	×		•	
17.5	27.0					-	
R	epl Closed						
	Line 4, Column 1				Tab Size: 4	Python	

Esta función se llama función anónima porque no tiene nombre y la podemos utilizar las veces que sea necesario.

También se denominan funciones "on then go", "on demand" o "online". Otro ejemplo:



En este ejemplo vemos dos formas distintas de calcular un valor al cubo.

Otro ejemplo:



También se utilizar la función lambda para imprimir determinados formatos.





FUNCIÓN FILTER (VIDEO 67)

- Función Filter y utilidad
- Verifica que los elementos de una secuencia cumplen una condición, devolviendo un iterador con los elementos que cumplan dicha condición.

Vamos a crear un nuevo proyecto llamado practica_filter.py



Hemos creado una función llamada numero_par(num) que admite un parámetro que comprobamos si un número es divisible por 2, es decir par, si es así nos retorna un True. A continuación creamos una lista de números.

Ahora para utilizar la función Filter, tenemos que decirle que imprima una lista utilizando la función filter, si numero_par retorna un True imprime el número y si no, no lo imprime. Este será el resultado cuando ejecutemos el código:



Aquí veremos como se puede simplificar el código utilizando la función lambda.



Sí ejecutamos obtendremos el mismo resultado:

5 *	*REPL*	[python] (C	_		Х							
<u>F</u> ile	<u>E</u> dit	Selection	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ce	s <u>H</u> elp			
.		practica_filte	r.py		*R	EPL* (pyt	:hon]	×				▼
[24	, ⁸ ,	92]									THE SECOND	
***	Repl	Closed*										
	Line 4	4, Column 1							Tab Size: 4	4	Python	

Vamos a utilizar filter para filtrar objetos.



Este será el resultado:



Hemos creado una clase llamada Empleados con los campos nombre, cargo y salario.

Con def __str__(self): le decimos con que formato de salida quiero que salgan los datos.

Creamos una listaEmpleados que contiene 5 empleados.

A la variable salarios altos le pasamos aquellos empleados que cobran más de 50000.

Al final con ayuda de un for imprimimos aquellos empelados que cobran más de 50000.





FUNCIÓN MAP. (VIDEO 68)

- Función Map y utilidad
- Aplica una función a cada elemento de una lista iterable (listas, tuplas, etc.) devolviendo una lista con los resultados.

Vamos a crear un nuevo proyecto llamado practica_map.py y copiaremos pare del código del proyecto anterior.



Queremos poner una comisión para aquellos sueldos que son bajos.



En la función calculo_comision(empleado) hacemos que el sueldo de cada empleado se incremente un 3%.

Con la instrucción señalada le estamos diciendo que en una nueva variable asigne a la lista los mismos empleados pero con un sueldo incrementado al 3%.

Con el for hacemos la consulta de todos los empleados.

Este será el resultado:



Ya hemos incrementado el sueldo mensual un 3%.

Ahora queremos que esta comisión solo sea para los sueldos menores de 3000 €.



Con la condición hacemos que la comisión solo sea para los salarios menores o igual a 3000.

Este será el resultado:



Como podrás observar los sueldos mayores a 3000 no se han tocado.


EXPRESIONES REGULARES I (VIDEO 69)

- Qué son las expresiones regulares: sintaxis y ejemplos sencillos.
- ¿Qué son?
 - Las expresiones regulares son una secuencia de caracteres que forman un patrón de búsqueda?

Expresiones regulares o regex

- ¿Para qué sirven?
 - Para el trabajo y procesamiento de texto.
- Ejemplos:
 - Buscar un texto que se ajusta a un formato determinado (correo electrónico).
 - Buscar si existe o no una cadena de caracteres dentro de un texto.
 - o Etc.

Vamos a crear un nuevo proyecto llamado practica_regex.py

Queremos encontrar la palabra aprender dentro de la variable cadena

D:\Curso Python\practica_regex.py • (Curso Python) - Sublime Text (UNREGISTERED)	-	- 🗆	\times										
<u>File E</u> dit <u>S</u> election Find <u>V</u> iew <u>G</u> oto <u>T</u> ools <u>P</u> roject Prefere <u>n</u> ces <u>H</u> elp													
◆ practica_regex.py ●			▼										
1 import re 2													
<pre>2 3 cadena="Vamos a aprender expresiones regulares" 4 4</pre>													
4 5 print(re.search("aprender", cadena))													
Line 5, Column 37	Tab Size: 4	Pytł	non										
Si guardamos y ejecutamos:													
REPL [python] (Curso Python) - Sublime Text (UNREGISTERED)	-	- 🗆	×										

♦ practica_regex.py × *REPL* [python] ×		•
<re.match <i="">object; span=(8, 16), match='aprender'></re.match>		
Repl Closed		
Line 4, Column 1	Tab Size: 4	Python

La ejecución nos devuelve un objeto llamado SER_Match, que el span es de 8 a 16 y que lo ha encontrado.

Si el texto no está:



Cuando ejecutemos.

5	*REPL* [python] (Curso Python) - Sublime Text (UNREGISTERED) - C													
<u>F</u> ile	<u>E</u> dit	<u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp					
•)	• /	practica_rege	ex.py		*R	EPL* (pyt	hon]	×					▼	
Non ***	e Repl	Closed*										-		
	Line	4, Column 1								Tab Size	: 4	Python		

 Para que nos diga si ha encontrado el texto o no vamos a realizar el siguiente código:

 Image: D:\Curso Python\practica_regex.py • (Curso Python) - Sublime Text (UNREGISTERED)

×



Este será el resultado:







Si ejecutamos este será el resultado:

	5 *	*REPL* [python] (Curso Python) - Sublime Text (UNREGISTERED) -													
E	ile	<u>E</u> dit	<u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp					
	• •	Γ	practica_rege			*R	EPL* (pyt	:hon]	×					▼	
8 1 (6 8, **[16) 2001	Closed										L		
		терт	CIOSed*										0.11	_	
		Line	6, Column 1								Tab Size: 4		Python		

Vamos a buscar una palabra que se repite en una frase.

D:\Curso Python\practica_regex.py • (Curso Python) - Sublime Text (UNREGISTERED)	-	- 0	×
Eile Edit Selection Find View Goto Tools Project Preferences Help			
◆ practica_regex.py ●			
2 3 cadena="Vamos a aprender expresiones regulares en Python. Python es un lenguaje de sintaxis sencil: 4 5 textoBuscar="Python"	las"	in the first state areas source and a time, the interaction of the state of the st	
<pre>6 7 print(re.findall(textoBuscar, cadena)) 7 print(re.findall(textoBuscar, cadena))</pre>			
Line 7, Column 10 T	Tab Size: 4	Pyth	ion

Para ello utilizamos el parámetro findall,

este será el resultado:

REPL [python] (Curso Python) - Sublime Text (UNREGISTERED) —													
<u>F</u> ile <u>E</u> dit <u>S</u>	Selection Fir	nd <u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp						
▲ ► pr	actica_regex.py		*RI	EPL* (pyt	hon]	×					▼		
['Python'	, 'Pytho	n']								The set			
['Python', 'Python'] ***Repl Closed***													
Line 4, C	Column 1							Tab Si	ze: 4	Python			

Nos devuelve una lista con el numero de veces que aparece la palabra "Python".

