



Grupo de Desarrollo en Python



- Python favorece un código legible por tener una sintaxis simple, clara y sencilla.
- La gran cantidad de librerías disponibles y la potencia del lenguaje, entre otros, hacen que desarrollar una aplicación en Python sea sencillo, muy rápido y, lo que es más importante, divertido.



Mi Primer Programa en Python

- "Hola Mundo"
- **Print** "Hola Mundo"
- **Print** 'Hola Mundo'



Tipos de Variables

Entero(int): 2, 3, 4, etc.

Cadena(str): "Hola Mundo", 'Hola Python', etc.

Punto Flotante(float): 2.1, 3.5, etc.

Boolean(bool): True o False



Comprobar tipo de variable

type(variable)

- **Print** type(3.4)
 <type 'float'>
- **Print** type('True')
 <type 'str'>
- **Print** type(4)
 <type 'int'>
- **Print** type(True)
 <type 'bool'>



Asignacion de Variables

- Se utiliza un operador de asignación(=)
mensaje = "Hola"

n = 1

pi = 3.1416

booleano = True

***print** variable



Introduccion de datos

- Existe 2 formas:

1. Cualquier tipo de variable **raw_input("")** y entre comillas va el mensaje que se mostrara al usuario:

```
a = raw_input("Ingresar cualquier tipo de variable: ")
```

Ingresar cualquier tipo de variable:



2. Para valores numericos **input(“”)** y entre comillas se coloca el mensaje que se mostrara al usuario:

```
a = input("Ingresar solo una variable numerica:  
")
```

Ingresar solo una variable numerica:



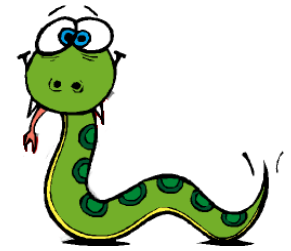
Operadores aritméticos

- Suma $+$ $r = 3 + 2$ # r es 5
- Resta $-$ $r = 4 - 7$ # r es -3
- Negación $-$ $r = -7$ # r es -7
- Multiplicación $*$ $r = 2 * 6$ # r es 12
- Exponente $**$ $r = 2 ** 6$ # r es 64
- División $/$ $r = 3.5 / 2$ # r es 1.75
- División entera $//$ $r = 3.5 // 2$ # r es 1.0
- Residuo $\%$ $r = 7 \% 2$ # r es 1



Operadores de Condición

- `==` y `#` x es igual a y
- `!=` y `#` x no es igual a y
- `>` y `#` x es mayor que y
- `<` y `#` x es menor que y
- `>=` y `#` x es mayor o igual que y
- `<=` y `#` x es menor o igual que y



Operadores Lógicos

- **and** # $x < 4$ y $x > 2$
- **or** # $x < 2$ o $x > 3$
- **not** # no $x < 2$



Sentencias de Decisión

- Sentencia de Decisión Simple
- Sentencia de Decisión Doble
- Sentencia de Decisión Multiple



Sentencia de Decisión Simple

if condicion:

instrucción 1

instrucción n



Sentencia de Decisión Doble

if condicion:

instrucción 1

instrucción n

else:

instrucción 1

instrucción n



Sentencia de Decisión Múltiple

if condición:

instrucción 1

instrucción n

elif condición:

instrucción 1

instrucción n

else:

instrucción 1

instrucción n



Gracias!

Preguntas???

Quejas???

Ejercicios de Aplicación