

ACTIVIDAD 1: PYTHON 3: Fundamentos de Python

1) ¿Qué es Python?

R/



Es un lenguaje de programación, con grandes similitudes con el lenguaje Perl (en principio se cogió de referencia el lenguaje ABC, que se utiliza en las escuelas). Gracias a su sintaxis sencilla es un muy buen lenguaje para aprender a programar. Python utiliza módulos y paquetes lo cual fomenta la modularidad y la reutilización de código.

2) Historia de Python (creador, año, versiones)

R/ El creador del lenguaje de programación Python es Guido van Rossum, al final de la década de los 80 (en el año 91 se publicó la primera versión).

¿Para que fue creado?

Cuando se programa en entornos Unix, se utiliza sobre todo el lenguaje C, y este sistema no es el más fácil para desarrollar del mundo, así que lo inventó para que esas tareas de programación en éstos sencillas fueran mucho más sencillos.

Versiones:

a. **La versión 1.0:** que se publicó en enero de 1994:

Python es un lenguaje de programación que Van Rossum empezó a desarrollar mientras trabajaba en CWI. Fue este centro de investigación quien liberó, en 1995 la versión 1.2 de Python. A partir de este momento, ya desvinculado de CWI, Van Rossum hizo aún más accesible el código y para el año 2000, el equipo principal de desarrolladores de Python se cambió a BeOpen.com para formar el equipo de BeOpen Python Labs.

b. La versión 1.6 de Python tuvo algunos problemas con su tipo de licencia hasta que la Free Software Foundation (FSF) consiguió cambiar Python a una licencia de Software Libre, que lo haría compatible con GPL.

c. **versión 2.0:** En octubre del año 2000 se publica la segunda versión de Python. Una nueva versión en la que se incluyó la generación de listas,

una de las características más importantes del lenguaje este lenguaje de programación.

- d. En 2001, se crea la Python Software Foundation, la cual a partir de Python 2.1 es dueña de todo el código, documentación y especificaciones del lenguaje.
- e. **versión 3.0:** La última gran actualización de la historia de Python se produjo en el año 2008 con el lanzamiento de la versión 3.0, que venía a solucionar los principales fallos en el diseño de este lenguaje de programación.

3) Características de Python.

- **Lenguaje de propósito general:**
Eso significa que no está orientado a un fin concreto, como puede ser PHP, pensado sobre todo para hacer páginas de internet. Con Python podrás crear páginas sin tener un alto conocimiento (con JavaScript como un poderoso aliado), pero también hacer scripts o software para el sistema operativo Windows.
- **Es multiparadigma:** Pues, aunque su fuerte sea la programación orientada a objetos (es un lenguaje de alto nivel), existen otros paradigmas o estilos de programación para sus usuarios, como es la programación imperativa (con sentencias de bucle) o la programación funcional (con módulos y funciones). Así que, si no sabes nada de POO y sólo sabes escribir código mediante métodos, puedes usar Python perfectamente, cosa que en otros lenguajes hacer eso es imposible.
- **Python es un lenguaje interpretado:**
Cuando programamos en Python, no compilamos el código fuente a código máquina, sino que hay un intérprete que es el que ejecutará el programa basándose en el código directamente.
Aunque esta propiedad hace pensar que los programas puedan ser más lentos, que en lenguaje Python no suele ser así, eso facilita el desarrollo para la siguiente característica.
- **Es multiplataforma:** Al contrario que muchos lenguajes como visual basic, que principalmente solo puedes hacer cosas para Windows, con Python tienes la posibilidad de usarlo en muchos dispositivos y sistemas operativos, ya que se han creado intérpretes para Unix, Linux, Windows y sistemas Mac Os.
- **Es de tipado dinámico:** Cuando declaramos una variable, no es necesario decirle de que tipos son los datos (si es int, string, float, etc.). La variable se adapta a lo que escribimos cuando se ejecuta el programa.
Antes esta característica siempre ha sido criticada en otros lenguajes, por la optimización de la memoria, errores a la hora de escribir código, etc. pero con Python el objetivo es que el lenguaje ayude a la creación de software, no tener que lidiar con peculiaridades propias del lenguaje.

