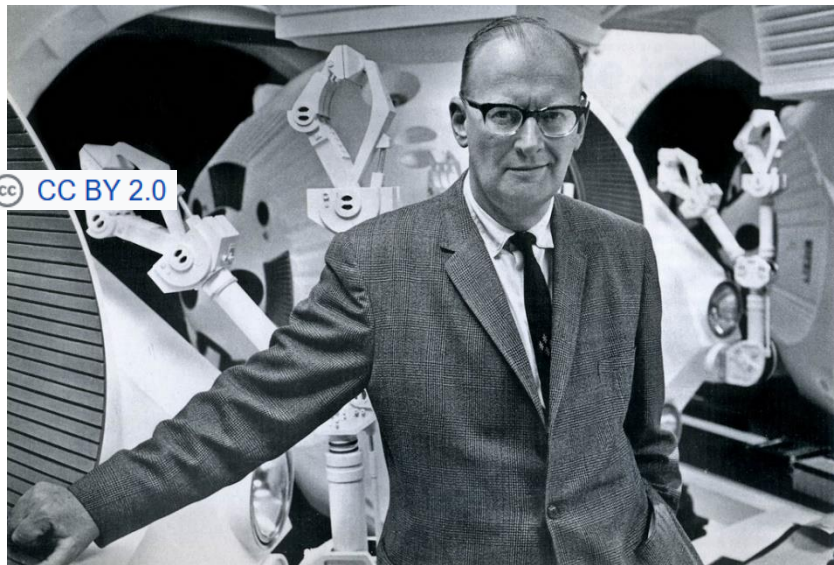


Introducción a la programación con Python

Javier García Algarra
javier.algarra@u-tad.com
2018-2023

¿Por qué estamos hoy aquí?

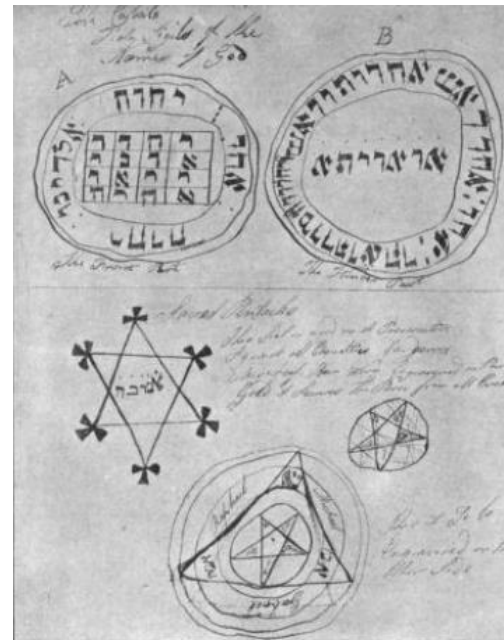


Cualquier tecnología suficientemente avanzada es indistinguible de la magia

Arthur C. Clarke (1917-2008)

¿Por qué estamos hoy aquí?

El mundo está lleno de prodigios. Si sabemos decir las palabras mágicas, en el orden exacto, ocurren fenómenos extraordinarios...



Las máquinas sólo entienden un código extraño

```
0000000 0000 0001 0001 1010 0010 0001 0004 0128
0000010 0000 0016 0000 0028 0000 0010 0000 0020
0000020 0000 0001 0004 0000 0000 0000 0000 0000
0000030 0000 0000 0000 0010 0000 0000 0000 0204
0000040 0004 8384 0084 c7c8 00c8 4748 0048 e8e9
0000050 00e9 6a69 0069 a8a9 00a9 2828 0028 fdfc
0000060 00fc 1819 0019 9898 0098 d9d8 00d8 5857
0000070 0057 7b7a 007a bab9 00b9 3a3c 003c 8888
0000080 8888 8888 8888 8888 288e be88 8888 8888
0000090 3b83 5788 8888 8888 7667 778e 8828 8888
00000a0 d61f 7abd 8818 8888 467c 585f 8814 8188
00000b0 8b06 e8f7 88aa 8388 8b3b 88f3 88bd e988
00000c0 8a18 880c e841 c988 b328 6871 688e 958b
00000d0 a948 5862 5884 7e81 3788 1ab4 5a84 3eec
00000e0 3d86 dcb8 5cbb 8888 8888 8888 8888 8888
00000f0 8888 8888 8888 8888 8888 8888 8888 0000
0000100 0000 0000 0000 0000 0000 0000 0000 0000
*
0000130 0000 0000 0000 0000 0000 0000 0000
000013e
```

Volcado de la memoria de un programa

Pero existe una solución...

Lenguaje de programación

```
#include<stdio.h>
main()
{
    int i;

    /* function returning the max between two numbers */
    int max(int num1, int num2) {

        /* local variable declaration */
        int result;

        if (num1 > num2)
            result = num1;
        else
            result = num2;

        return result;
    }

    // for (i=0;i<10;i++)
    // printf("Hello World! %d\n",i);
    printf("%d",max(3,8));
}
```



Programa traductor



```
0000: 4B C7 E0 9B 0D 98 2F FD B7 48 84 B2
000C: 6A F9 47 5E 05 09 FE E6 60 F0 E6 06
0018: 7A B4 83 3A 53 EB 96 B5 55 E9 EB 69
0024: 54 09 BC 11 FB 6C FF 59 15 E7 A1 C9
0030: DB DE 35 BC 0B 08 45 A7 94 60 B8 18
003C: 22 5B E8 50 65 FB E2 91 DC F1 4B 3F
0048: 9B AF B7 9F 31 35 C1 08 16 A3 35 D0
0054: 23 D5 EB B2 B0 B9 12 A3 00 E9 14 CD
0060: 89 B0 64 23 6A E4 78 8E 80 1C 42 38
006C: E9 2F 44 C2 77 AC 5A CA B4 B2 7B DB
```

Código máquina

Hay muchos lenguajes de programación

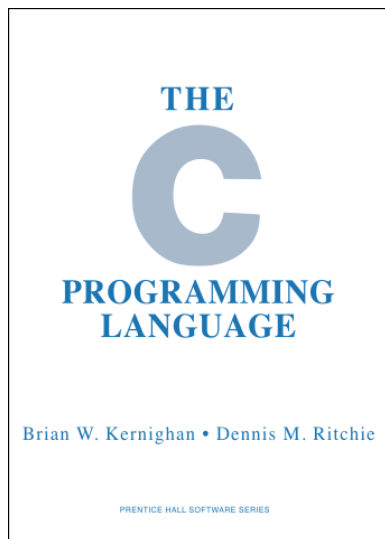
COBOL

```
IDENTIFICATION DIVISION.  
  PROGRAM-ID. HELLO-WORLD.  
*  
  ENVIRONMENT DIVISION.  
*  
  DATA DIVISION.  
*  
  PROCEDURE DIVISION.  
    PARA-1.  
      DISPLAY "Hello, world."  
*  
      EXIT PROGRAM.  
  END PROGRAM HELLO-WORLD.
```



Grace Hopper (1906-1992)

Hay muchos lenguajes de programación



Dennis Ritchie (1969-1973)

```
public void processData()
{
    do
    {
        int data = getData();
        if(data < 0)
            performOperation1(data);
        else
            performOperation2(data);
    }
    while(hasMoreData());
}
```

James Gosling, 1995



¿Por qué aprender Python?