Si queremos saber cuantas veces aparece la palabra Python utilizaremos la función len.



Si ejecutamos, este será el resultado.



Aparece 2 veces.





EXPRESIONES REGULARES II (VÍDEO 70)

- Expresiones regulares. Metacaracteres (caracteres comodín)
 - Anclas y clases de caracteres

Vamos a comenzar por las Anclas.



Hemos creado una lista con 4 personas, con el ciclo for hacemos un recorrido por toda la lista y queremos encontrar aquellas personas que su nombre empieza por Sandra, con el condicionante le estamos diciendo que si encuentra a alguna Sandra me imprima el nombre completo. Este será el resultado:



Agregamos a una nueva Sandra.



Este será el resultado:

۶	REPL*	[python] (C		_		Х						
<u>F</u> ile	<u>E</u> dit	<u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> c	es <u>H</u> elp			
4 Þ		practica_rege	ex.py		*RI	EPL* (pyt	hon]	×				▼
San San	dra dra Repl	López Fernánde Closed*	:z **					_			EL.	
	Line !	5, Column 1							Tab Size: 4	ļ	Python	

Ahora queremos buscar a todas aquellas personas que su apellido termina por Martín.



Si ejecutamos este será el resultado:



Para saber los dominios que tengo con terminación es.

D:\Curso Python\practica_regex.py • (Curso Python) - Sublime Text (UNREGISTER	_		×
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
♦ practica_regex.py			▼
1 import re		100	
2 2 lista nombras_['http://mildonasinformaticas.as'			
5 lista_nombres=[nttp://pildorasinformaticas.es ,			
4 'ttp://pildorasinformaticas.es',			
5 'http://pildorasinformaticas.com',			
6 'ftp://pildorasinformaticas.com']			
7			
<pre>8 for elemento in lista nombres:</pre>			
<pre>9 if re_findall('es\$', elemento):</pre>			
10 print(elemento)			
in himerence)			
Line 10, Column 24 Tab Size: 4		Python	

Este será el resultado:



Los dominios que empiezan por ftp.

5):\Cu	rso Python∖p	oractica	_regex.	oy•(Cu	rso Pytł	hon) - Sub	lime Text (UN	REGISTER	—		\times
<u>F</u> ile	<u>E</u> dit	<u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp			
▲ ►		practica_reg	ex.py	•								▼
1		import <mark>r</mark>	e								- 1997 	
2 3 4 5 6 7		lista_no	mbres	s=['ht '+ ' '+	ttp:// ftp:// nttp:/ ftp://	/pild /pild //pil /pild	orasin orasin dorasi orasin	formatica formatica nformatic formatica	s.es', s.es', as.com', s.com']			
9 10		for elem if r	ento e.fin print	in li ndall(: <u>(</u> eler	ista_ ('^ft mento	nombr p', e <u>)</u>	es: lemento	o):				
	Line	10, Column 2	4						Tab Size: 4		Python	

Este será el resultado:



También podemos buscar caracteres que se encuentran en este texto estos irán entre corchetes.



Nos devolverá aquellos dominios que tengan el carácter 'ñ'.

Ejecutamos.



Vamos a realizar una búsqueda para encontrar los niños y las niñas.

5	D:\Cu	rso Python∖p	oractica	_regex.	oy•(Cu	rso Pytł	hon) - Sub	lime Text (UNI	REGISTER	_		Х
<u>F</u> ile	<u>E</u> dit	<u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp			
•	• /	practica_reg	ex.py	•								▼
	1 2	import r	e									
	3	lista_no	mbres	5=['ho	ombre	s',						
4	4			''	nujer	es',						
	5			· ·	nasco	tas',						
(6			· · ·	niños	' ,						
	7			· · ·	niñas	']						
1	8											
9	9	<mark>for</mark> elem	ento	in li	ista_	nombr	es:					
10	9											
1	1	if r	e.fir	ndall	('niñ	[oa]s	', ele	mento):				
1	2											
1	3		print	(eler	nento)						
	Line	13, Column 2	4						Tab Size: 4		Python	

Si ejecutamos.

REPL [python] (Curso Python) - Sublime Text (UNREGISTERED) -													
<u>F</u> ile	<u>E</u> dit	<u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp				
••	. /	practica_rege	ex.py		*RI	EPL* (pyt	hon]	×				▼	
niñ niñ	os as												
***	Repl	Closed*											
	Line	5, Column 1							Tab Size: 4		Python		

Queremos buscar la palabra camión que puede estar escrita con acento o sin acento.



Si ejecutamos.







EXPRESIONES REGULARES III (VÍDEO 71)

- Expresiones regulares.
 - o Rangos

Los rangos nos permiten buscar por un rango de caracteres o un rango de números.

Queremos buscar nombres que empiezan por O,P,Q,R,S,T esto es igual al rango O-T

5	D:\Cur	so Python∖p	oractica	_regex.	py•(Cu	irso Pytł	non) - Sub	lime Text (UNI	REGISTER	—		×
<u>F</u> ile	<u>E</u> dit	<u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp			
••		practica_reg	ex.py	•								▼
1	! : !	import r	e									
3 4 5 6 7 8	; ; ; ;	lista_no	mbres	5=['Aı '! '! !!	na', Pedro María Rosa' Sandr Celia	', , a', ']						
10 11		for elem	ento	in 1 :	ista_	nombr	es:					
12		if r	e.fir	ndall	('^[0	-T]',	eleme	nto <u>)</u> :				
14	, 		print	(ele	nento)						
	Line	12, Column 2	5						Tab Size: 4		Python	

Si lo ejecutamos.



Queremos ver todos los nombres que terminan con el rango o-t.

