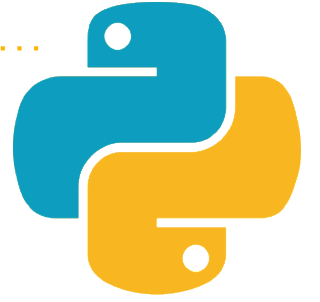


Unidad No.2 Lenguajes Altamente Dinámicos/débilmente Tipados

TI2011 – Introducción a la Programación



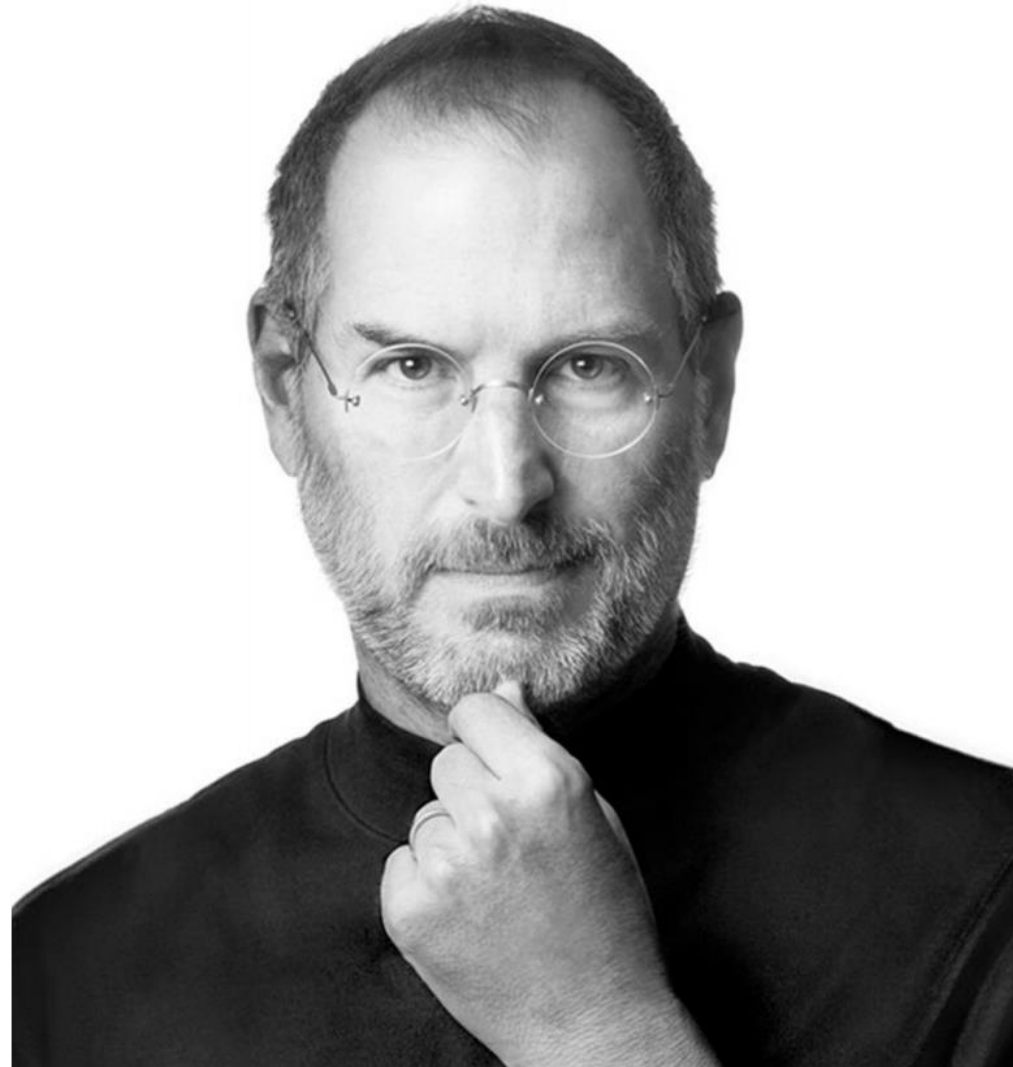
Motivación

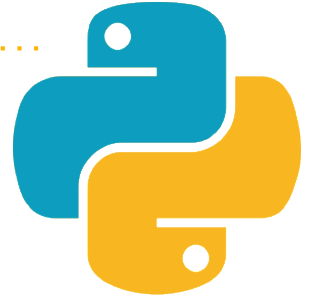
Reflexión

Una posible respuesta que nos hace sentido, es que la programación es útil en diversas áreas del conocimiento, desarrolla el pensamiento lógico y potencia una de las habilidades fundamentales que es la resolución de problemas.