Es multiplataforma, el mismo fichero se ejecuta sin cambios en Windows, Mac o Linux



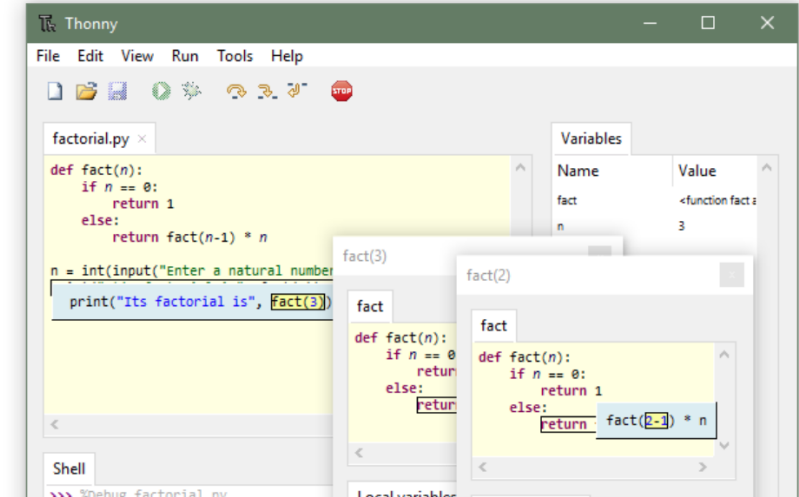
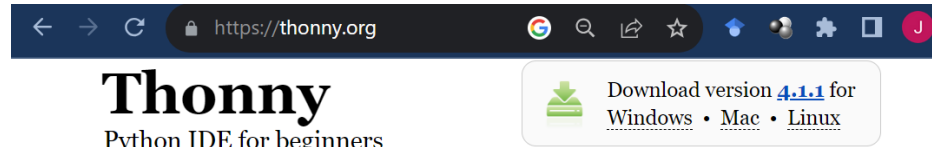
Guido van Rossum, creador de Python

Tiene una sintaxis muy limpia y elegante, se puede leer con facilidad y eso facilita la colaboración

Es Open Source y lo mantiene una organización sin ánimo de lucro, la [Python Software Foundation](https://python.org/foundation/)

Existe un catálogo enorme de paquetes, programas ya hechos por otras personas, que cubren casi todas las necesidades imaginables.

Instalación de Python



Usaremos Thonny, un entorno especialmente diseñado para principiantes, que instala Python y un editor. Conéctate a <http://thonny.org>

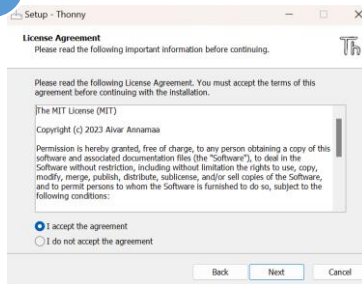
Instalación de Python

Ve a la carpeta en la que se ha descargado el fichero, haz click con el botón derecho del ratón y selecciona “instalar”. Sigue las instrucciones y NO cambies los valores por defecto