5	D:\Cur	rso Python∖p	oractica	_regex.	py•(Cu	irso Pytł	non) - Sub	lime Text (UNI	REGISTER	_		×	
<u>F</u> ile	<u>E</u> dit	<u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp				
••		practica_rege	ex.py	•								▼	
1	L :)	import r	e										
	3 1 5 7 3	lista_no	mbres	5=['Aı '! '! '!	na', Pedro María Rosa' Sandr Celia	', , a', ']							
10 11	9 10 for elemento in lista_nombres: 11												
12 13	<pre>11 12 if re.findall('[o-t]\$', elemento): 13</pre>												
14	l.		print	(ele	nento)							
	Line	14, Column 2	4						Tab Size: 4		Python		

Este será el resultado:

	>	'REPL*	[python] (C	Curso P	ython) -	Sublim	ne Text (UNREGIST	ERED)			_		×
E	ile	<u>E</u> dit	\underline{S} election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>r</u>	<u>n</u> ces	<u>H</u> elp			
	<►		practica_reg	ex.py		*R	EPL* (pyt	thon]	×					▼
P	edı	ro												
*	**	Repl	Closed*											
		Line	4, Column 1								Tab Size: 4		Python	

Queremos ver los códigos Ma1, Ma2 y Ma3.



Este será el resultado:

5	*REPL*	[python] (C	Curso P	ython) -	Sublim	ne Text (UNREGIST	ERED)			_		×
<u>F</u> ile	<u>E</u> dit	<u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp				
••	• /	practica_rege	ex.py		*RI	EPL* (pyt	:hon]	×					▼
Ma1 Ma2 Ma3												L_	
***	Repl	Closed*											
	Line	6, Column 1								Tab Size: 4		Python	

Queremos buscar los códigos de Madrid que no sea Ma1, Ma2 ni Ma3



Este será el resultado:

5	'REPL*	[python] (C	urso Py	ython) ·	Sublim	ne Text (UNREGIST	ERED)			_		×	
<u>F</u> ile	<u>E</u> dit	<u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp					
••		practica_rege	ex.py		*RI	EPL* (pyt	hon]	×					▼	
Ma4														
***	***Repl Closed***													
	Line 4, Column 1 Tab Size: 4 Python													

Queremos los de Ma1, Ma2, Ma3, MaA y MaB dos rangos.



Este será el resultado:

_												
1	REPL*	' [python] (C	urso P	/thon) -	Sublim	ne Text (UNREGIST	ERED)		-		×
<u>F</u> ile	<u>E</u> dit	Selection	Find	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	Help			
••					*RI	EPL* (pyt	thon]	×				▼
Ma1 Ma2 Ma3 MaA MaB		•								I		
***	Repl	Closed*										
	Line	8, Column 1							Tab Size: 4		Pythor	1

Queremos ver aquellos códigos que tengan un . o dos :.



Este sería el resultado:

REPL [python] (0	Curso Py	thon) -	Sublim	ne Text (I	UNREGIST	ERED)		-		×
<u>File Edit Selection</u>	Find	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp			
<pre>practica_reg</pre>		×	*RI	EPL* (pyt	hon]	×				
Ma.1								L		
Ma:3										
Ma.5										
Ma:C										
***Repl Closed										
Line 7, Column 1							Tab Size: 4		Python	





EXPRESIONES REGULARES IV (VÍDEO 72)

- Funciones del módulo re
 - Match y search

Queremos buscar si al principio del valor de una variable contiene el nombre de Sandra.

5	D:\Cu	rso Python∖p	oractica	_regex.	oy•(Cu	irso Pytł	non) - Sub	lime Text (—		\times
<u>F</u> ile	<u>E</u> dit	<u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp		
▲ ►		practica_rege	ex.py	•							▼
1		import r	e								-
3	3	nombre1=	"Sand	Ira Lo	ópez"						
4	l I	nombre2=	"Anto	onio (Gómez						
5	i 1	nombre3=	"Mari	a Lóp	oez"						
6											
7		if re.ma	tch("	Sandi	ra", I	nombr	e1):				
8		prin	t("He	ems er	ncont	rado	el nom	bre")			
9		else:									
10)	prin	t("No) lo	nemos	enco	ntrado'	")			
11											
	Line	11, Column 1						Tab Size: 4		Python	

Ejecutamos.



Si se lo preguntamos por la variable nombre2 que no contiene el nombre de Sandra.

5	D:\Cu	ırso Python∖p	oractica	_regex.	oy•(Cu	irso Pytł	non) - Sub	lime Text (_		\times
<u>F</u> ile	<u>E</u> dit	<u>Selection</u>	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp		
▲ ►	- /	practica_rege	ex.py	•							▼
1		import r	e								-
_ 2											
З	}	nombre1=	"Sand	ira Lo	ópez"						
4		nombre2=	"Anto	onio (Gómez	••					
5	5	nombre3=	"Mari	ía Lói	bez"						
6	5										
		if re ma	tch('	'Sandı	ra"	nombr	e2).				
5	2	nnin	+/"H4		a ,	nado	el nomi	hre")			
	, \	- prim	C(116	- III 5 EI	iconc	lauo		bie)			
5	,	erse:									
10)	prin	t("No	o lo H	nemos	enco	ntrado'	")			
11											
	Line	11, Column 1						Tab Size: 4		Python	

Y ejecutamos.

5	REPL*	[python] (C	urso Py	/thon) -	Sublim	e Text (UNREGIST	ERED)	_		×			
<u>F</u> ile	<u>E</u> dit	<u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp					
▲ ►		practica_rege	ex.py		*RI	EPL* (pyt	hon]	×			▼			
No	No lo hemos encontrado													
***	***Repl Closed***													
	Line 4	4, Column 1						Tab Size: 4		Python				

Si queremos que no diferencie entre minúsculas y mayúsculas utilizaremos IGNORECASE



Ejecutamos.

5	*REPL*	[python] (C	Curso Py	/thon) -	Sublim	ne Text (UNREGIST	ERED)	_		×			
<u>F</u> ile	<u>E</u> dit	<u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp					
.		practica_rege	ex.py		*RI	EPL* (pyt	hon]	×			▼			
Hem	Hemos encontrado el nombre													
***	***Repl Closed***													
	Line 4	4, Column 1						Tab Size: 4		Python				

Como utilizar el carácter comodín con el punto



Sustituye la primera letra del nombre, ejecutamos tanto con el nombre1 y el nombre3 dirá "Hemos encontrado el nombre".

Queremos que busque una cadena que tiene números.



Ejecutamos.



En este segundo caso:



Ejecutamos.



Y en el tercer ejemplo.

5):\Curso P	ython∖p	ractica	_regex.	py•(Cu	irso Pytł	non) - Sub	lime Text (_		\times
<u>F</u> ile	<u>E</u> dit <u>S</u> e	lection	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp		
••	prac	tica_rege	ex.py	•							▼
1 2	imp	ort r	e								
З	cad	ena1=	"Jara	Lópe	ez"						
4	cad	ena2=	"5465	4654	5"						
5	cad	ena3=	"a546	54654	4"						
6											
7	if	re.ma	tch("	'∖d",	cade	na3 <u>)</u> :					
8		prin	t("He	emos e	encon	trado	el nú	mero")			
9) els	e:									
10)	prin	t("No) lo	nemos	enco	ntrado	el númer	o")		
11											
	Line 7, Co	olumn 26						Tab Size: 4		Python	

Ejecutamos.

