

pythonTM

Yorman Robles



Una introducción a Python

1	code	output
2	while a < 7 :	
3	if(a % 2 == 0):	
4	print(a, "is even")	
5	else:	
6	print(a, "is odd")	
7	a += 1	



¿Qué es Python?



Python es un lenguaje de programación de propósito general muy poderoso y flexible, a la vez que sencillo y fácil de aprender, y fue creado a principios de los noventa por Guido van Rossum en los Países Bajos.

Es un lenguaje de alto nivel, que permite procesar fácilmente todo tipo de estructuras de datos, tanto numéricas como de texto.

Es software libre, y está implementado en todas las plataformas y sistemas operativos habituales.



Historia de Python



Guido Van Rossum es el creador y responsable de que Python exista. Se trata de un informático de origen holandés que fue el encargado de diseñar Python y de pensar y definir todas las vías posibles de evolución de este popular lenguaje de programación.

empezando su desarrollo en 1989 y empezando a implementarlo en febrero de 1991, momento en el que se publicó la primera versión pública: la 0.9.0.

Esta primera versión ya incluía clases con herencias, manejo de excepciones, Funciones y una de las características principales de Python: funcionamiento modular. Esto permitía que fuese un lenguaje mucho más limpio y accesible para la gente con pocos conocimientos de programación. Una característica que se mantiene hasta el día de hoy.



Características de Python



Las características del lenguaje de programación Python se resumen a continuación:

- Es un lenguaje interpretado, no compilado que usa tipado dinámico, fuertemente tipado (el tipo de valor no cambia repentinamente).
- Es multiplataforma, lo cual es ventajoso para hacer ejecutable su código fuente entre varios sistemas operativos.
- Es un lenguaje de programación multiparadigma, el cual soporta varios paradigmas de programación como orientación a objetos, estructurada, programación imperativa y, en menor medida, programación funcional.
- En Python, el formato del código (p. ej., la indentación) es estructural.



Las características del lenguaje de programación Python se resumen a continuación:

- El código fuente es interpretado sin necesidad de crear programas externos que usa tipado dinámico, fuertemente tipado (el tipo).
- Es multiplataforma, lo cual es ventajoso para hacer ejecutable su código fuente entre varios sistemas operativos.
- Es un lenguaje de programación multiparadigma, el cual soporta varios paradigmas de programación como orientación a objetos, estructurada, programación imperativa y, en menor medida, programación funcional.
- En Python, el formato del código (p. ej., la indentación) es estructural.



Las características del lenguaje de programación Python se resumen a continuación:

- Es un lenguaje de programación que usa **El código fuente es interpretado sin necesidad de crear programas externos** (el tipo de variables se comprueba durante la ejecución).
- Es multiplataforma, lo cual es ventajoso para hacer ejecutable su código fuente entre varios sistemas operativos.
- Es un lenguaje de programación multiparadigma, el cual soporta varios paradigmas de programación como orientación a objetos, estructurada, programación imperativa y, en menor medida, programación funcional.
- En Python, el formato del código (p. ej., la indentación) es estructural.



Las características del lenguaje de programación Python se resumen a continuación:

- Es un lenguaje de programación interpretado sin necesidad de crear programas externos que usa el tipo de las variables se comprueba durante la ejecución.
- Es multiplataforma, lo cual es ventajoso para hacer ejecutable su código fuente entre varios sistemas operativos.
- Es un lenguaje de programación que soporta varios paradigmas de programación como el código se organiza en clases que se relacionan, estructurada, programación imperativa y, en menor medida, programación funcional.
- En Python, el formato del código (p. ej., la indentación) es estructural.



Las características del lenguaje de programación Python se resumen a continuación:

- Es un lenguaje de programación que usa **El código fuente es interpretado sin necesidad de crear programas externos** **El tipo de las variables se comprueba durante la ejecución** (el tipo de las variables se comprueba durante la ejecución).
- Es multiplataforma, lo cual es ventajoso para hacer ejecutable su código fuente entre varios sistemas operativos.
- Es un lenguaje de programación que **El código se organiza en clases que se relacionan** **Se basa en utilizar funciones que describen las operaciones a realizar** (se basa en utilizar funciones que describen las operaciones a realizar).
- En Python, el formato del código (p. ej., la indentación) es estructural.



Ventajas



- **Simplificado y rápido:** Este lenguaje simplifica mucho la programación “hace que te adaptes a un modo de lenguaje de programación, Python te propone un patrón”. Es un gran lenguaje para scripting, si usted requiere algo rápido (en el sentido de la ejecución del lenguaje), con unas cuantas líneas ya está resuelto.
- **Elegante y flexible:** El lenguaje le da muchas herramientas, si usted quiere listas de varios tipos de datos, no hace falta que declares cada tipo de datos. Es un lenguaje tan flexible usted no se preocupa tanto por los detalles.
- **Programación sana y productiva:** Programar en Python se convierte en un estilo muy sano de programar: es sencillo de aprender, direccionado a las reglas perfectas, le hace como dependiente de mejorar, cumplir las reglas, el uso de las líneas, de variables”. además, es un lenguaje que fue hecho con productividad en mente, es decir, Python le hace ser más productivo, le permite entregar en los tiempos que me requieren.



