



CURSO BÁSICO DE PYTHON

ING. MATIAS DIEZ

python™

CURSO DE PROGRAMACIÓN

Fechas del curso

- Primer clase: 05/09/2020
- Segunda clase: 11/09/2020
- Tercera clase: 18/09/2020
- Cuarta clase: 25/09/2020



CURSO DE PROGRAMACIÓN

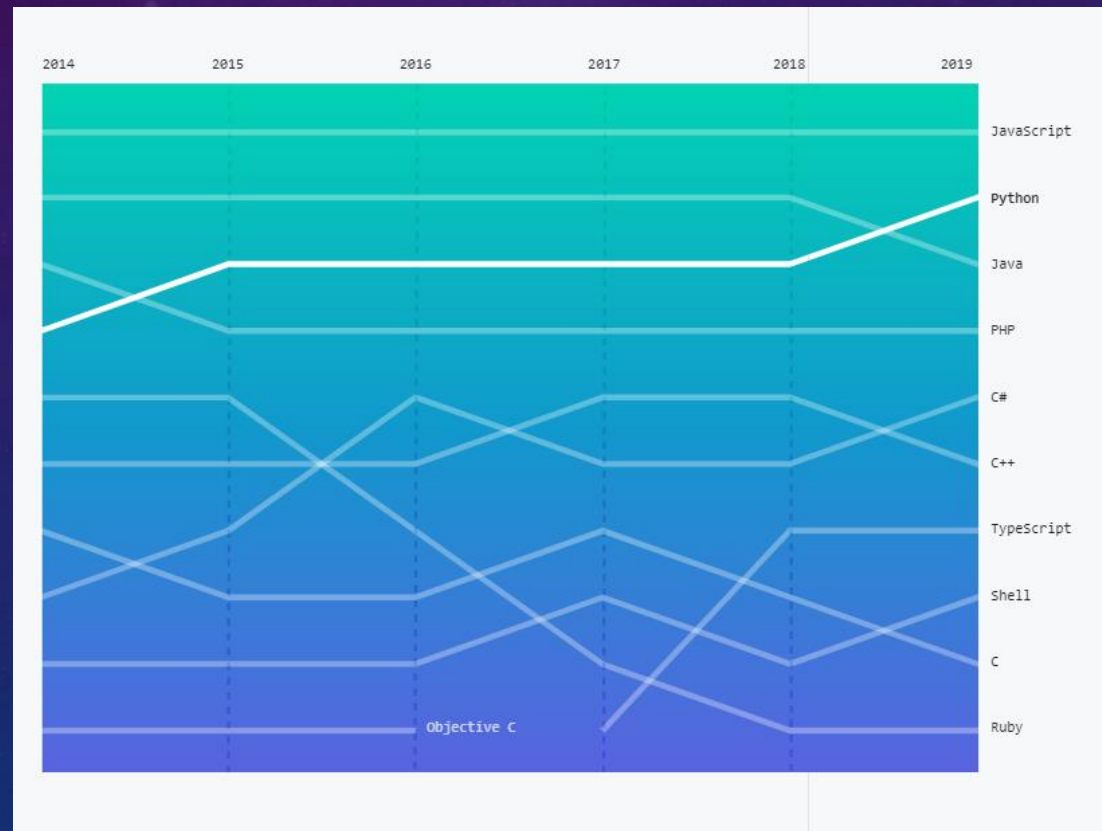
Temario del Curso

- Primer clase:
 - > Introducción a herramienta “VS Code”
 - > Variables y datos alfanuméricos
 - Segunda clase:
 - > Variables Numéricas y librerías matemáticas.
 - > Expresiones lógicas y condicionales
 - Tercera clase:
 - > Funciones, listas y Diccionarios, tuplas
 - Cuarta clase:
 - > Backend API REST usando FLASK c/POSTMAN
- Nota: si no das el tiempo LAMBDA y API GATEWAY en



CURSO DE PROGRAMACIÓN

¿Por qué Python? Es el tercer lenguaje más empleado en el mundo de acuerdo al reporte mensual 09/20 de GITHUB.COM (<https://octoverse.github.com/>)