🗾 *RE	PL* [python] (C	Curso P	/thon) -	Sublim	ne Text (UNREGIST	ERED)	_		\times			
<u>F</u> ile <u>E</u>	dit <u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp					
4 ►	practica_rege	ex.py		*RI	EPL* (pyt	hon]	×			▼			
No lo	No lo hemos encontrado el número												
***Re	pl Closed*												
Li	ne 4, Column 1						Tab Size: 4		Python				

La función search busca en toda la cadena de texto.



En la cadena3 empieza por una letra la función match como solo busca al principio nos decía que no había encontrado números, pero la función search busca en toda la cadena, lo vamos a probar con la variable cadena3 que empieza por letras.



Ejecutamos.



La función search es muy útil en cadenas de texto muy largas si hay un patrón a seguir.





DECORADORES I (VIDEO 73)

- Qué son los decoradores
 - Funcionamiento y sintaxis

Decoradores. ¿Qué son)

- Funciones que a su vez añaden funcionalidades a otras funciones.
 - Por eso se las denomina "decoradores", porque "decoran" alas funciones. Les añaden funcionalidades.
- Estructura de un decorador:
 - 3 funciones (A,B y C) donde A recibe como parámetro a B para devolver C.
 - Un decorador devuelve una función

Estructura:

def función_decorador(función):

def función interna():

código de la función interna

return función_interna

Vamos a realizar un nuevo proyecto llamado Ejemplo_decoradores.py.



La función decoradora hace que otras funciones sean completadas con más operaciones ya que la función en enviada a la función decoradora que esta la ejecuta con el resto de código que esta tiene.

Una función de suma la ejecutamos utilizando la función decoradora y la función resta la ejecutamos directamente, cuando ejecutemos veremos las diferencias.



Ahora vamos a decorar la función resta.



Este será el resultado:







DECORADORES II (VIDEO 74)

• Decoradores con parámetros



Este será el resultado.



Podemos agregar más parámetros:

D:\Curso Python\Ejemplo_decoradores.py • (Curso Python) - Sublime Text (UNREGISTERED)	_		×
<u>File Edit Selection Find View Goto Tools Project Preferences H</u> elp			
Ejemplo_decoradores.py			▼
<pre>1 def funcion_decoradora(funcion_parametro):</pre>		The second secon	
2		Pagasar Pagasar	
3 def funcion_interior(*args):			
4 # Acciones adicionales que decoran			
6			
7 funcion parametro(*args)			
8			
9 # Acciones adicionales que decoran			
10 print("Hemos terminado el cálculo:")			
11			
12 return funcion_interior			
14 @tuncion_decoradora			
15 aef suma(num1, num2, num2):			
18 Ofuncion decoradora			
19 def resta(num1, num2):			
20 print(num1-num2)			
21			
22 suma(7,5,8) 🥊			
23			
24 resta(12,10)			
Line 16, Column 25 Tab Size: 4		Python	
		Fython	

Ejecutamos



No importa el número de argumentos que tenga la función, los admite todos.

Cuando pasamos a la función potencia 2 argumentos que llevan su clave para ello lo vamos a realizar de la siguiente forma:



Este será el resultado:







FUNCIONES DECORADORAS II (CORRECCIÓN) (VIDEO 74b)

En el capítulo anterior se olvidó decorar la función que trabajaba y la ejecutó sin más, en este vídeo explica el proceso final.



DOCUMENTACIÓN (VIDEO 75)

• Documentar nuestros programas.

¿Qué es?

• Incluir comentarios en clases, métodos, módulos, etc.

¿Para qué?

• Para ayudar en el trabajo en equipo. Especialmente útil en aplicaciones complejas.

Vamos a realizar un nuevo proyecto llamado Ejemplo_documentacion.py



En una función podemos agregar un comentario para posibles recordatorios que irán entre tres comillas dobles, este comentario no se ejecuta, pero si en un momento determinado queremos imprimir este comentario realizaremos la línea que hemos marcado.

Le estamos diciendo que emprima el comentario de la función areaCuadrado, si ejecutamos este será el resultado:





También podemos utilizar la función help pasándole como parámetro la función.

Este será el resultado:



Agregamos un nuevo comentario en función areaTriangulo.



Con la función help hacemos referencia a la función areaTraingulo, este será el resultado:



Vamos a crear una clase llamada Areas y que dichas funciones pertenezcan a dicha clase.



Como la función areaTriangulo está dentro de la clase Areas, para pedir ayuda habrá que especificar la clase seguido de un punto y después la función.

Este será el resultado:



Si queremos obtener todas las ayudas de la clase Areas, no hace falta especificar las funciones.



Este será el resultado:



Vamos a agregar un comentario para la clase.



Ahora tenemos un comentario de clase y un comentario en cada función.

Al ejecutar observaremos la siguiente información:

File Edit Selection Find View Goto Tools Project Preferences Help	_		×
Eiemplo documentacion.py * FREPL* [python] *			▼
Help on class Areas in modulemain:		an an an an an	Т
<i>class</i> Areas(<i>builtins.object</i>) Esta clase calcula las áreas de diferente fiuras geométricas		Production of the second secon	
Methods defined here:			
areaCuadrado(lado) Calcula el área de un cuadrdo elevando al cuadrado el lado pasado por parámetros			
areaTriangulo(base, altura) Calcula el área de un triángulo utilizando los parámetros base y altura			
Data descriptors defined here:			
<pre>dict dictionary for instance variables (if defined)</pre>			
<pre>weakref l</pre>			
Repl Closed			
Line 29, Column 1 Tat	o Size: 4	Python	
Observamos la documentación del apartado clase, así como la documentación de cada función que está comprendida en la clase.



En el módulo funciones.matematicas.py hemos agregado un comentario.



Este será el resultado:







DOCUMENTACIÓN Y PRUEBAS (VIDEO 76)

- Utilizar la documentación para realizar pruebas
 - o Módulo doctest

Vamos a realizar un proyecto nuevo llamado ejemplo_doctest.py



Hemos creado una función y entre """, agregamos un comentario, seguido de >>> escribimos la función y le pasamos los parámetros, en la línea inferior escribimos el resultado.

Importamos la librería doctest y ejecutamos doctest.testmod(), si ahora ejecutamos la aplicación y no observamos ningún mensaje, nos está indicando que la función funciona correctamente.

Vamos a ejecutar.