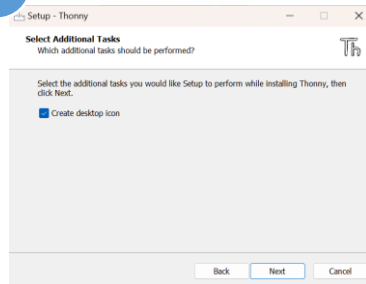
1



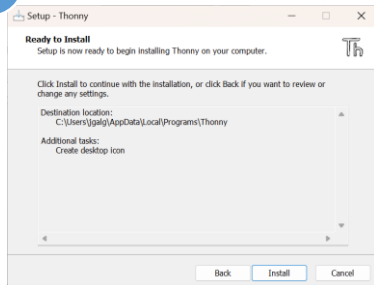
2



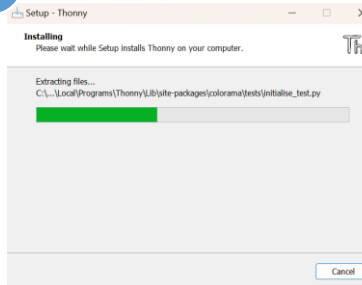
3



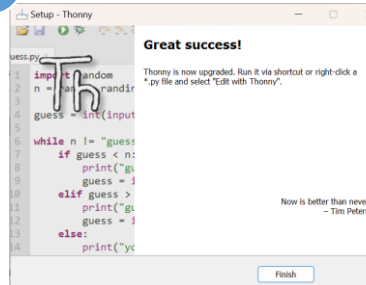
4



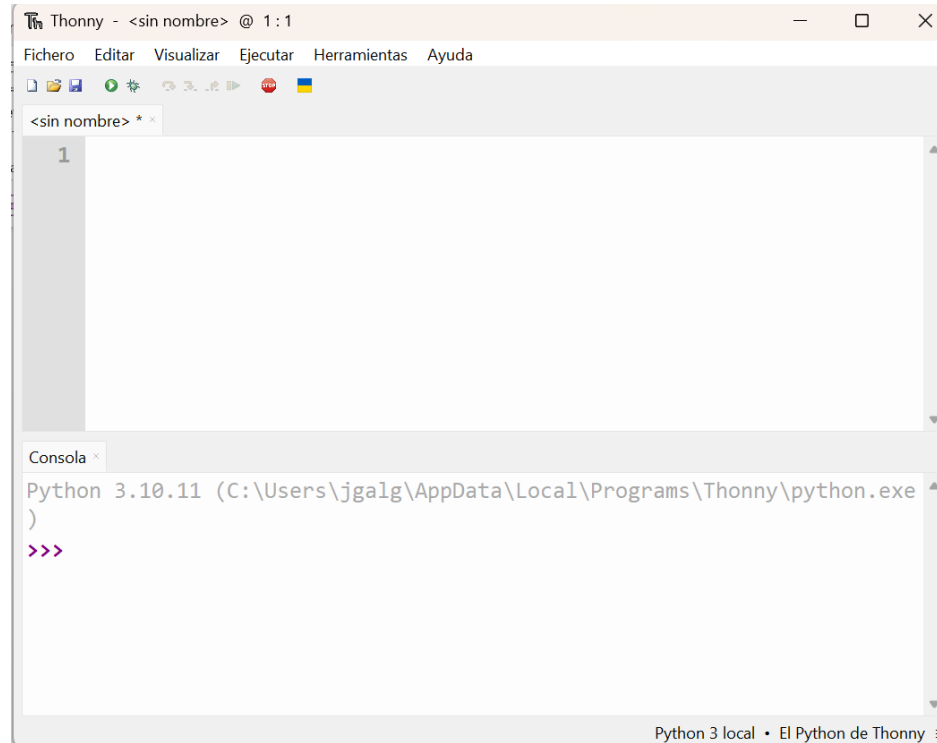
5



6



Instalación de Python



Haz doble click en el icono de *Thonny* el escritorio. ¡Ya podemos programar!

Descargar los materiales del curso

Abre la página web <https://github.com/jgalgarra/introprogpython>

The image shows a screenshot of a web browser displaying the GitHub repository page for `jgalgarra/introprogpython`. The browser's address bar shows the URL `https://github.com/jgalgarra/intropr`. The repository page includes a header with navigation links (Features, Business, Explore, Marketplace, Pricing), a search bar, and a 'Sign in' button. Below the header, the repository name `garra / introprogpython` is displayed, along with 'Watch' (1) and 'Star' (0) buttons. A large banner promotes joining GitHub today. The repository description is 'Introducción a la programación con Python'. It shows 4 commits, 1 branch, 0 releases, and 1 contributor. A 'Clone or download' button is highlighted with a red box. A modal window titled 'Clone with HTTPS' is open, showing the URL `https://github.com/jgalgarra/introprogpython` and two buttons: 'Open in Desktop' and 'Download ZIP'. A red arrow points from the 'Download ZIP' button to the text '... y luego aquí'. Another red arrow points from the text 'Haz click aquí...' to the 'Clone or download' button on the repository page.

Find file Clone or download

Clone with HTTPS ⓘ
Use Git or checkout with SVN using the web URL.
`https://github.com/jgalgarra/introprogpython`

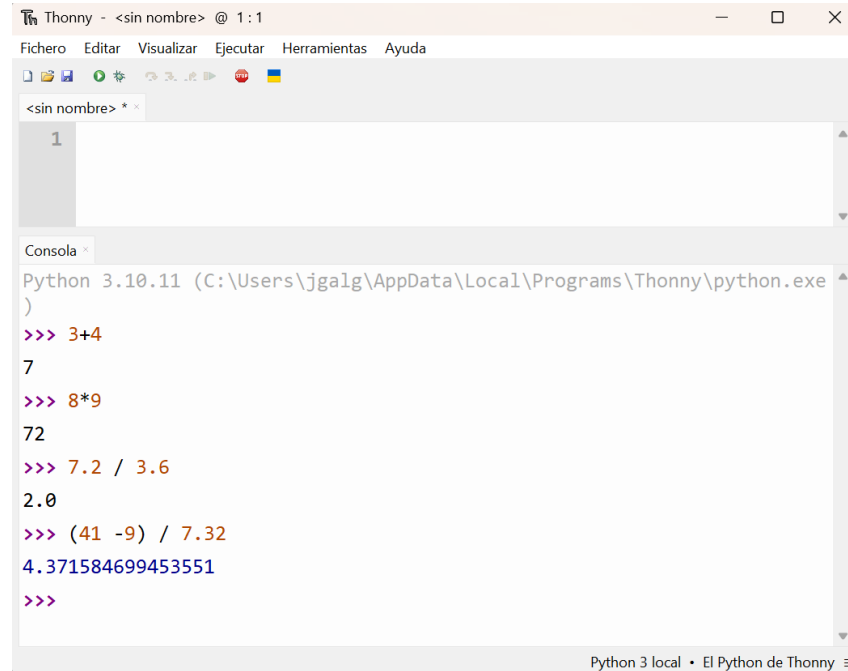
Open in Desktop Download ZIP

... y luego aquí

Haz click aquí...

Python Shell como calculadora

La parte superior de la interfaz de *Thonny* es el editor, la de de abajo la shell de comandos, donde podemos dar órdenes al ordenador. Lo más simple es usar la shell como una calculadora



The screenshot shows the Thonny Python IDE interface. The top part is the editor, which is currently empty. The bottom part is the console, which shows the results of several Python commands entered in the shell. The commands and their outputs are:

```
Python 3.10.11 (C:\Users\jgalg\AppData\Local\Programs\Thonny\python.exe)
>>> 3+4
7
>>> 8*9
72
>>> 7.2 / 3.6
2.0
>>> (41 -9) / 7.32
4.371584699453551
>>>
```

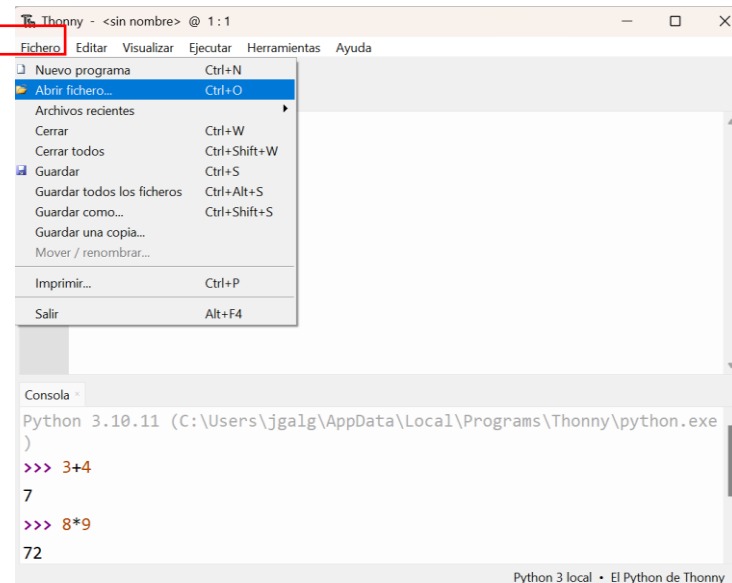
The status bar at the bottom indicates "Python 3 local" and "El Python de Thonny".

