

# Python

## Programación Web2

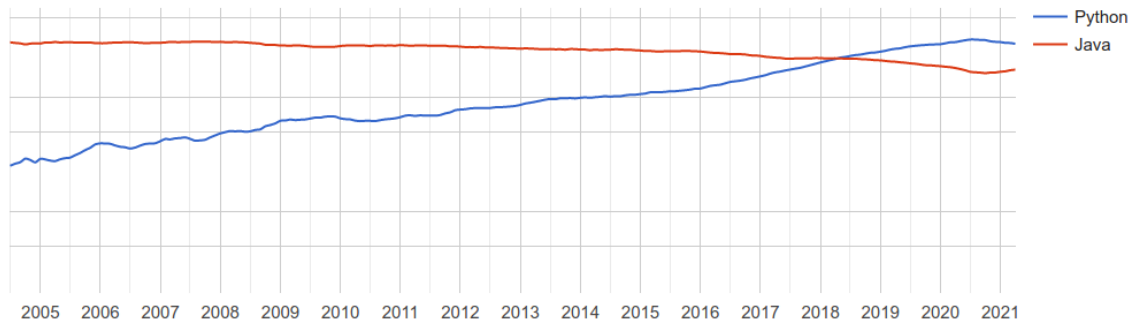
[Basado en el material del w3schools](#)



# El Lenguaje Python

- Creado por Guido van Rossum en 1991
- Es más antiguo que Java!
- Sin embargo su mayor popularidad es más bien reciente

PYPL Popularity of Programming Language



Fuente:

<http://pypl.github.io/PYPL.html>

# Entorno de programación

Python es un lenguaje interpretado, por lo que sus programas se llaman scripts

Python cuenta con un muy útil ambiente interactivo que permite probar el código antes de que este se escriba en archivos

```
[apaz pweb2 $ python3
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 03:13:28)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
[>>> print("Hola Mundo")
Hola Mundo
[>>> ^D
apaz pweb2 $
```

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

```
apaz pweb2 $ vim hello.py
apaz pweb2 $ python hello.py
Hola Mundo
apaz pweb2 $
```

# Sintaxis

- Muy expresivo, quizá demasiado flexible!
- Parecido a la de otros lenguajes como Java o C++ pero un gran toque de simplicidad
- Los errores de indentación ya no serán un problema :)
- La ausencia de paréntesis, llaves y punto y coma, podría causar costumbres problemáticas en otros lenguajes y viceversa
- En python todo es un objeto!
  - Use type para comprar el tipo de “valores primitivos”

# Comentarios

Sólo hay comentarios de una línea:

Si se desea comentarios de más de una línea se puede usar sus habilidades con los comandos del editor

```
#comentario de una línea  
print("Hola Mundo");  
~
```

```
##comentario de una línea  
#print("Hola Mundo");  
~  
~  
~
```

```
~  
~  
:1,$s/^/\#/g
```

# Identificadores de Variables

Al estilo javascript o perl, dinámicamente tipado

Las variables no se tienen que declarar

Las mismas reglas para los nombres que Java

Para usar variables globales en funciones se necesita declararlas con global



# Tipos de datos

Text Type: `str`

Numeric Types: `int`, `float`, `complex`

Sequence Types: `list`, `tuple`, `range`

Mapping Type: `dict`

Set Types: `set`, `frozenset`

Boolean Type: `bool`

Binary Types: `bytes`, `bytearray`, `memoryview`

# Operadores

+

-

\*

/

%

\*\*

//

=

=+

=-

=\*

=\

=0<sup>0</sup>

=\\

=\*\*

=&

=|

=^

=<<

=>>

and

or

not

is

is not

in

not in

&

|

^

~

<<

>>

# Contenedores

- Listas []
  - Los elementos tienen posición
  - Pueden haber elementos repetidos
  - Elementos se puede acceder con índices, positivos y negativos
  - Mutables
- Tuplas ()
  - Similar a las listas pero son inmutables
- Conjuntos {}
  - Similar a las listas, pero no hay posiciones y no hay elementos repetidos
- Diccionarios { : }
  - Similar a los diccionarios de JavaScript

# Condicionales

```
if a > b:  
    print("a es el mayor")  
else:  
    print("b es el mayor")
```

```
if a == 1:  
    print("acción 1")  
elif a == 2:  
    print("acción 2")  
elif a == 3:  
    print("acción 3")  
else:  
    print("lo que sobra")
```

# Ciclos

- Tiene break y continue
- While incluso tiene ELSE
- For, sólo funciona como un foreach
  - La función range() para iterar sobre índices
  - pass para ciclos vacíos

# Funciones

- Se declaran con la palabra reservada
  - `def`
- Si la cantidad de parámetros es desconocida
  - `*arg`
  - Los argumentos se reciben en la lista `arg[0]`, `arg[1]`, etc.
- Si se desea parámetros con nombres propios de cantidad desconocida
  - `**arg`
  - Se puede enviar cualquier nombre, se reciben como un diccionario.
- Funciones lambda similares a las de JavaScript y Java
  - **lambda** arguments : expression

# Modulos

- Como los import de java, pero se usa el nombre del archivo
- Para usar la funciones dentro del módulo se debe usar la notación con punto, como si de un objeto se tratara, igual con las variables
- Es posible renombrarlos para ahorrar escritura
  - **import** modulo **as** m
- Se puede evitar la notación con punto se puede importar una función específica
  - **from** modulo **import** funcion