Ahora vamos a provocar un error diciendo que el valor a retornar será un 8.0 en lugar de un 9.0

5	D:\Cur	so Python\e	jemplo	_doctes	t.py (C	urso Pyt	hon) - Sul	blime Text (UN	IREGISTERED)		_		Х
<u>F</u> ile	<u>E</u> dit	<u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp				
••		ejemplo_doc	test.py	×									▼
1	Le	<i>def</i> area	Triar	ngulo	(base	,altu	ra):					Enterna .	
2	2												
3	}												
4	ļ	Calc	ula e	el áre			riángu	lo dado					
5	5		areal	rian	gulo(
6	5		-										
7													
8	3												
9		retu	<mark>rn (</mark> t	base*a	altur	a)/2							
10													
11		import d	octes	st									
12	2 0	loctest.	testm	nod()									
13													
	Line	13, Column 1							T	ab Size: 4		Python	

Cuando ejecutemos este será el resultado:



Nos está diciendo que el resultado esperado 8.0 no es correcto y que nos devuelve 9.0.

En el siguiente ejemplo la función nos retorna un string, para comprobar si esta función funcionará correctamente lo realizaremos de la siguiente forma:



En este ejemplo le estamos pasando varios test en este caso 3, vamos a ejecutar.

🗾 *REP	L* [python] (Curso	Python) - Subl	ime Text (U	INREGIST	ERED)			_		×
<u>F</u> ile <u>E</u> d	it <u>S</u> election F <u>i</u> nd	l <u>V</u> iew <u>G</u> ot	o <u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp				
<	ejemplo_doctest.p	y x Y	*REPL* [pyth	non]	×					▼
Rep	l Closed									
Lir	e 3, Column 1					Ta	b Size: 4		Python	
Al no sa	lir ningún men	saje quiere	decir qu	ie func	iona correc	tamente.				
🗾 D:\C	urso Python\ejemp	lo_doctest.py (Curso Pyth	ion) - Suł	olime Text (UN	REGISTERED)		_		Х
<u>F</u> ile <u>E</u> d	it <u>S</u> election F <u>i</u> nd	l <u>V</u> iew <u>G</u> ot	o <u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp				
 	eiemplo doctest.p	v ×								▼
1	def areaTria	angulo (bas	e.altur	va):	_	_		-	TANK IN THE	
2		<u> </u>	,						CONTROL AND CONTROL AND CONTROL AND CONTROL AND AND AND AND CONTROL AND	
3										
4	Calcula	el area d Triangulo	e un tr (36)	'iangu	Lo dado					
6	'El área	a del tria	ngulo e	es: 9.0						
7										
8		aTriangulo								
9	'El área	a del tria	ngulo e	es: 11	.0' 🔶 🚽	_				
10		Tniongulo	(0.2)							
12	'Fl área	arrianguio a del tria	ngulo e							
13			inguio c							
14										
15	return "	'El área d	lel tria	ngulo	es: " + 9	str((base*	'altura)/2)		
16										
1/	import docte	est								
10	doctest.test									
Lin	e 18, Column 18; Sav	ed D:\Curso Pyt	hon\ejemplo	o_doctest.	py (UTF-8)				Tab Si	ze: 4
Vamos	a provocar que	la segunda	tenga e	rror, ej	ecutamos:					
💋 *REP	L* [python] (Curso	Python) - Subl	ime Text (U	INREGIST	ERED)			_		Х
<u>F</u> ile <u>E</u> d	it <u>S</u> election Find	l <u>V</u> iew <u>G</u> ot	o <u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	<u>H</u> elp				
4 ►	ejemplo_doctest.p	y x V	*REPL* [pyth	non]	×					▼
*****		******			******	*****	*****	****		- 1
Filo "	eiemplo doct	ast ny"	line 8	in	main an	eaTriangu	10			
Failed	egempio_doct	сэстру , .	une o,			carrinangu.	10			
ar	eaTriangulo(4	4,5)								
Expect	ed:									
'E	l área del t	riangulo 🤅	es: 11.0	9.						
Got:	l ánga dal tu	niongulo (. 10 0	יג						
*****	**************************************	*********	**************************************	*****						
*										
1 item	s had failur	es:								
1 0										
	of 3 inma	ainarea	aTriangu	ulo						
Tes	of 3 inma t Failed :	ainarea 1 failures	aTriangu 5.	ulo						

Nos indica la línea que contiene el error que hemos indicado con una flecha.

Vamos a realizar una función que controlará el número de arroba y su posición en un correo electrónico para comprobar su error.



Ejecutamos:

5	*REPL* [python] (Curso Python) - Sublime Text (UNREGISTERED) — 🗆 🗙										
<u>F</u> ile	<u>E</u> dit	<u>S</u> election	F <u>i</u> nd	<u>V</u> iew	<u>G</u> oto	<u>T</u> ools	<u>P</u> roject	Prefere <u>n</u> ces	Help		
	• /				*RE	EPL* (pyt	hon]	×			
***	Repl	. Closed*									
	Line	3, Column 1							Tab Size: 4	Python	

Funciona correctamente.



Vamos a realizar varias pruebas, este será el resultado:



Hemos cambiado el 1 por el 0, para ver que al no funcionar bien la condición cuando ejecutemos el programa, este nos dará un error.



Esto nos demuestra que la función no controla que solo puede haber una @.





DOCUMENTACIÓN Y PRUEBAS II (VÍDEO 77)

- Utilizar la documentación para realizar pruebas
 - o Módulo doctest

Vamos a realizar un nuevo proyecto llamado ejemplo_doctest_2.py

Vamos a crear una función que retorne la raíz cuadrada de aquellos valores introducidos en la lista.



Este será el resultado:



Los valores que devuelve son double.



En este ejemplo hemos realizado una prueba en la encontramos un bucle, la opción ... (tres puntos) es como si hiciéramos un sangrado hacia la derecha, para decirles que la línea

... lista.append(i) pertenece al ciclo for.

Al ciclo for le estamos pasando una serie de valores 4, 9, 16.

Al final esta función retorna una serie de valores calculando la raíz cuadrada de tipo double, por eso se representan 2.0, 3.0, 4.0

Si al ejecutar la aplicación esta no retorna ningún mensaje, significa que todo funciona correctamente.



Vamos a comprobar que un numero negativo para la raiz cuadrada da error .



Si ejecutamos observaremos el siguiente error.



Seleccionaremos todo este texto y lo guardaremos en el portapapeles.



Pasamos a comentario el print.

Y agregamos las dos últimas líneas si no las tenemos.

Del texto que le hemos pegado, seleccionado en rojo, solo nos interesa la primera línea y la última porque en las demás pueden haber variaciones, las eliminamos he insertamos tres puntos, como diciendo que el texto intermedio sea el que sea.