Editar un programa

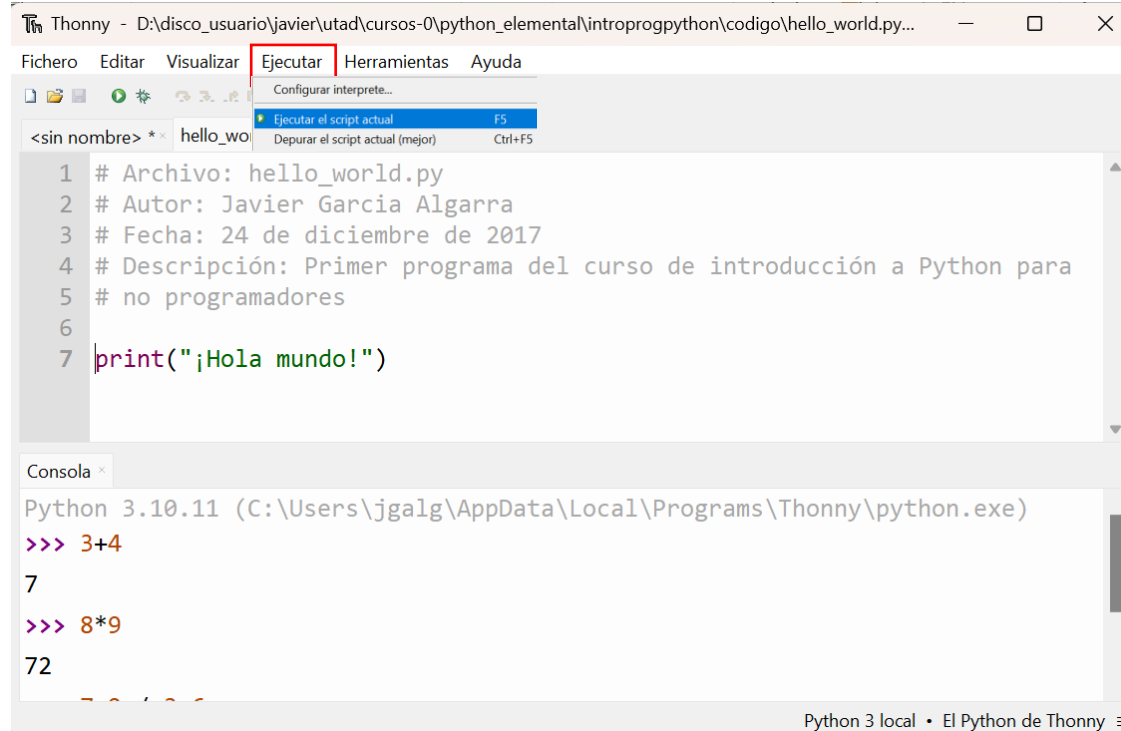
Una calculadora es interesante pero no permite ejecutar las mismas acciones una y otra vez. Para eso lo que hacemos es escribir las instrucciones en un fichero de texto al que damos un nombre conveniente. En Python todos los ficheros tienen la extensión `.py`

Este es el primer programa que se escribe cuando aprendemos un lenguaje de programación. Decimos “¡Hola!” al mundo, para que sepan que ya sabemos hacer algo nuevo. En Python es tan simple como invocar la función `print()`.

Abre el fichero `hello_world.py` que está en la carpeta `codigo`.

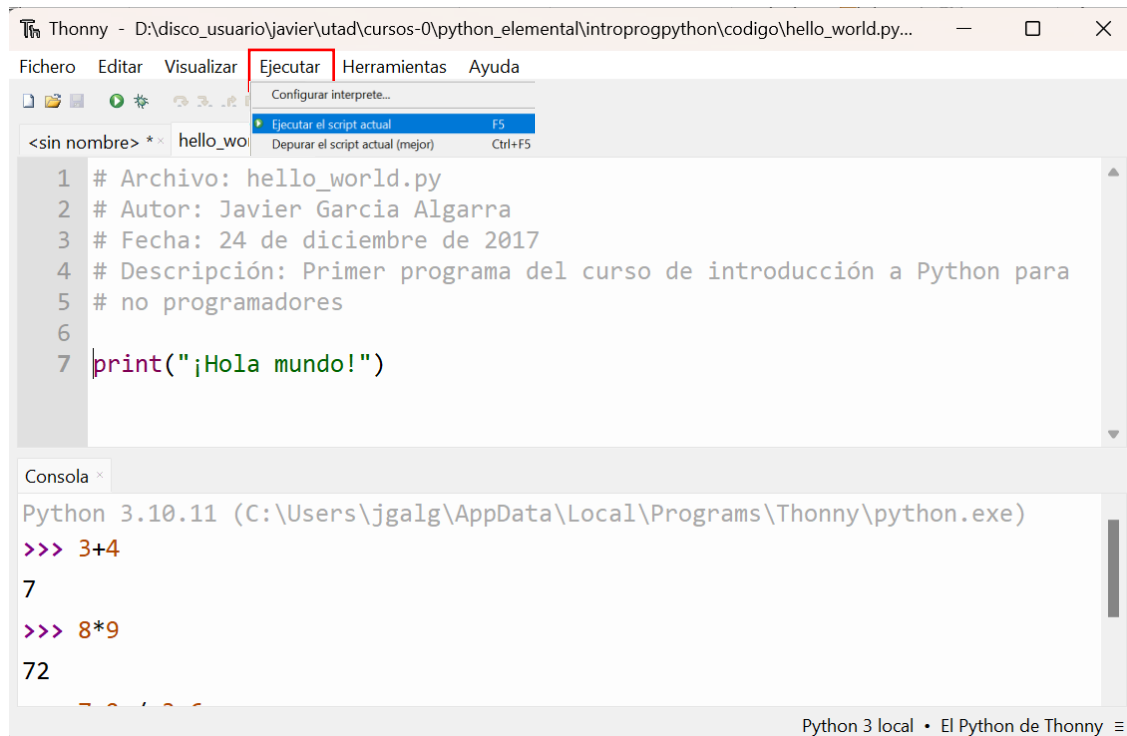


Hola mundo



Haz click en Ejecutar y después en Ejecutar el script actual

Hola mundo



Haz click en Ejecutar y después en Ejecutar el script actual

Hola mundo



The screenshot shows the Thonny Python IDE interface. The title bar indicates the file path: D:\disco_usuario\javier\utad\cursos-0\python_elemental\introprogpython\codigo\hello_world.py... The menu bar includes 'Fichero', 'Editar', 'Visualizar', 'Ejecutar', 'Herramientas', and 'Ayuda'. The toolbar contains icons for file operations, running, and debugging. The editor window shows the following code:

```
<sin nombre> * * hello_world.py <
1 # Archivo: hello_world.py
2 # Autor: Javier Garcia Algarra
3 # Fecha: 24 de diciembre de 2017
4 # Descripción: Primer programa del curso de introducción a Python para
5 # no programadores
6
7 print("¡Hola mundo!")
```

