

PYTHON

Sintaxis Basica

Elaborado por: Inst. Leticia Mendieta

Qué es Python ?

Python....

...es un lenguaje de programación de propósito general, orientado a objetos, que también puede utilizarse para el desarrollo web.

...es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web.

Características del Lenguaje

- Propósito general

Se pueden crear todo tipo de programas. No es un lenguaje creado específicamente para la web, aunque entre sus posibilidades sí se encuentra el desarrollo de páginas.

- Multiplataforma

Hay versiones disponibles de Python en muchos sistemas informáticos distintos. Originalmente se desarrolló para Unix, aunque cualquier sistema es compatible con el lenguaje siempre y cuando exista un intérprete programado para él.

- Interpretado

Quiere decir que no se debe compilar el código antes de su ejecución. En realidad sí que se realiza una compilación, pero esta se realiza de manera transparente para el programador. En ciertos casos, cuando se ejecuta por primera vez un código, se producen unos bytecodes que se guardan en el sistema y que sirven para acelerar la compilación implícita que realiza el intérprete cada vez que se ejecuta el mismo código.

Características del Lenguaje

- Interactivo

Python dispone de un intérprete por línea de comandos en el que se pueden introducir sentencias. Cada sentencia se ejecuta y produce un resultado visible, que puede ayudarnos a entender mejor el lenguaje y probar los resultados de la ejecución de porciones de código rápidamente.

- Orientado a Objetos

La programación orientada a objetos está soportada en Python y ofrece en muchos casos una manera sencilla de crear programas con componentes reutilizables.

- Funciones y librerías

Dispone de muchas funciones incorporadas en el propio lenguaje, para el tratamiento de strings, números, archivos, etc. Además, existen muchas librerías que podemos importar en los programas para tratar temas específicos como la programación de ventanas o sistemas en red o cosas tan interesantes como crear archivos comprimidos en .zip.

Características del Lenguaje

- Sintaxis clara

Por último, destacar que Python tiene una sintaxis muy visual, gracias a una notación indentada (con márgenes) de obligado cumplimiento. En muchos lenguajes, para separar porciones de código, se utilizan elementos como las llaves o las palabras clave begin y end. Para separar las porciones de código en Python se debe tabular hacia dentro, colocando un margen al código que iría dentro de una función o un bucle. Esto ayuda a que todos los programadores adopten unas mismas notaciones y que los programas de cualquier persona tengan un aspecto muy similar.

¿Qué es una instrucción en python?

- Una instrucción es un conjunto de datos insertados en una secuencia estructurada para ordenar al intérprete realizar una operación determinada.

Tipos de instrucciones

Existen dos tipos de instrucciones en python:

- Instrucciones simples

Son aquellas órdenes que python comprende respetando su sintaxis que comienzan y culminan en una sola línea.

- Instrucciones compuestas

Son aquellas que python comprende respetando su sintaxis comenzando con una cláusula de sentencia compuesta y terminando con «:» continúan debajo con una indentación conformando así un bloque de código.

Ejemplo de instrucciones simples y compuestas:

```
Ejemplo de instrucciones simples y compuestas Python
1 #Instrucciones simples Ejemplo:
2 b = 12
3 c = (2 + 2 + 8)
4 a = 'Hola'
5 print (a)
6
7 #Instrucciones compuestas Ejemplo:
8
9 if b == c:
10     print (a)
```


Sintaxis- Indentación

- En otros lenguajes de programación la sangría en el código es solo para facilitar la lectura, en Python la sangría es muy importante.
- Python utiliza la sangría para indicar un bloque de código.
- Las funciones de Python no tienen *begin* o *end* explícitos, ni llaves que marquen dónde empieza o termina su código. El único delimitador son dos puntos (:) y la sangría del propio código. Esto quiere decir que el espacio en blanco es significativo y debe ser consistente.
- En este ejemplo el código de la función (incluida la cadena de documentación) **está con sangría a cuatro espacios**. No tienen por qué ser cuatro, el único requisito es que sea consistente. La primera línea que no esté con sangría queda ya fuera de la función.

El siguiente ejemplo muestra un ejemplo de sangría de código con sentencias if.

```
prueba_01.py x
prueba_01.py ▸ ...
1  a = 33
2  b = 200
3
4  if b > a:
5      print("b es mayor que a")
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL

```
PS C:\Users\pc\Desktop\PYTHON> python prueba_01.py
b es mayor que a
PS C:\Users\pc\Desktop\PYTHON> |
```

```
prueba_01.py x
prueba_01.py ▸ ...
1  a = 33
2  b = 200
3
4  #"sin indentacion"
5  if b > a:
6  print("b es mayor que a")
```

PROBLEMAS 1 SALIDA CONSOLA DE DEPURACIÓN TERMINAL

```
PS C:\Users\pc\Desktop\PYTHON> python prueba_01.py
File "prueba_01.py", line 6
    print("b es mayor que a")
    ^
IndentationError: expected an indented block
PS C:\Users\pc\Desktop\PYTHON> |
```

Python-Variables

Las variables son contenedores para almacenar valores de datos.

A diferencia de otros lenguajes de programación, Python no tiene comandos para declarar una variable.

Una variable se crea en el momento en que primero le asigna un valor.

Ejemplo:

Asignar valor a múltiples variables