Si ahora guardamos y ejecutamos y no aparece ningún mensaje, es que estamos controlando que se está produciendo un error de tipo ValueError: math domain error.

Guardamos y ejecutamos:







GENERAR EJECUTABLES. FIN DE CURSO. (VIDEO 78)

- Creación de ejecutable
 - Final de curso... No del todo

Según el tipo de sistema que estemos trabajando el ejecutable será un EXE si trabajamos en Windows, TGZ si trabajamos en Linux, DNG si trabajamos con Mac, etc.

Vamos a coger un proyecto que ya realizamos en su tiempo, en nuestro caso la calculadora.



Vamos a instalar pyinstaler, esto lo realizaremos desde la consola de Windows.



Ya hemos instalado el pyinstaller.

Ahora tenemos que ir al directorio donde tenemos la consola.



El siguiente paso será hacer ejecutable la aplicación

📼 Símbolo del sistema	-	×
		~
D:\Curso Python\graficos>pyinstaller practicaCalculadora.py_		
		Ť

Le damos a Enter.

📧 Símbolo del sistema	_		×
7572 INFO: Found binding redirects:			~
7575 INFO: Warnings written to D:\Curso Python\graficos\build\practicaCalculadora\warn-pr	actica	aCalcu	la
dora.txt			
7624 INFO: Graph cross-reference written to D:\Curso Python\graficos\build\practicaCalcul	adora	\xref-	pr
acticaCalculadora.html			
7701 INFO: checking PYZ			
7701 INFO: Building PYZ because PYZ-00.toc is non existent			
7702 INFO: Building PYZ (ZlibArchive) D:\Curso Python\graficos\build\practicaCalculadora\	PYZ-00	ð.pyz	
8253 INFO: Building PYZ (ZlibArchive) D:\Curso Python\graficos\build\practicaCalculadora\	PYZ-00	ð.pyz	co
mpleted successfully.			
8267 INFO: checking PKG			
8268 INFO: Building PKG because PKG-00.toc is non existent			
8268 INFO: Building PKG (CArchive) PKG-00.pkg			
8289 INFO: Building PKG (CArchive) PKG-00.pkg completed successfully.			
8290 INFO: Bootloader c:\users\pmver\appdata\local\programs\python\python38-32\lib\site-p	ackage	es∖PyI	ns
taller\bootloader\Windows-32bit\run.exe			
8290 INFO: checking EXE			
8291 INFO: Building EXE because EXE-00.toc is non existent			
8291 INFO: Building EXE from EXE-00.toc			
8350 INFO: Appending archive to EXE D:\Curso Python\graficos\build\practicaCalculadora\pr	actica	aCalcu	la
dora.exe			
8425 INFO: Building EXE from EXE-00.toc completed successfully.			
8429 INFO: checking COLLECT			
8429 INFO: Building COLLECT because COLLECT-00.toc is non existent			
8443 INFO: Building COLLECT COLLECT-00.toc			
14171 INFO: Building COLLECT COLLECT-00.toc completed successfully.			
D:\Curso Python\graficos>			\sim

Entramos en la carpeta dist.

Ar Evo Inicio Compartir	Vista							_		×
Anclar al Acceso rápido	V Mover a ▼ X EI Copiar a ▼	iminar 🕶 ambiar nombre	Nueva carpeta	Propiedades	Select	cionar to elecciona tir selecc	odo ar nada ión			E
Portapapeles	Organiza	ar	Nuevo	Abrir	Se	lecciona	r			
\leftarrow \rightarrow \checkmark \uparrow \square \rightarrow Este equi	← → · · ↑ → Este equipo → Disco local (D:) → Curso Python → graficos · ·						^	Buscar	en graf	icos
Acceso rápido	pycache	🧉 manzana.ico	þ	📄 practicaLabel	.py					
🔜 Escritorio 🛛 🖈 👘	dist	practica dni	.pv	practicaRadio	Button.pv					
🖊 Descargas 🛷 🔰	avion.png	practica_dni	i.pyw	primera interface.pv						
🛱 Documentos 🖈 🛛 🚺	calculadoralCO.ico	practicaCalo	uladora.py	primera_inter	face.pyw					
📰 Imágenes 🖈	calculadoralCO.jpg	practicaCalo	uladora.pyw	Prueba_File_C	Chooser.py	,				
Curso Python	EjemploEntry.py	practicaCalo	uladora.spec	澷 PruebaCheck	sButtons.p	у				
🖉 Documentor	🖥 mano.png	📄 practicaCalo	uladoraProfe.py							
23 elementos 1 elemento selec	cionado									== 📰

Abrimos la carpeta practicaCalculadora

📊 🛃 📑 🛨		Administrar	practicaCalculadora		- 🗆 ×
Archivo Inicio Compa	rtir Vista	Herramientas de aplicación			^ (?
Anclar al Copiar Pegar		ver a 🔹 🗶 Eliminar 🔹	Nueva	Seleccionar too Box Seleccionar	do nada
Acceso rápido		0	carpeta 👻 🍪		n
Portapapeles		Organizar	NUEVO ADTIT	Seleccionar	
🗲 🔶 👻 🛧 🔄 « Dis	co local (D:) >	Curso Python > graficos >	dist → practicaCalculadora →	~ Č	Ø Buscar en practica
📌 Acceso rápido	Nombre	^	Fecha de modificación	Тіро	Tamaño
📃 Escritorio 🛛 🖈	Include	2	19/08/2020 13:29	Carpeta de archivos	
🚽 Descargas 🛛 🖈	lib2to3		19/08/2020 13:29	Carpeta de archivos	
Documentos	tcl		19/08/2020 13:29	Carpeta de archivos	
🔤 Imágenes 🛷		io nvd	19/08/2020 13:29	Duthon Extension	56 KB
Curso Python	bz2.pv	rd	19/08/2020 13:29	Python Extension	77 KB
Documentos	Ctypes	.pyd	19/08/2020 13:29	Python Extension	114 KB
araficos	🗟 _decim	al.pyd	19/08/2020 13:29	Python Extension	223 KB
grancos	🌏 _eleme	nttree.pyd	19/08/2020 13:29	Python Extension	174 KB
modulos	📄 _hashli	b.pyd	19/08/2020 13:29	Python Extension	38 KB
o Creative Cloud Files	📄 _lzma.p	byd	19/08/2020 13:29	Python Extension	155 KB
Creative Cloud Files	📄 _multip	processing.pyd	19/08/2020 13:29	Python Extension	26 KB
	📝 _overla	pped.pyd	19/08/2020 13:29	Python Extension	39 KB
📥 OneDrive - Personal	📄 _queue	.pyd	19/08/2020 13:29	Python Extension	25 KB
💻 Este equipo	🧭 _socket	t.pyd	19/08/2020 13:29	Python Extension	68 KB
Descargas	Ssl.pyc		19/08/2020 13:29	Python Extension	140 KB
Documentos	🧑 _testca	pi.pyd	19/08/2020 13:29	Python Extension	91 KB
Escritorio	bace lil	.pyu branczin	19/06/2020 15:29	Archivo WinBAR 7	760 KB
		to-1 1.dll	19/08/2020 13:29	Extensión de la an	2.177 KB
imagenes	libffi-7.	dll	19/08/2020 13:29	Extensión de la ap	29 KB
	libssl-1	_1.dll	19/08/2020 13:29	Extensión de la ap	526 KB
Objetos 3D	Practic	aCalculadora.exe	19/08/2020 13:29	Aplicación	2.152 KB
Vídeos	practic	aCalculadora.exe.manifest	19/08/2020 13:29	Archivo MANIFEST	2 KB
Windows (C:)	📄 pyexpa	t.pyd	19/08/2020 13:29	Python Extension	164 KB
Disco local (D:)	🗟 python	38.dll	19/08/2020 13:29	Extensión de la ap	3.957 KB
🔿 Red	📄 select.p	byd	19/08/2020 13:29	Python Extension	24 KB
	🖄 tcl86t.d	III	19/08/2020 13:29	Extensión de la ap	1.304 KB
	ik86t.d	I	19/08/2020 13:29	Extensión de la ap	1.174 KB
	澷 unicod	edata.pyd	19/08/2020 13:29	Python Extension	1.066 KB
	S VCRUN	ITIME140.dll	19/08/2020 13:29	Extensión de la ap	82 KB