Below the editor is the 'Consola' (Console) window, which shows the execution output:

```
>>> %Run hello_world.py
¡Hola mundo!
>>>
```

The status bar at the bottom right indicates 'Python 3 local • El Python de Thonny'.

Enhorabuena, ya has escrito tu primer programa. Prueba a cambiar el mensaje

Variables

En programación, una variable es una zona de la memoria en la que se almacena un valor. Cada variable tiene un nombre, que elegimos al crearla y su contenido puede ser de diferentes tipos: un número, una letra, un texto...

Ejemplos de variables en Python:

```
edad = 21                # Número entero
peso = 77.6              # Número real
comida = "chocolate"    # Caracteres (String)

a = 4
b = 5

c = a + b                # El valor de c es 9
```

Variables

Para poder interactuar con el usuario necesitamos que pueda introducir valores, que almacenamos en variables. Por ejemplo, podemos pedirle el nombre, la edad y la estatura.

Para eso usamos la instrucción `input()`

```
tu_ciudad = input("¿En qué ciudad vives?: ")
```

Abre el fichero `variables_1_con_lectura.py`

Listas

Una lista es una variable que contiene varios valores. En Python los elementos de una lista se numeran desde 0 a n-1. Esto puede parecer extraño pero es muy común en programación.

```
# También podemos tener listas de números
lista_b = [8, 9, 10, 4]
# Python es muy inteligente y puede sumar todos los elementos de una vez
suma_lista_b = sum(lista_b)
print("Los valores de la lista numérica son ", lista_b, "y su suma es", suma_lista_b)
# También sabe hallar el máximo y el mínimo
maximo = max(lista_b)
minimo = min(lista_b)
```

Abre el fichero `variables_2_listas.py`

Variables booleanas

Las variables booleanas solo pueden tomar dos valores: `True` o `False`

Se emplean para hacer comparaciones. Por ejemplo la expresión `5 > 3` es `True` porque el operando de la izquierda es mayor que el de la derecha

Las condiciones booleanas permiten tomar decisiones y hacer que el programa ejecute unas instrucciones u otras. Para eso se usa la construcción `if ... else`

```
if ( a > b ):  
    print("La variable a es mayor que b")  
else:  
    print("La variable a no es mayor que b")
```

Abre el fichero `variables_3_booleanas.py`

Bucles

A menudo necesitamos repetir una instrucción o un conjunto de instrucciones. Para ello, los lenguajes de programación usan bucles. En Python hay dos tipos, el bucle que se repite un número de veces dado y otro que continúa de manera indefinida hasta que se cumple una condición.

```
a = 1
while ( a < 7 ):          # Este bucle se repetirá hasta
    a = a + 1             # que la variable a valga 7

# Este otro bucle recorre todos los elementos de la lista
for j in ["a","e","i"]:
    print(j)
```

Abre el fichero `bucles.py`

Funciones

Una función es un conjunto de líneas de código que pueden invocarse en cualquier punto del programa. En general es muy mala práctica repetir código porque si detectamos que hay algo mal tendríamos que cambiarlo todas las veces que aparece en el programa

```
hipotenusa_1 = math.sqrt(cateto_a**2+cateto_b**2)
hipotenusa_2 = math.sqrt(cateto_c**2+cateto_d**2)
```

NO

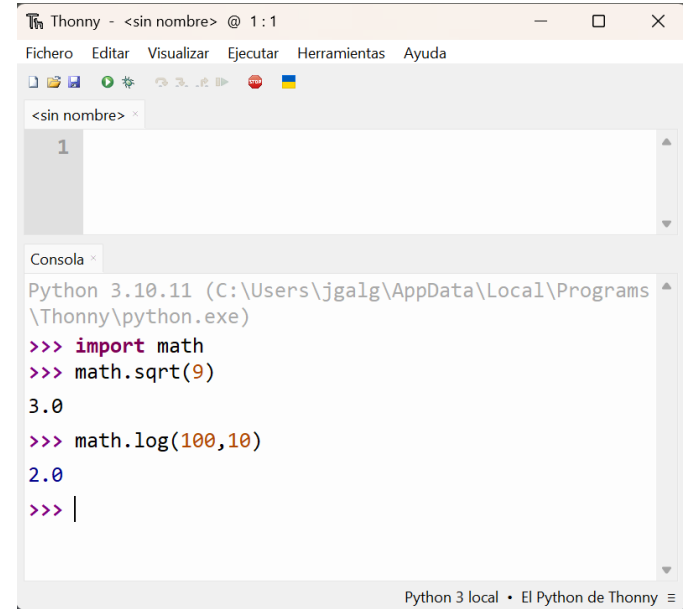
```
def pitagoras(ca,cb):
    hip = math.sqrt(ca**2+cb**2)
    return(hip)
```

```
hipotenusa_1 = pitagoras(cateto_a,cateto_a)
hipotenusa_2 = pitagoras(cateto_c,cateto_d)
```

SI

Adivina el número

Hay operaciones que requieren usar un paquete software. Como la raíz cuadrada. Para ello hay que importar el paquete con la instrucción `import` y luego llamar a la función que realiza la operación. Aquí puedes ver como se calcula la raíz cuadrada de 3 o el logaritmo de 100 en base 10.



The screenshot shows the Thonny Python IDE interface. The top menu bar includes 'Fichero', 'Editar', 'Visualizar', 'Ejecutar', 'Herramientas', and 'Ayuda'. Below the menu is a toolbar with icons for file operations and execution. The main editor window displays a script with the following code:

```
1
```

The console window at the bottom shows the execution of the script using Python 3.10.11. The output is as follows:

```
Python 3.10.11 (C:\Users\jgalg\AppData\Local\Programs\Thonny\python.exe)
>>> import math
>>> math.sqrt(9)
3.0
>>> math.log(100,10)
2.0
>>> |
```

The status bar at the bottom right indicates 'Python 3 local • El Python de Thonny'.

