

# Phyton

*Beautiful is better than ugly.*

*Explicit is better than implicit.*

*Simple is better than complex.*

*Complex is better than complicated.*

*Readability counts.*



# ¿Qué es Phyton?

- Es un lenguaje de programación de alto nivel
- Se considera un lenguaje de propósito general
- Es un lenguaje de programación multi-paradigma
- Es un lenguaje interpretado.



# Clasificación de Lenguajes

- **Manera de ejecutarse**

**Compilados.** Código fuente > código objeto > Programa ejecutable

**Interpretados.** Un programa ejecuta las instrucciones de manera directa

# ¿Quién inventó Phyton?

- Guido van Rossum (1956)
- Centrum Wiskunde & Informatik
- Holandés
- BDFL (*Benevolent Dictator For Life*)
- Sucesor de lenguaje ABC



# Python

- Uno de los aspectos más importantes en la filosofía de Python es la legibilidad del código.
- Los bloques de código se delimitan utilizando espacios en blanco o sangrías (indentación) en lugar del uso de corchetes { }
- Lo cual permite a los programadores reducir el número de líneas de código de un programa.

# Versiones

- Versión 2.- 2000 (recolector de basura y administración de memoria).
- Versión 3.- 2008 (Unicode , manejo de excepciones, división de enteros)
- El principal problema de la versión 3 es la no-compatibilidad con la versión 2

# Primeros pasos

- Instalación de Python 3.6
  - Python Puro (bibliotecas estándar)
- Instalación de una distribución o un IDE
  - Anaconda
  - PyPy
  - ActivePython
- Ejecución del intérprete
  - Consola
  - Parte de la distribución que se instale

# Variables

- 2+2
- 3\*5
- cadena="hola"
- cadena\*3
- a = 10
- c = 20
- print(a)

# Identificadores

## Válidos

- hola
- hola12t
- \_hola
- Hola

## Inválidos

- hola abc
- 8hola
- hola~\
- Hola\*9

# Entrada / Salida

- `a = input("ingresa el valor de a")`
- `print(a)`
- `print("hola ",nombre)`

# Ciclos definidos

**for** <variable> **in** <secuencia de valores>:  
    <cuerpo>

for x in range(10)

    x\*x

for x in range (5,10)

    x\*x

for x in [1,2,4,7,10]

    x\*x

# Funciones

```
def funcion():
    print("hola")
    print("...más cosas...")
    print("...aquí termina la función")
#más código
```

# Funciones

```
def funcion2(x):  
    resultado=x*x  
    print(resultado)
```

```
def funcion3(x,y):  
    resultado=x*y  
    print(resultado)
```



# Funciones

```
def funcion2(x):  
    resultado=x*x  
    return resultado  
a = funcion(3)
```

```
def funcion3(x,y):  
    resultado1=x*y  
    resultado2=x+y  
    return resultado1,resultado2  
a,b=funcion3(2,10)
```



# Condicional

```
def otra_funcion():
    x = input("Ingrese un numero: ")
    if x > 0:
        print ("Numero positivo")
    else:
        print ("Numero no positivo")
```

# Mutables y no mutables

- En Python no solo se toma en cuenta el tipo de dato (int, string, etc) también se debe considerar si es mutable o no mutable.
- Los *números*, los *strings* y las *tuplas* son *inmutables*.
- Las *listas*, las *clases* o los *conjuntos* son *mutables*.

# Listas

- Una lista es una secuencia de valores. A diferencia de otros lenguajes de programación, los elementos de las listas pueden ser de cualquier tipo
- Se utilizan listas para poder modelar datos compuestos pero cuya cantidad y valor varían a lo largo del tiempo.

```
lista1=[1,2,3,4]
```

```
lista2=['perro' , 'gato' , 'delfin ']
```

```
lista3=[1 , 5.5 , 'perro' , -10]
```

# Métodos para listas

- lista.append(x)
- lista.extend(L)
- lista.insert(i,x)
- lista.remove(x)
- lista.pop(i)
- lista.sort()
- lista.reverse()

# Tuplas

- Similar a una lista, la diferencia es que la tupla no se va a modificar.
- Se puede acceder a los elementos para consultarlos pero no se pueden eliminar

mitupla=(1,2,3,4)

miTupla=("hola","adiós","pepe")

miTupla=(1,"hola",2.34)

- Las tuplas no tienen métodos

"You can't just copy-pase pseudocode  
into a program and expect it to work"



**He : You are ';' to my code**

**She : Sorry, I Code in Python!!**

# Actividad

- Instalar Python 3
  - \* Sugerencia: Instalar la suite de Anaconda
- Buscar en Google: “Anaconda Python”
- Instalar con la variable de entorno (activar casilla)
- Probar instalación