31 elementos 1 elemento seleccionado 2,10 MB

Observamos el ejecutable.

La vamos a ejecutar:

🎴 D:\C	Curso P	ython	\calcula	adora\d	iist\practicaCalculadora\practicaCalculadora.exe — 🗆 🗙	~
	Ø –			\times		
	7	8	9	/		
	4	5	6	x		
	1	2	3	-		
	0		=	+		
						~

Nos aparece con la consola por detrás.

Si queremos que se ejecute sin la consola por detrás, realizaremos la instalación de la siguiente forma:



🖼 Símbolo del sistema	-		×
D:\Curso Python\calculadora>pyinstallerwindowed practicaCalculadora.py			^
63 INFO: PyInstaller: 4.0			
63 INFO: Python: 3.8.5			
63 INFO: Platform: Windows-10-10.0.19041-SP0			
64 INFO: wrote D:\Curso Python\calculadora\practicaCalculadora.spec			
66 INFO: UPX is not available.			
71 INFO: Extending PYTHONPATH with paths			
['D:\\Curso Python\\calculadora', 'D:\\Curso Python\\calculadora']			
77 INFO: checking Analysis			
140 INFO: checking PYZ			
155 INFO: checking PKG			
156 INFO: Building because D:\Curso Python\calculadora\build\practicaCalculadora\p	ractio	aCalc	ul
adora.exe.manifest changed			
156 INFO: Building PKG (CArchive) PKG-00.pkg			
177 INFO: Building PKG (CArchive) PKG-00.pkg completed successfully.			
178 INFO: Bootloader c:\users\pmver\appdata\local\programs\python\python38-32\lib\	site-p	backag	es
\PyInstaller\bootloader\Windows-32bit\runw.exe			
178 INFO: checking EXE			
179 INFO: Building because console changed			
179 INFO: Building EXE from EXE-00.toc			
238 INFO: Appending archive to EXE D:\Curso Python\calculadora\build\practicaCalcu	ladora	a\prac	ti
caCalculadora.exe			
311 INFO: Building EXE from EXE-00.toc completed successfully.			
315 INFO: checking COLLECT			
WARNING: The output directory "D:\Curso Python\calculadora\dist\practicaCalculador	a" and	ALL	IT
S CONTENTS will be REMOVED! Continue? (y/N)y			
On your own risk, you can use the option 'noconfirm' to get rid of this question			
5008 INFO: Removing dir D:\Curso Python\calculadora\dist\practicaCalculadora			
515/ INFO: Building COLLECT COLLECT-00.toc			
6427 INFO: Building COLLECT COLLECT-00.toc completed successfully.			
D:\Curso Python\calculadora>			
			\sim



Anclar al Acceso rápido Portapapeles	Vista V Mover a • Copiar a • Copiar a • Organizar	Nueva Nuevo	iedades 🧑 Abrir	Selecciona No selecci Invertir se Seleccio	ar todo onar nada lección onar			×
← → × ↑ 🔤 > Este equi	po > Disco local (D:) > Curso Python	i > calculadora		v ⊽	<u> Р</u>	Buscar e	n calcula	ł
∧ N	ombre	Fecha de mod	ificación Tij	ро	Tama	ño		
Acceso rapido	pycache	19/08/2020 13:	41 Ca	arpeta de archiv	/05			
Escritorio	build	19/08/2020 13:	53 Ca	arpeta de archiv	/os			
- Descargas 🖈	dist	19/08/2020 13:	54 Ca	arpeta de archiv	/OS			
🖆 Documentos 🖈] practicaCaculadora.spec	19/08/2020 13:	53 Ar	rchivo SPEC		2 KB		
📰 Imágenes 💉 🚦	👌 practicaCalculadora.py	19/08/2020 13:	38 Py	thon File		6 KB		
Curso Python] practicaCalculadora.spec	19/08/2020 13:	54 Ar	rchivo SPEC		2 KB		
 Documentos elementos 1 elemento selecciona 	ionado							

La abrimos.

Ejecutamos la calculadora y ya aparece sin el fondo de la consola.

Ø –	-		×
7	8	9	/
4	5	6	x
1	2	3	-
0		=	+

Para poder hacer un único archivo ejecutable.

Nos vamos a la consola:



Ya solo tenemos un archivo ejecutable.

Es un poco más pesado.

<i>q</i> –	-		Х
7	8	9	1
4	5	6	x
1	2	3	-
0		=	+

Para que aparezca el icono que tu quieras en tu aplicación en la carpeta donde tienen el proyecto Python copia un archivo ico.