Adivina el número

Con todo lo que ya hemos aprendido podemos desarrollar nuestro primer juego. Es muy sencillo, no tiene gráficos ni sonido, ¡pero es el primer juego que vas a desarrollar!

La máquina pensará un número secreto entre 1 y 100 y el jugador tendrá que adivinarlo. Gana el que lo consiga con menos intentos. Para este juego usaremos el paquete `random`, que genera números aleatorios



Abre el fichero `adivina_el_numero_esqueleto.py`

Sonido

Los videojuegos actuales tienen unos efectos sonoros que no tienen nada que envidiar a los de las grandes producciones de Hollywood. Sin embargo, hubo una época en la que la capacidad del hardware era muy limitada y sólo se podían producir beeps.

Los sintetizadores primitivos solo dejaban indicar la frecuencia y duración y eso daba lugar a lo que se conoce como el sonido 8 bits.

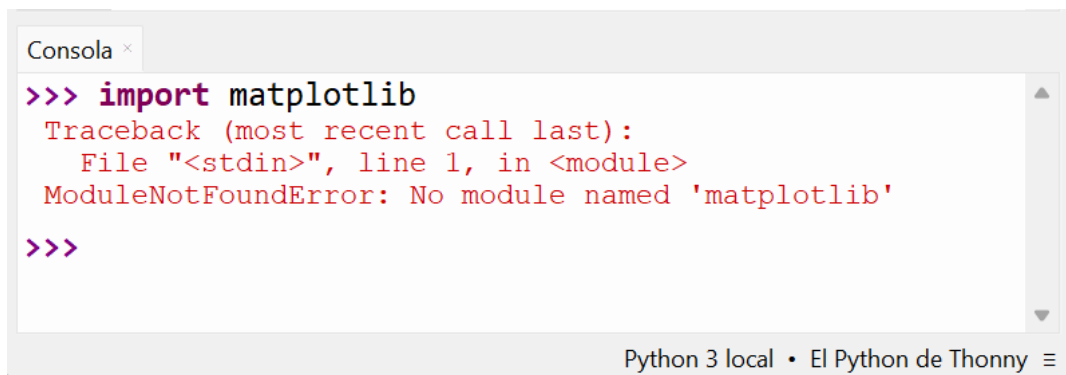
Existe una librería en Python que emula este funcionamiento, pero antes vamos a escuchar uno de los grandes clásicos de los 80...



Abre el fichero `audio.py`

Gráficos

Python dispone de distintos tipos de gráficos. El paquete `matplotlib` permite generar gráficas estáticas. Este paquete no se descarga con la distribución estándar de Thonny. Obtendrás el siguiente mensaje de error

A screenshot of the Thonny Python IDE's console window. The window has a title bar that says "Consola x". Inside the console, the following text is displayed: a prompt ">>>" followed by the command "import matplotlib" in purple. Below this, a red traceback message is shown: "Traceback (most recent call last):", "File "<stdin>", line 1, in <module>", and "ModuleNotFoundError: No module named 'matplotlib'". This is followed by another prompt ">>>". At the bottom right of the console window, the text "Python 3 local • El Python de Thonny" is visible next to a menu icon.

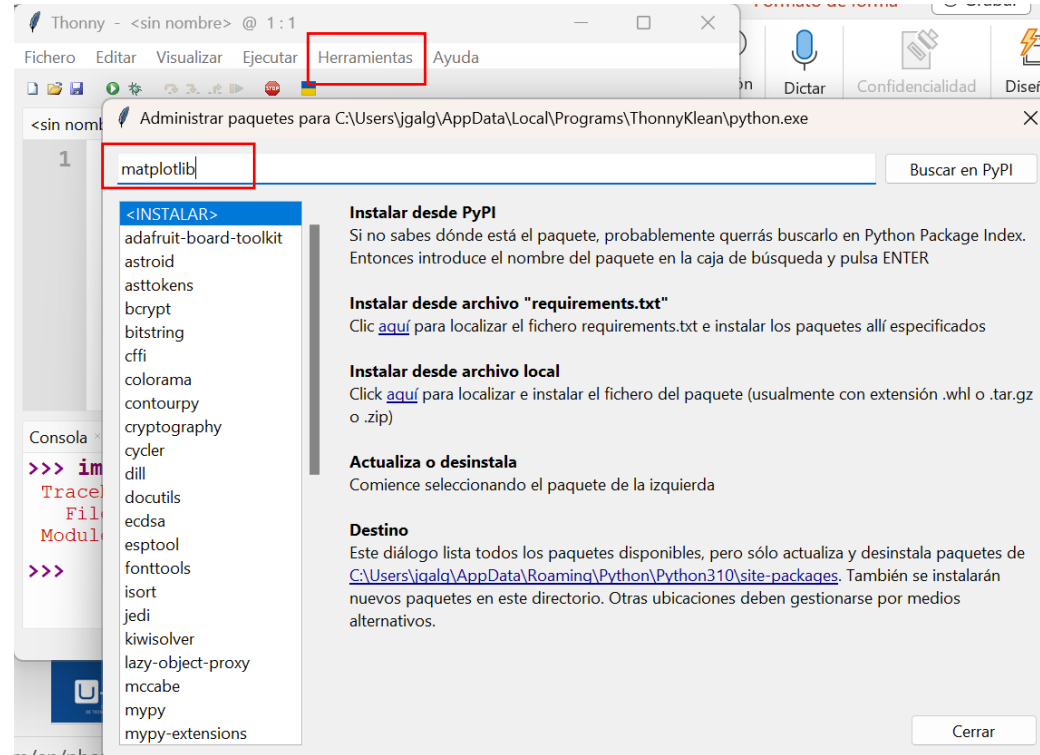
```
>>> import matplotlib
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'matplotlib'

>>>
```