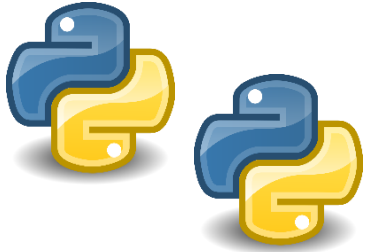
- **Ordenado y limpio:** El orden que mantiene Python, es de lo que más les gusta a sus usuarios, es muy legible, cualquier otro programador lo puede leer y trabajar sobre el programa escrito en Python. Los módulos están bien organizados, a diferencia de otros lenguajes.
- **Portable:** Es un lenguaje muy portable (ya sea en Mac, Linux o Windows) en comparación con otros lenguajes. La filosofía de baterías incluidas, son las librerías que más usted necesita al día a día de programación, ya están dentro del interprete, no tiene la necesidad de instalarlas adicionalmente con en otros lenguajes.
- **Comunidad:** Algo muy importante para el desarrollo de un lenguaje es la comunidad, la misma comunidad de Python cuida el lenguaje y casi todas las actualizaciones se hacen de manera democrática.



Desventajas



- **Curva de aprendizaje:** La “curva de aprendizaje cuando ya estás en la parte web no es tan sencilla”.
- **Hosting:** La mayoría de los servidores no tienen soporte a Python, y si lo soportan, la configuración es un poco difícil.
- **Librerías incluidas:** Algunas librerías que trae por defecto no son del gusto de amplio de la comunidad, y optan a usar librerías de terceros.



Definición de una variable



Las variables en Python se crean cuando se definen por primera vez, es decir, cuando se les asigna un valor por primera vez. Para asignar un valor a una variable se utiliza el operador de igualdad (=). A la izquierda de la igualdad se escribe el nombre de la variable y a la derecha el valor que se quiere dar a la variable.

En el ejemplo siguiente se almacena el número decimal 2.5 en una variable de nombre `x` (como se comenta en el apartado anterior, realmente habría que decir que se crea la etiqueta `x` para hacer referencia al objeto número decimal 2.5). Fíjese en que los números decimales se escriben con punto (.) y no con coma (,).



```
>>> x = 2.5
```

La variable se escribe siempre a la izquierda de la igualdad. Si se escribe al revés, Python genera un mensaje de error:

```
>>> 2.5 = x
SyntaxError: can't assign to literal
```

Para que IDLE muestre el valor de una variable, basta con escribir su nombre:

```
>>> x = 2.5
>>> x
2.5
```

Si una variable no se ha definido previamente, escribir su nombre genera un mensaje de error:

```
>>> x = -10
>>> y
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    y
NameError: name 'y' is not defined
```



Una variable puede almacenar números, texto o estructuras más complicadas (que se verán más adelante). Si se va a almacenar texto, el texto debe escribirse entre comillas simples (') o dobles ("), que son equivalentes. A las variables que almacenan texto se les suele llamar cadenas (de texto).

```
>>> nombre = "Pepito Conejo"
>>> nombre
'Pepito Conejo'
```

Si no se escriben comillas, Python supone que estamos haciendo referencia a otra variable (que, si no está definida, genera un mensaje de error):

```
>>> nombre = Pepe
Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    nombre = Pepe
NameError: name 'Pepe' is not defined
```

```
>>> nombre = Pepito Conejo
SyntaxError: invalid syntax
```

Borrar una variable

La instrucción `del` borra completamente una variable.

```
>>> nombre = "Pepito Conejo"
>>> nombre
'Pepito Conejo'
>>> del nombre
>>> nombre
Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    nombre
NameError: name 'nombre' is not defined
```



Tipos de variables



En el apartado anterior hay definiciones de algunos de los tipos de variables que hay en Python: números decimales, números enteros y cadenas (una o más letras).

Aunque se definan de forma similar, para Python no es lo mismo un número entero, un número decimal o una cadena ya que, por ejemplo, dos números se pueden multiplicar, pero dos cadenas no (curiosamente, una cadena sí que se puede multiplicar por un número).



Por tanto, estas tres definiciones de variables no son equivalentes:

```
>>> fecha = 1997
>>> fecha = 1997.0
>>> fecha = "1997"
>>> fecha = [27, "octubre", 1997]
```

En el primer caso la variable *fecha* está almacenando un número entero, en el segundo *fecha* está almacenando un número decimal y en el tercero *fecha* está almacenando una cadena de cuatro letras. En el cuarto, *fecha* está almacenando una lista (un tipo de variable que puede contener varios elementos ordenados).

Este ejemplo demuestra también que se puede volver a definir una variable. Python modifica el tipo de la variable automáticamente.