📕 🛛 🛃 🖛 🗧 calculadora				– 🗆 ×
Archivo Inicio Compartir	Vista			~ 😮
Anclar al Acceso rápido	Mover a * Eliminar *	Nueva carpeta	Seleccionar toda No seleccionar r Image: Seleccionar Image: Seleccionar <th>o nada n</th>	o nada n
Portapapeles	Organizar	Nuevo Abrir	Seleccionar	
← → ∽ ↑ 📙 > Este eq	uipo → Disco local (D:) → Curso Python	> calculadora	ٽ ~	
calculadora ^	Nombre	Fecha de modificaciór	п Тіро	Tamaño
Curso Python	pycache	19/08/2020 13:41	Carpeta de archivos	
Documentos	build	19/08/2020 13:53	Carpeta de archivos	
graficos	dist	19/08/2020 14:08	Carpeta de archivos	
o Creative Cloud File	🗔 calculadora.ico	15/08/2020 7:37	lcono	11 KB
	practicaCaculadora.spec	19/08/2020 13:53	Archivo SPEC	2 KB
o Creative Cloud File	📄 practicaCalculadora.py	19/08/2020 13:38	Python File	6 KB
OpeDrive - Persons	practicaCalculadora.spec	19/08/2020 14:03	Archivo SPEC	1 KB
7 elementos				



Pyinstaller --windowed --onefile --icon=./calculadora.ico practicaCalculadora.py

Enter.

Ya tenemos nuestra aplicación con nuestro icono.

12 - 1	dist	- 🗆	\times
Ar Evo Inicio	Compartir Vista		^ 🔮
	Iconos muy grandes Iconos grandes ▲ III I		
Panel de 📺 navegación 🛪	Iconos medianos Iconos pequeños Iconos medianos Vista Mostrar u opciones Ista Isto Detalles Iconos necessaria Iconos necessaria Iconos necessaria		
Paneles	Diseño		
$\leftarrow \rightarrow ~ \star$	🔥 « Disco local (D:) » Curso Python » calculadora » dist 🔹 🗸		
Creative Clo			
OneDrive - I	practicaCal		
💻 Este equipo	culadora.ex		
🖊 Descargas	~		
1 elemento			

Contenido

TEMARIO (VIDEO 1)	1
INTRODUCCIÓN (VIDEO 2)	2
Características	2
Instalación Python	2
SINTAXIS BÁSICA I (VIDEO3)	7
SINTAXIS BÁSICA II Tipos, operadores y variables (VIDEO 4)	
Variable	
Operadores	
Variables	
Condiciones	
SINTAXIS III FUNCIONES I (VIDEO 5)	
SINTAXIS BÁSICA IV FUNCIONES II (VIDEO 6)	
SINTAXIS BÁSICA V. LAS LISTAS (VIDEO 7)	
SINTAXIS BÁSICA VI. LAS TUPLAS (VIDEO 8)	
SINTAXIS BÁSICA VII. LOS DICCIONARIOS (VIDEO 9)	
CONDICIONALES I (VIDEO 10)	
CONDICIONALES II (VIDEO 11)	53
CONDICIONES III (VIDEO 12)	
CONDICIONALES IV (VIDEO 13)	
BUCLES I. FOR (VIDEO 14)	
Bucles	
BUCLES II BUCLE FOR (VIDEO 15)	
BUCLES II BUCLE FOR (VIDEO 16)	75
BUCLES IV. BUCLE WHILE (VIDEO 17)	
While (Sintaxis)	
BUCLES V. CONTINUE, PASS Y ELSE (VIDEO 18)	
Pass	
GENERADORES I (VIDEO 19)	
GENERADORES II (VIDEO 20)	
EXCEPCIONES I (VIDEO 21)	
EXCEPCIONES II (VIDEO 22)	
EXCEPCIONES III (VIDEO 23)	
POO I (PROGRAMACIÓN ORIENTADA A OBJETOS) (VIDEO 24)	
POO II (VIDEO 25)	
POO III (VIDEO 26)	

POO IV (VIDEO 27)	120
POO V (VIDEO 28)	126
POO VI. HERENCIA (VIDEO 29)	128
POO VII HERENCIA II (VIDEO 30)	132
POO VIII. HERENCIAS III (VIDEO 31)	136
POO IX. POLIMORFISMO (VIDEO 32)	
MÉTODOS DE CADENA (VIDEO 33)	146
MÓDULOS (VIDEO 34)	152
PAQUETES (VIDEO 35)	157
PAQUETES DISTRIBUIBLES (VIDEO 36)	164
ARCHIVOS EXTERNOS I (VIDEO 37)	173
ARCHIVOS EXTERNOS II (VIDEO 38)	179
SERIALIZACIÓN I (VIDEO 39)	
SERIALIZACION II (VIDEO 40)	189
GUARDADO PERMANENTE (VIDEO 41)	194
INTERFACES GRÁFICAS (VIDEO 42)	200
INTERFACE GRÁFICAS II (VIDEO 43)	209
INTERFACES GRÁFICAS III (VIDEO 44)	219
INTERFACES GRÁFICAS IV (VIDEO 45)	225
INTERFACES GRÁFICAS V (VIDEO 46)	235
INTERFACES GRÁFICAS VI (VIDEO 47)	242
INTERFACES GRÁFICAS VII (VIDEO 48)	247
INTERFACES GRÁFICAS VIII (VIDEO 49)	252
INTERFACES GRÁFICAS IX (VIDEO 50)	256
INTERFACES GRÁFICAS X (VIDEO 51)	
INTERFACES GRÁFICAS XI (VIDEO 52)	271
INTERFACES GRÁFICAS XII (VIDEO 53)	277
INTERFACES GRÁFICAS XIII (VIDEO 54)	285
BBDD I (VIDEO 55)	290
BBDD II (VIDEO 56)	296
BBDD III (VIDEO 57)	300
BBDD IV (VIDEO 58)	305
PRACTICA GUIADA I (VIDEO 59)	
PRÁCTICA GUIADA II (VIDEO 60)	
PRACTICA GUIADA III (VIDEO 61)	
PRÁCTICA GUIADA (VIDEO 62)	320

PRACTICA GUIADA V (VIDEO 63)	324
PRÁCTICA GUIADA VI (VIDEO 64)	
PRÁCTICA GUIADA VII (VIDEO 65)	
FUNCIONES LAMBDA (VIDEO 66)	
FUNCIÓN FILTER (VIDEO 67)	
FUNCIÓN MAP. (VIDEO 68)	355
EXPRESIONES REGULARES I (VIDEO 69)	
EXPRESIONES REGULARES II (VÍDEO 70)	
EXPRESIONES REGULARES III (VÍDEO 71)	
EXPRESIONES REGULARES IV (VÍDEO 72)	
DECORADORES I (VIDEO 73)	
DECORADORES II (VIDEO 74)	
FUNCIONES DECORADORAS II (CORRECCIÓN) (VIDEO 74b)	
DOCUMENTACIÓN (VIDEO 75)	
DOCUMENTACIÓN Y PRUEBAS (VIDEO 76)	
DOCUMENTACIÓN Y PRUEBAS II (VÍDEO 77)	403
GENERAR EJECUTABLES. FIN DE CURSO. (VIDEO 78)	409