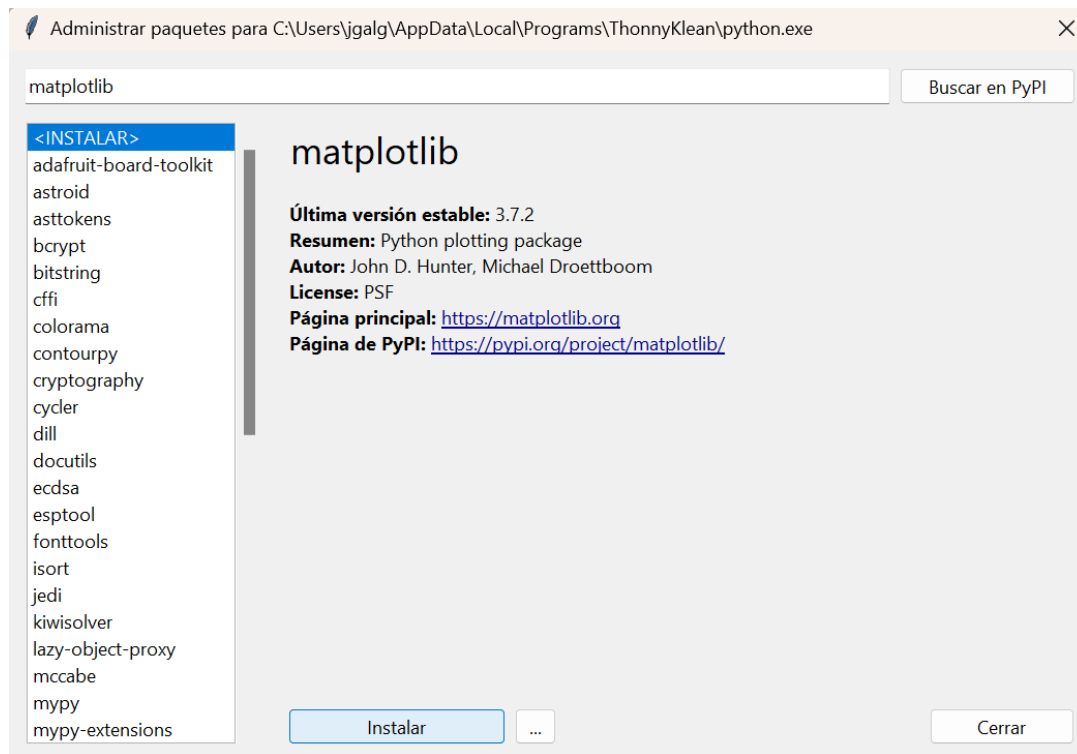
Python 3 local • El Python de Thonny

Gráficos

Para solucionarlo, lo instalaremos, desde la opción principal del menú Herramientas, Gestionar Paquetes. Escribimos matplotlib y buscamos en PyPi



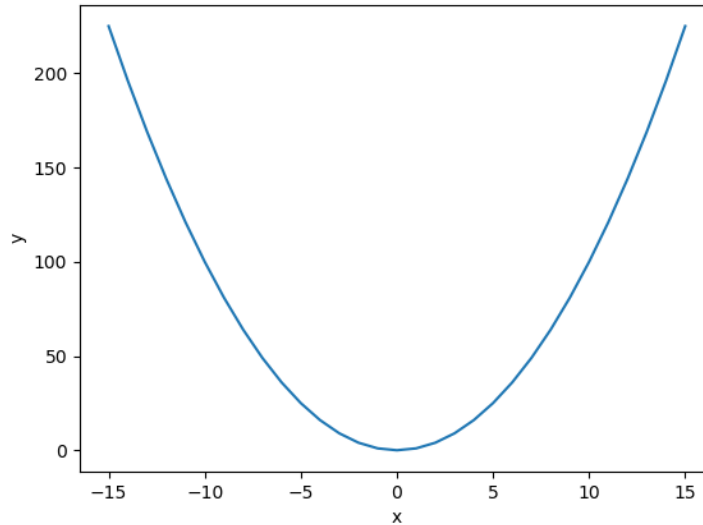
Gráficos



Click en esta opción

Gráficos

¿Recuerdas las fórmulas de la recta, la parábola o la circunferencia que viste en matemáticas?



Abre los ficheros `pinta_....py`

Gráficos

Un matemático francés del siglo XIX descubrió que cualquier forma de onda puede conseguirse sumando funciones $\sin(x)$. Esto te puede parecer extraño e incluso aburrido, pero es la base del funcionamiento de los ficheros de audio y video digitales. Vamos a comprobar si es cierto...



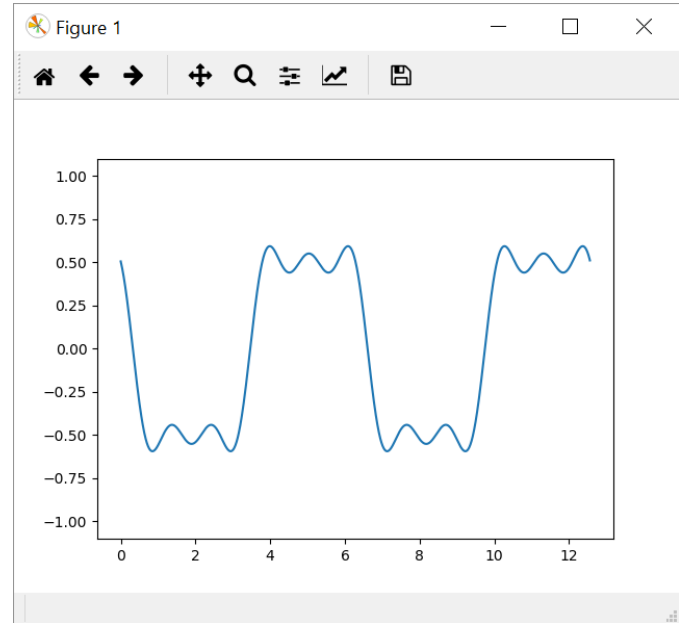
Joseph Fourier (1768-1830)

Animación

Los gráficos estáticos son interesantes pero, ¡resultan mucho más divertidos los animados!.

Eso lo podemos conseguir con las funciones de animación de matplotlib.