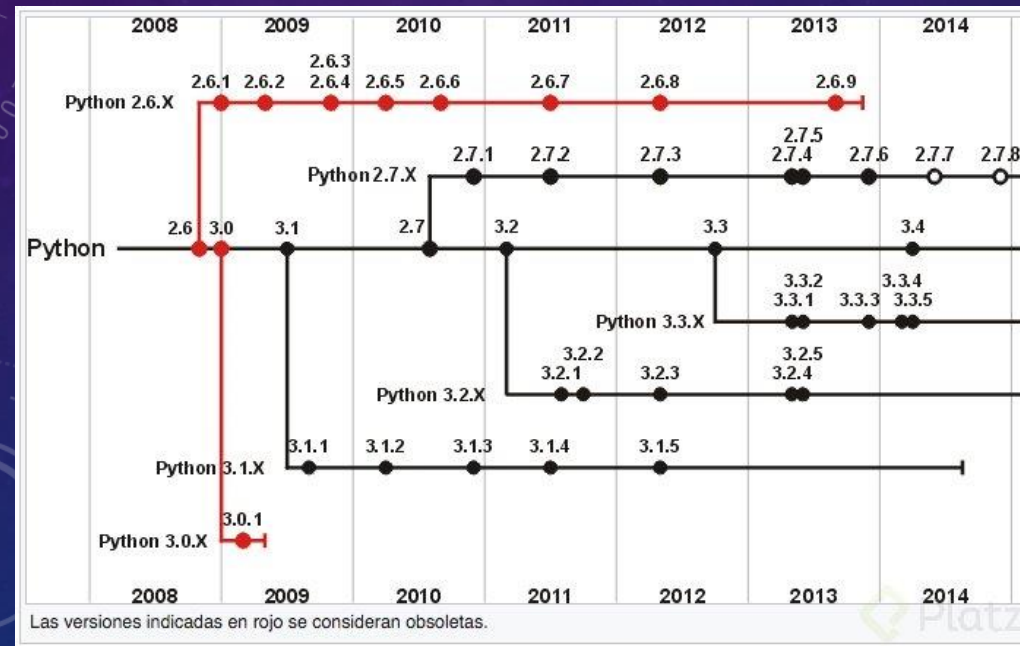


python™

CURSO DE PROGRAMACIÓN

Historia Rápida

A principios de los 90 de la mano del matemático holandés Guido Van Rossum a partir un proyecto de investigación. El nombre del lenguaje proviene debido al nombre de un programa británico llamado [Monty Python](#).



- Fuente: [Wikipedia](#)
- Nota: el “baseline” rojo expresa la obsolescencia.



python™

CURSO DE PROGRAMACIÓN

Arquitectura del lenguaje

El Python es un lenguaje interpretado. Esto implica que cada línea de código que desarrollamos es leído y procesado por un motor de ejecución.

El resultado de este procesamiento es un nuevo formato llamado *bytecode*. Como otros lenguajes (java, NET) es luego transformado byte para que sea interpretado por sistema operativo.

Esta característica permite que este programa sea portable a cualquier plataforma. En los últimos años la arquitectura serverless tiene como principio que el programador solo ponga su foco en la necesidad y los recursos de infraestructura son transparentes. Ejemplo de esto son los AWS Lambda o App Runtime de Google Cloud.



python™

CURSO DE PROGRAMACIÓN

Versiones más populares

Las dos versiones más populares de Python son la 2.7 o superior y 3.x.

La primera todavía se las encuentra en la mayoría de las plataformas unix o Windows. Aparece como estándar de facto.

Veamos un ejemplo desde un terminal en Linux NOOS en ARM Raspeberry PI.

Les sugerimos ejecutar la instrucción *print* para ver diferencias entre ambas versiones.

Nota: pueden encontrar documentación sobre este tema [aquí](#).



python™

CURSO DE PROGRAMACIÓN

Instalación de Python y VS

También existen la posibilidad de ejecutarlo en la nube de Microsoft Azure usando el componente de [Codespace](#). No es el objetivo de este curso , pero vale la pena mencionarlo (por 12 meses es gratis, luego se paga 12 dólares por mes).

Otra opción para trabajar es [PyCharm de JetBrains](#). Existen una versión comunidad que es gratuita y luego esta la versión comercial EE.

También existen las Jupyter Netbooks que se pueden utilizar dentro de la nube de forma gratuita en los sitios que forman parte de los “CODA” o “Colaboratory” (son gratuitos y emplean un instancia de computo brindada por Google Cloud.



python™

CURSO DE PROGRAMACIÓN

Instalación de Python y VS

HAND'S ON (DEMONSTRACION)



python™

CURSO DE PROGRAMACIÓN

Palabras Reservadas de Python

Son 33 palabras para versión que estamos usando al (07/06/2020) , todas son sensitivas. A lo largo de las clases iremos viendo en su mayoría.