```
prueba_01.py x
prueba_01.py ▾ ...
1 x = 5
2 y = "Juan"
3 print(x)
4 print(y)
5

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL

PS C:\Users\pc\Desktop\PYTHON> python prueba_01.py
5
Juan
PS C:\Users\pc\Desktop\PYTHON> 
```

```
prueba_01.py x
prueba_01.py ▾ ...
1 x, y, z = "Juan", "Pedro", "Maria"
2
3 print(x)
4 print(y)
5 print(z)

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL

PS C:\Users\pc\Desktop\PYTHON> python prueba_01.py
Juan
Pedro
Maria
PS C:\Users\pc\Desktop\PYTHON> 
```

Python- Números

Hay tres tipos numéricos en Python:

- Int
- float
- Complejo

Las variables de tipos numéricos se crean cuando se les asigna un valor:

```
x = 1    # int  
y = 2.8  # float  
z = 1j   # complex
```


Conversión de un tipo a otro

Se puede convertir de un tipo a otro con el `int()`, `float()` y `complex()` con los siguientes métodos:

```
x = 1 # int
```

```
y = 2.8 # float
```

```
z = 1j # complex
```

```
#convert from int to float:
```

```
a = float(x)
```

```
#convert from float to int:
```

```
b = int(y)
```

```
#convert from int to complex:
```

```
c = complex(x)
```

Números Aleatorios

Python no tiene una función `random()` para generar un número aleatorio, pero Python tiene un módulo incorporado llamado `random` que se puede usar para generar números aleatorios:

```
import random
```

```
print(random.randrange(1,10))
```

```
#se genera un numero entre 1 y 10
```


Python- String

Los literales de cadena en python están rodeados por comillas simples o comillas dobles.

- 'hola' es lo mismo que "hola" .

Asignar cadena a una variable: La asignación de una cadena a una variable se realiza con el nombre de la variable seguido de un signo igual y la cadena:

```
a = "Hola"  
print(a)
```

Cadena Multilineas: Se debe utilizar tres comillas dobles o tres comillas simples.

```
a = """Lorem ipsum dolor sit amet,  
consectetur adipiscing elit,  
sed do eiusmod tempor incididunt  
ut labore et dolore magna aliqua."""  
print(a)
```

Python- String métodos

El método `strip()` elimina cualquier espacio en blanco desde el principio o el final:

```
a = " Hola, Mundo! "  
print(a.strip()) # retorna "Hola, Mundo!"
```

El método `len()` devuelve la longitud de una cadena:

```
a = "Hola, Mundo!"  
print(len(a)) # retorna 13
```

El método `lower()` devuelve la cadena en minúsculas:

```
a = "Hola, Mundo!"  
print(a.lower()) # retorna "hola, mundo!"
```


Python- String métodos

El método `upper()` devuelve la cadena en mayúsculas:

```
a = "Hola, Mundo!"  
print(a.upper()) # retorna "HOLA, MUNDO!"
```

El `format()` es un método para insertar números en cadenas:

```
edad = 31  
txt = "Mi nombre es Juan, y tengo {}"  
print(txt.format(edad)) # retorna "Mi nombre es Juan, y tengo 31"
```

La función `str()` convierte el valor especificado en una cadena.

```
x = str(3.8) #devuelve 3.8 en formato string
```

Python- Operadores aritmeticos

+	Adición	$x + y$
-	Sustracción	$x - y$
*	Multiplicación	$x * y$
/	División	x / y
%	Modulo	$x \% y$
**	Exponenciación	$x ** y$
//	Floor división// redondea el resultado al número entero más cercano	$x // y$

prueba_01.py x

prueba_01.py ▸ ...

```
1 x= 9
2 y= 3
3 print("Suma", x + y)
4 print("Resta", x - y)
5 print("Multiplicacion", x * y)
6 print("Division", x / y)
7 print("Modulo", x % y)
8 print("Exponenciacion", x **y) #igual a 9*9*9
9
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL

1: powershell ▼

PS C:\Users\pc\Desktop\PYTHON> python prueba_01.py

Suma 12

Resta 6

Multiplicacion 27

Division 3.0

Modulo 0

Exponenciacion 729

PS C:\Users\pc\Desktop\PYTHON>

Python- Ingresar datos por teclado

Para introducir una cadena:

```
cadena = input("Introduce una cadena de texto:")  
print ("La cadena que ingreso es:\n",cadena)
```

Para introducir un numero int:

```
numero = int(input("Introduce un numero:"))  
print ("El numero que ingreso es:\n",numero)
```

Para introducir un numero float:

```
numero = float(input("Introduce un numero:"))  
print ("El numero que ingreso es:\n",numero)
```

Python- Comentarios

En python los comentarios se pueden poner de dos formas:

- Escribiendo el símbolo # delante de la línea de texto donde está el comentario.
- Escribiendo triple comilla doble («»») al principio y al final del comentario (que puede ocupar más de una línea).

Este es un comentario en python

Este es otro comentario en python

aunque lo hemos escrito en dos lineas

""" Este es un comentario multilinea. La
siguiente es otro comentario """

""" Aunque es multilinea, sólo usamos una """

Fuentes Consultadas

<https://desarrolloweb.com/articulos/1325.php>

<https://pythones.net/sintaxis-basica-del-lenguaje/>

<https://www.w3schools.com/python/default.asp>

<https://linux.com/captura-de-datos-en-python/>