Abre `animacion.py`

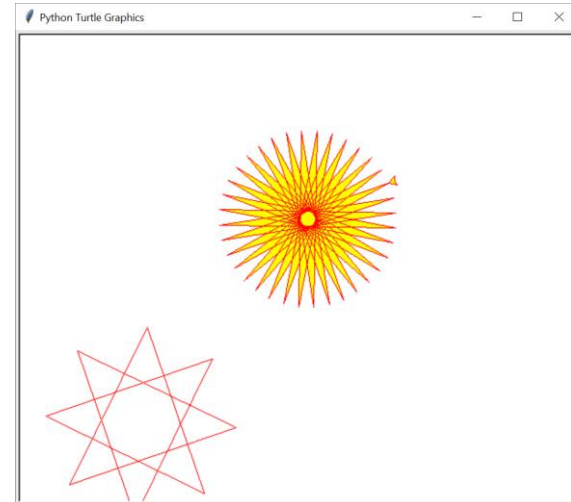


La tortuga

La tortuga nació de una idea para enseñar a programar a niños, con un lenguaje que se llamaba PROLOG. Es una flecha que se mueve por la pantalla gráfica según las órdenes que recibe, del tipo “avanza”, “gira”, etc...

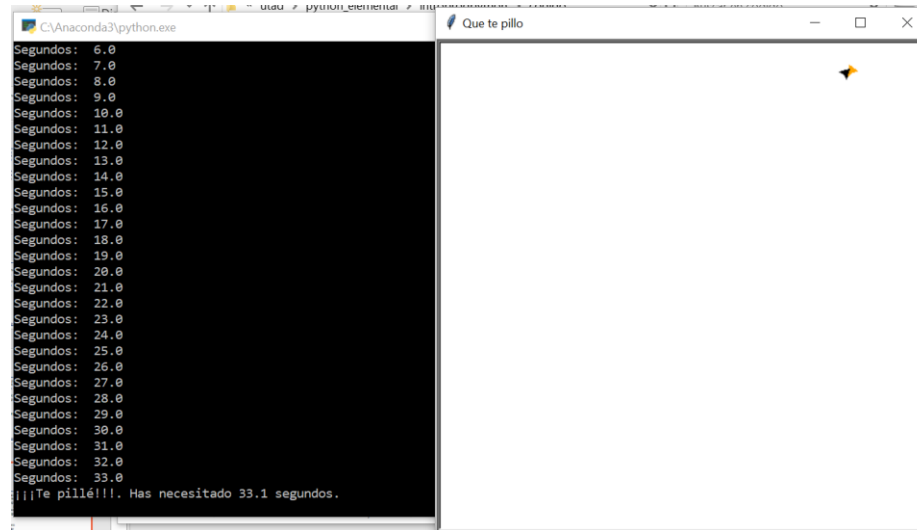
Hay una adaptación magnífica en Python que nos va a servir para construir nuestro primer gran videojuego

Abre `tortuga_1_movimiento.py`



Que te pillo

¡Ya estamos preparados para desarrollar nuestro primer videojuego! La tortuga cazadora tiene que atrapar a una veloz presa que se mueve al azar. Gana quien lo consiga en menos tiempo. Abre el fichero `quetepillo.py` con las instrucciones detalladas.



Para seguir aprendiendo

En este curso solo hemos podido explorar la superficie del planeta Python. Para aprender a programar hay que practicar, practicar, practicar... Hay muchos recursos gratuitos para convertirte en un gran programador Python.

Esta página es excelente, y puede servirte para reforzar los conceptos que hemos visto en el curso.

 **Introducción a la programación con Python**

Estos apuntes del **Curso de iniciación a la programación en Python** se están impartiendo en la segunda mitad del curso 2022/2023 en el módulo **Lenguaje de Marcas y Sistemas de Gestión de la Información** del ciclo formativo **Administración de Sistemas Informáticos en Red** (ASIR) en el IES Abastos de Valencia (España).

- [Presentación y licencias](#)

Lecciones Python

Hola, mundo

- [Elementos de un programa](#)
- [Programa básico](#)

Tipos de datos

- [Números y operaciones aritméticas](#)
- [Cadenas de texto \(1\)](#)
- [Variables \(1\)](#)
- [Cadenas de texto \(2\)](#)
- [Booleanos](#)
- [Secuencias: Tuplas, Rangos, Listas](#)
- [Variables \(2\)](#) - [Variables \(3\)](#)

Entrada y salida

- [Salida por pantalla](#)
- [Entrada por teclado](#)
- [Ficheros](#)

Estructuras de control

- [Sentencias condicionales: if...elif...else...](#)
- [Iteraciones: bucle for \(1\)](#) - [bucle for \(2\)](#)
- [Iteraciones: bucle while](#)

Exámenes

- [Exámenes](#)

Ejercicios

- [Puesta en marcha](#)
- [Variables, E/S y Operaciones aritméticas](#)
- [Expresiones lógicas \(1\)](#) - [Expresiones lógicas \(2\)](#)
- [Sentencias condicionales: if elif else \(1\)](#) - [if elif else \(2\)](#) - [Minijuegos \(1\)](#)
- [Listas de enteros: el tipo range\(\)](#)
- [Iteraciones: Minijuegos \(2\)](#) - [bucle for \(1\)](#) - [bucle for \(2\)](#) - [Sucesiones](#)
- [Ficheros \(1\)](#)
- [Listas \(1\)](#) - [Listas \(2\)](#)
- [Bucle while \(1\)](#) - [Bucle while \(2\)](#) - [Bucle while \(3\)](#)
- [Funciones \(1\)](#) - [Funciones \(2\)](#)
- [Repaso \(1\)](#) - [Repaso \(2\)](#)
- [Gráficos SVG: Coordenadas - viewBox](#)
- [Ficheros SVG:](#)
 - [Ficheros \(1\)](#) - [Ficheros \(1.B\)](#)
 - [Ficheros \(2\)](#) - [Ficheros \(3\)](#) - [Ficheros \(4\)](#)
- [Juegos \(pygame\): Pong \(1\)](#) - [Pong \(2\)](#)
- [Más ejercicios \(de exámenes anteriores\)](#)
- [Más ejercicios \(de años anteriores\)](#)

Software

- [Lenguajes de programación](#)
- [Historia de Python](#)
- [Instalación](#)
- [El entorno interactivo de Python](#)
- [Ejecutar programas](#)
- [IDLE, entorno de desarrollo](#)

Visual Studio Code

- [Configuración recomendada](#)

mclibre Python testing

- [Presentación](#)
- [Instalación](#)
- [Ejemplos](#)

Otros

- [Libro de estilo](#)
- [Documentación](#)
- [Retos de programación](#)
- [Temas obsoletos](#)
- [Lecciones impartidas en clase](#)

<http://www.mclibre.org/consultar/python/index.html>

Para seguir aprendiendo



Este libro gratuito es ya un clásico para aprender Python en español. Contiene material mucho más avanzado que el que hemos visto en este curso, pero te servirá de referencia.