False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield

Notas: son las que el interprete de Python reconoce naturalmente, el resto son módulos y sus respectivas funciones.



python™

CURSO DE PROGRAMACIÓN

Funciones básicas para poder trabajar

Dado, que vamos a estar trabajando y agregando contenidos clase a vamos a ver *las tres operaciones básicas para que de manera interactiva y/o programática podamos realizar las tres operaciones básicas de cualquier algoritmo.*

1) *Entrada de datos: `input = ("Mensaje")`*

ejemplo = `dato1 = input ("Ingrese Número Entero")`

2) *Procesamiento utilizando cualquiera de los operadores (que vamos a ver más adelante).*

`int valor = dato1 + 20`

3) *Output de datos: `print ("expresión a evaluar")`*

`print (valor)`



python™

CURSO DE PROGRAMACIÓN

Bloques

Python no soporta como otros lenguajes llaves, paréntesis, etc.

Los bloques se utilizan por “indentación” de líneas.

Vamos un ejemplo simple:

```
bandera = True  
if bandera == True:  
...     print("verdadero")  
... else:  
...     print("falso")  
...  
verdadero
```

Los comentarios se ponen en el código con el literal “#”



python™

CURSO DE PROGRAMACIÓN

Comentarios

Los comentarios se ponen en el código con el literal "#"

Ejemplo

Comentario 1

Se pueden intercalar

bandera = true # declare una variable tipo "bool"

Se pueden poner multiples lines usando el operador

'''

Este es un comentario multilinea

'''



python™

CURSO DE PROGRAMACIÓN

Variables

Las variables no necesitan ser declaradas , el lenguaje tiene la capacidad de ir creándolas dinámicamente:

Ejemplo:

```
edad = 46  
sueldo = 10.50  
nombre= "Fulanito"
```

Usando la function `type(var)`, puedo saber de que tipo es:

```
type(edad)
```



python™

CURSO DE PROGRAMACIÓN

Operadores Matematicos

Los operadores que ofrece el lenguaje son los siguientes:

- *suma = "+"*
- *resta = "-"*
- *multiplicación = "*"*
- *división = /*
- *cociente de división = //*
- *resto de una división = %*



python™

CURSO DE PROGRAMACIÓN

Operadores Relacionales

- Mayor ">"
- Menor "<"
- Igual a "=="
- Distinto a "!="
- Mayor o igual a ">="
- Menor igual a "<="

```
Python 3.7.3 (default, Dec 20 2019, 18:57:59)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> #Datos dos valores
... a = 13
>>> b = 33
>>> print(a > b)
False
>>> print(a < b)
True
>>> print(a == b)
False
>>> print(a != b)
True
>>> print(a >= b)
False
>>> print(a <= b)
True
>>> █
```



python™

```
from os import system
```

```
system("clear")
```

CURSO DE PROGRAMACION

Operadores Lógicos

- *AND "and"*
- *OR "or"*
- *NOT (not)*

```
>>> #Dadas dos variables tipo bool
... a = True
>>> b = False
>>> print(a and b)
False
>>> print(a or b)
True
>>> print(not a)
False
>>>
...
>>> █
```

Nota: para limpiar la consola del interprete ejecutar:

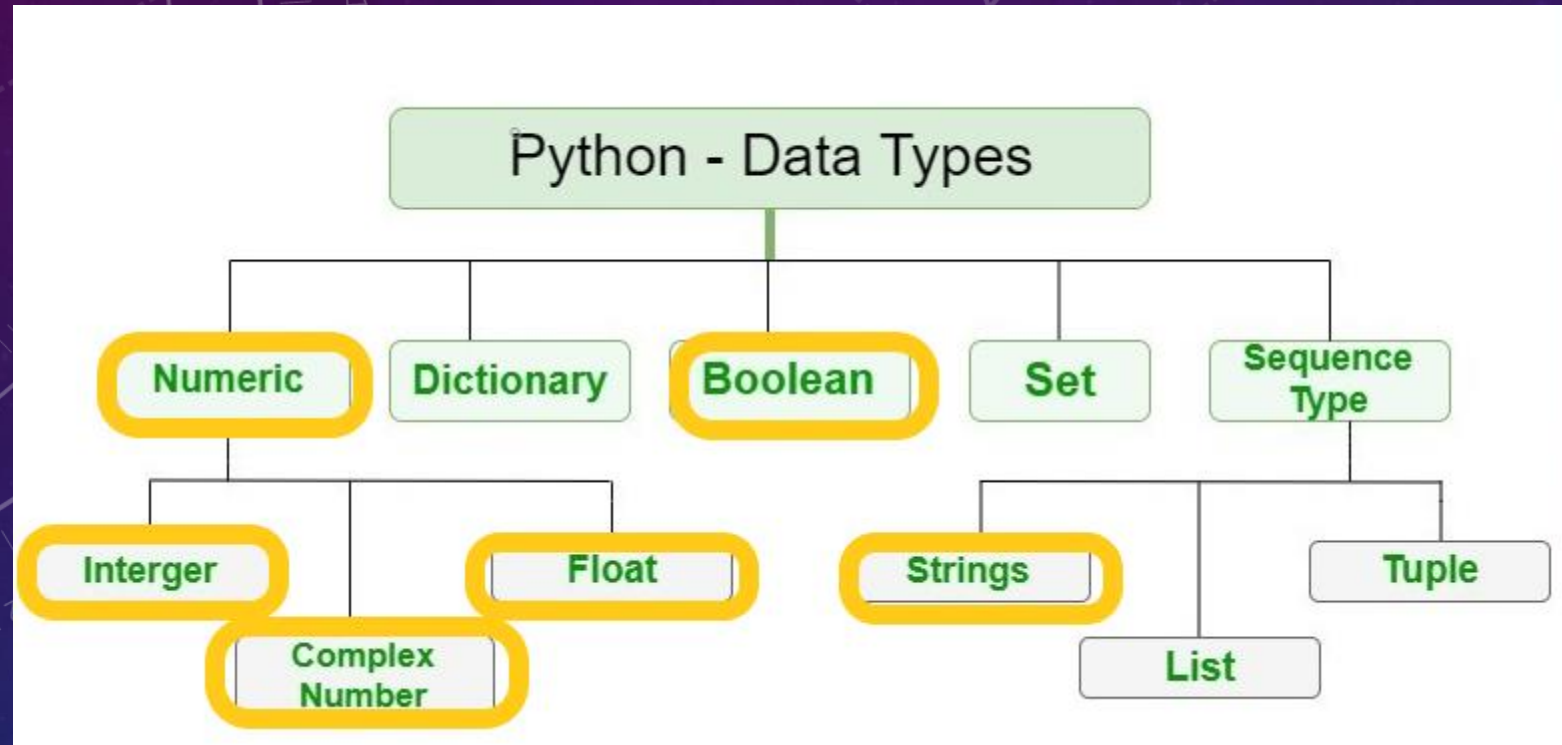
```
from os import system # se ejecuta una sola vez por el modulo ya fue importado
system ("clear") # si fuera Windows "cls"
```



python™

CURSO DE PROGRAMACIÓN

Tipos de Datos



Nota: los que están marcados color amarillo son los que utilizaremos de hoy en adelante.



python™

CURSO DE PROGRAMACIÓN

Tipos de Datos: Jerarquía “Numeric”

Entero

a = 7

Flotante

b = 10.50

complejo

c = a + 3j

Ejemplo:

El valor del dato a es 7

El tipo de dato a es <class 'int'>

El valor del dato b es 10.5

El tipo de dato b es <class 'float'>

El valor del dato c es (7+3j)

El tipo de la variable c es <class 'complex'>



python™

CURSO DE PROGRAMACIÓN

Tipos de Datos: Jerarquía "String"

```
# Declaro un string con comilla simple
salida = "Hola"
print ("Salida",salida)
salida_cc = 'Hola'
print ("Salida",salida_cc)
salida = 'Salida ahora incluye "comillas dobles"'
print ("Salida",salida)
salida = "Salida ahora incluye 'comillas simples'"
print ("Salida",salida)
# usamos un caracter de escape
salida = "Salida ahora incluye \"comillas dobles\""
print ("Salida",salida)
# usamos un TAB o tabulador
salida = "Salida ahora \t\tab"
print ("Salida",salida)
```

```
# usamos un salto de carro
salida = "Salida ahora \nsalto de carro"
print (">>>",salida)
# usamos un operador datos en crudo
salida = r">>> ahora c:\Windows"
print ("Salida",salida)
# literales largos
salida = """
Linea 1
Linan 2 \t\tab
Linea3
"""
print ("Salida",salida)
```



python™

CURSO DE PROGRAMACIÓN

Tipos de Datos: Jerarquía "String"

```
# conactenacion de cadenas
cadena1 = "Nombre "
cadena2 = "Apellido "
cadena3 = cadena1 + cadena2
print ("Salida>>>", cadena3)
# hago lo mismo en la salida
print ("Salida>>>", cadena1 + cadena2)
#relleno de caracteres
caracter="*" * 10
print (caracter)
```



python™

CURSO DE PROGRAMACIÓN

Final Feliz!



python™