“Todo el mundo debería aprender a programar un ordenador, porque te enseña a pensar”

Steve Jobs



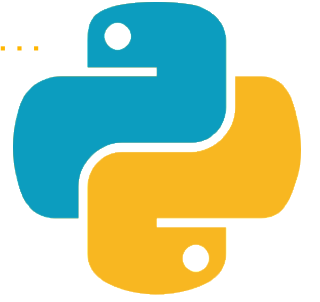


Reflexión

¿Por qué deberíamos aprender a programar en Python?

En particular se ocupará el lenguaje de programación Python, porque está presente en muchas aplicaciones y servicios que utilizamos de manera habitual, por ejemplo, Smartphone, computadores, video juegos, YouTube y Google entre muchos otros usos.

- ❑ Es uno de los lenguajes de programación más populares en el mundo
- ❑ Simple y fácil de aprender
- ❑ Usado en muchos contextos (especialmente en ciencia de datos, que luego explicaremos por qué es importante)



¿Qué es Python?

- Lenguaje interpretado
- Independiente de la plataforma
- Orientado a Objetos

Instalación del Interprete



Instalación

Para programar en Python es necesario instalar el interprete de Python.

El Interprete tiene una versión para cada sistema operativo.

Puede ser descargado desde

<https://www.python.org/downloads/>



Donate



Search

About

Downloads

Documentation

Community

Success Stories

News

Events

Download the latest version for Windows

Download Python 3.7.3

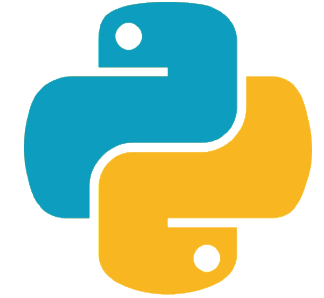
Looking for Python with a different OS? Python for [Windows](#),
[Linux/UNIX](#), [Mac OS X](#), [Other](#)

Want to help test development versions of Python? [Pre-releases](#),
[Docker images](#)

Looking for Python 2.7? See below for specific releases

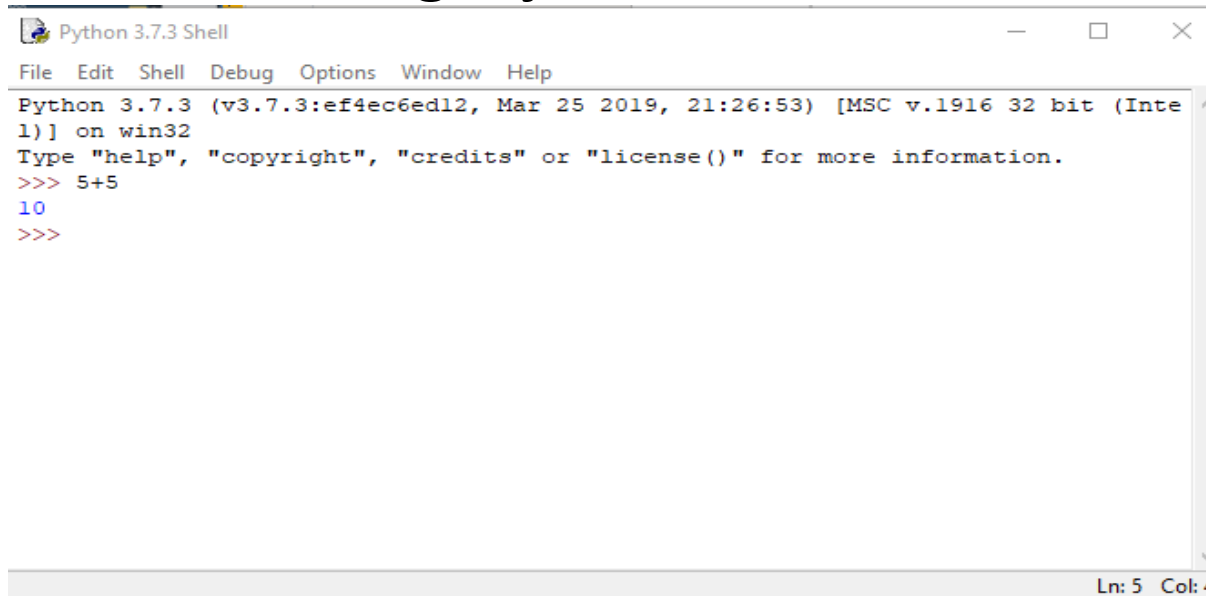


Formas de Ejecutar Código



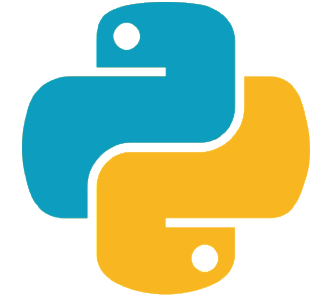
Podemos ejecutar código en Python mediante 2 formas:

A.Utilizando el Intérprete: Al utilizar el IDLE de Python se abrirá el terminal queda en modo de espera por una instrucción de lenguaje:

A screenshot of the Python 3.7.3 Shell window. The window has a title bar that says "Python 3.7.3 Shell" and standard window controls. Below the title bar is a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main area of the window shows the Python prompt "Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)] on win32". Below this, it says "Type 'help', 'copyright', 'credits' or 'license()' for more information." The prompt ">>>" is followed by the input "5+5", which has been executed, resulting in the output "10". The prompt ">>>" is now on a new line. At the bottom right of the window, the status bar shows "Ln: 5 Col: 4".

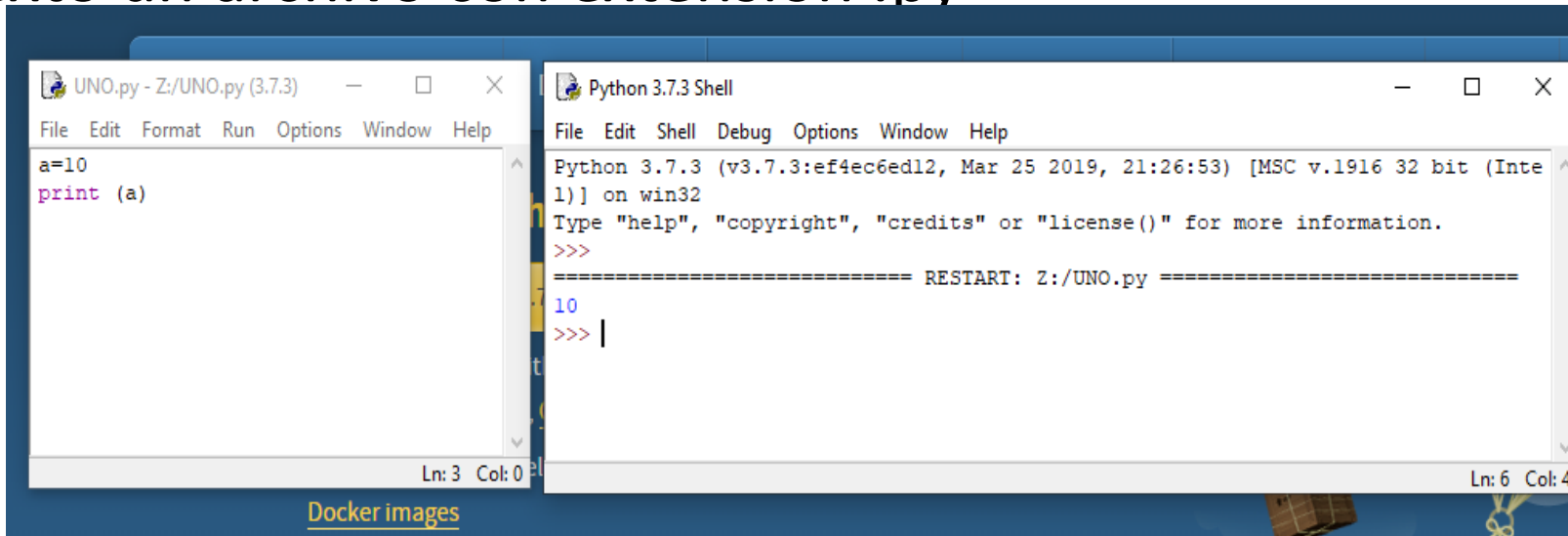
```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 5+5
10
>>>
```


Formas de Ejecutar Código

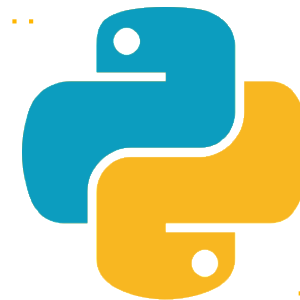


Podemos ejecutar código en Python mediante 2 formas:

B.Creando un fichero Python: Creamos con un editor de texto un archivo con extensión .py

A screenshot of a Python IDE interface. On the left, a window titled 'UNO.py - Z:/UNO.py (3.7.3)' shows a text editor with the code 'a=10' and 'print (a)'. The status bar at the bottom of this window indicates 'Ln: 3 Col: 0'. On the right, a window titled 'Python 3.7.3 Shell' shows the Python interpreter's prompt '>>>'. The status bar at the bottom of this window indicates 'Ln: 6 Col: 4'. The background of the IDE is dark blue with a subtle pattern.