- **Es orientado a objetos:**

Ya hemos dicho que podemos aplicar otro estilo de programación, hacer software orientado a objetos conlleva una serie de ventajas estándar, sobre todo a la hora de reutilizar los componentes gracias a la herencia y sus funciones de polimorfismo.

4) Ventajas y Desventajas de Python.

VENTAJAS:

1. Lenguaje de alto nivel:

Python es un lenguaje de alto nivel, por lo que es más fácil de usar que los de bajo nivel, puesto que estos últimos no tienen mucha abstracción de lenguaje de máquina. Con Python se pueden usar elementos del lenguaje natural, ya que tiene una sintaxis similar al inglés, por lo que es fácil de leer, escribir y aprender.

2. Polivalente y de paradigmas:

Una de las grandes ventajas de Python. Python, al ser un lenguaje de propósito general, se puede usar para diversos propósitos. Es una gran opción para el desarrollo de software, ya que permite a los desarrolladores utilizar grandes frameworks como Django y Flask.

3. Bibliotecas y frameworks:

Dentro de las ventajas y desventajas de Python su mayor beneficio es que tiene una amplia colección de bibliotecas y frameworks.

La biblioteca estándar de Python es muy extensa, puesto que contiene muchos módulos integrados. Además, los usuarios de Python también pueden encontrar bibliotecas adicionales disponibles en PyPI (índice de paquetes de Python).

4. Portabilidad

Python es compatible con todos los sistemas operativos (macOS, Linux, UNIX y Windows), y los programadores solo necesitan escribir código una vez y luego podrá ejecutarse en todas partes.

5. Gratis y de código abierto:

Python es un lenguaje de programación desarrollado bajo la licencia de código abierto aprobada por OSI, todos pueden usarlo y distribuirlo libremente.

6. Baja curva de aprendizaje:

La sencillez de la sintaxis de Python permite escribir programas totalmente funcionales en pocas líneas de código, por lo que su curva de aprendizaje

es muy baja. En pocos días de aprendizaje con Python, cualquier persona puede programar programas sencillos.

7. Comunidad fuerte:

El hecho de que Python sea gratuito y de código abierto contribuye a crear una comunidad sólida. Los programadores de Python pueden descargar el código fuente, modificarlo y distribuirlo como deseen.

DESVENTAJAS:

1. Lentitud:

La lentitud de Python se debe principalmente a su naturaleza dinámica y versatilidad. No obstante, hay formas de optimizar las aplicaciones de Python aprovechando la sincronización, entendiendo las herramientas de creación de perfiles y considerando el uso de múltiples intérpretes.

2. Consumo de memoria

Otra de las desventajas de Python es que, en el caso de que una tarea requiera mucha memoria, Python no es la mejor opción. El consumo de memoria de Python es muy alto, y esto se debe a la flexibilidad de los tipos de datos.

3. Desarrollo móvil

Python es ideal para plataformas de escritorio y servidor, pero para el desarrollo móvil no es un lenguaje muy adecuado. Por este motivo, apenas vemos aplicaciones móviles desarrolladas con Python.

5) Como se definen las variables y los tipos de variables.

R/

Python tiene cuatro tipos primitivos: enteros, flotantes, booleanos y cadenas o Strings.

En la operación de asignación se ven involucradas tres partes:

El operador de asignación =

Un identificador o nombre de variable, a la izquierda del operador

Un literal, una expresión, una llamada a una función o una combinación de todos ellos a la derecha del operador de asignación

Después escribiremos el nombre de la variable seguido de igual y su valor.
Ejemplo:

```
edad = 33
```

Si queremos ver su valor, podemos utilizar la función para imprimir su valor por consola:

```
print(edad)
```

También podemos ver el tipo de dato que guarda (entero o integer, cadena de texto o string, etc...) con la función type:

```
print(type(edad))
```

Que nos dará como resultado:

```
<class 'int'>
```

Para guardar una cadena de texto (string) será similar a guardar un número, pero en este caso tendremos que guardarlo entre comillas, podemos usar comillas simples o comillas dobles.

```
nombre_1 = 'Alber'
```

```
nombre_2 = "Alberto"
```

En ambos casos funcionará exactamente igual.

También podemos interactuar entre ellas, por ejemplo podemos crear dos variables de tipo entero y sumarlas.

```
num_1 = 5
```

```
num_2 = 3
```

```
resultado = num_1 + num_2
```