Bases del Lenguaje



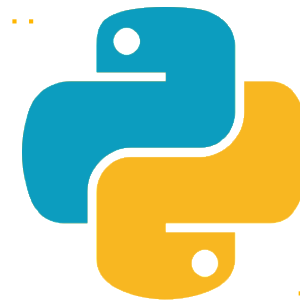
Tipos de Datos

Python es un lenguaje que implementa el polimorfismo, por lo cual no es requerido dar un tipo específico a una variable a la hora de definirla.

Sin embargo, para hacer operaciones de un tipo de dato específico, se debe indicar directamente al interprete como debe el "tratar" a la variable.

Dentro de los tipos de datos existentes están:

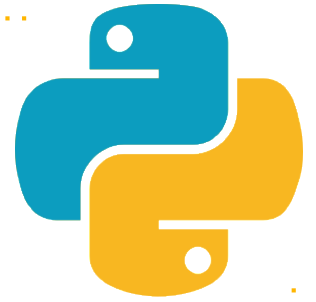
Tipo	Descripción
String (str)	Cadenas de texto
Int (int)	Números enteros
Float (float)	Números con decimales
Boolean (bool)	Lógicos, puede tener valores True o False



Operadores Aritméticos

Python define los operadores aritméticos básicos, existentes en cualquier lenguaje y, adicionalmente, agrega variaciones de los mismos:

Operador	Descripción	Ejemplo
+	Suma	<code>r = 3 + 2 # r es 5</code>
-	Resta	<code>r = 4 - 7 # r es -3</code>
*	Multiplicación	<code>r = 2 * 6 # r es 12</code>
**	Exponente	<code>r = 2 ** 6 # r es 64</code>
/	División	<code>r = 3.5 / 2 # r es 1.75</code>
//	División entera	<code>r = 3.5 // 2 # r es 1.0</code>
%	Módulo	<code>r = 7 % 2 # r es 1</code>



Operadores Relacionales

Python define los siguientes operadores relacionales, los cuales siempre devuelven un valor Booleano (True o False)

Operador	Descripción	Ejemplo
<code>==</code>	¿son iguales a y b?	<code>r = 5 == 3 # r es False</code>
<code>!=</code>	¿son distintos a y b?	<code>r = 5 != 3 # r es True</code>
<code><</code>	¿es a menor que b?	<code>r = 5 < 3 # r es False</code>
<code>></code>	¿es a mayor que b?	<code>r = 5 > 3 # r es True</code>
<code><=</code>	¿es a menor o igual que b?	<code>r = 5 <= 5 # r es True</code>
<code>>=</code>	¿es a mayor o igual que b?	<code>r = 5 >= 3 # r es True</code>

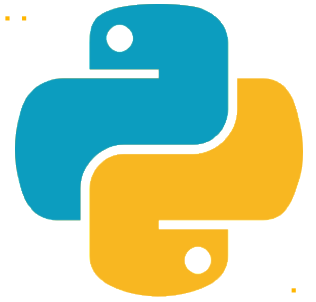


El Primer Hola Mundo

```
print("Hola Mundo")
```

La función `print` permite mostrar por pantalla un resultado al usuario.

Python trata siempre de hacer las cosas más fáciles para el desarrollador, razón por la cual el contenido de `print` puede ser, un número, una cadena, un decimal, un booleano, lo que sea.

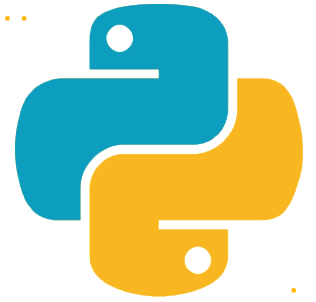


El Primer Hola Mundo

La función `input()` permite obtener texto escrito por teclado. Al llegar a la función, el programa se detiene esperando que se escriba algo y se pulse la tecla **Intro**, como muestra el siguiente ejemplo:

```
nombre = input("Ingrese Nombre")  
print("Hola " + nombre)
```

De forma predeterminada, la función `input()` convierte la entrada en una cadena, aunque escribamos un número. Si intentamos hacer operaciones, se producirá un error.



El Primer Hola Mundo

```
nombre = input("Ingrese Nombre")  
print("Hola {0}".format(nombre))
```

La función format permite unir texto para ser desplegado o utilizado dentro de una variable.
Es el equivalente a la concatenación con el signo +



Conversión de tipos

- Convertir a cadena de texto (string): `str()`
- Convertir a entero: `int()`
- A punto flotante (números decimales): `float()`
- A booleano: `bool()`