

01 Variables, Expresiones y Sentencias

Valores y tipos

In []:

```
print(3)
print('Esto es un texto')
print(True)
print(3.2)
```

para conocer el tipo de un valor se utiliza la función `type()`

In []:

```
print('el valor {} es de tipo {}'.format(2, type(2)))
print('el valor {} es de tipo {}'.format(2.0, type(2.0)))
print('el valor \'{}\' es de tipo {}'.format('str', type('str')))
print('el valor {} es de tipo {}'.format(True, type(True)))
print('el valor {} es de tipo {}'.format([1,2], type([1,2])))
print('el valor {} es de tipo {}'.format((1,2), type((1,2))))
print('el valor {} es de tipo {}'.format({'a': 1}, type({'a': 1})))
print('el valor {} es de tipo {}'.format(set([1, 3, 1, 4]), type(set([1, 3, 1, 4]
))))))
```

Variables

In []:

```
texto = 'En un lugar de la mancha'
n = 13
pi = 3.1415926535897931
lista = [1, 2, 3, 5, 8, 13]

print('texto: {}'.format(texto))
print('n: {}'.format(n))
print('pi: {}'.format(pi))
print('lista: {}'.format(lista))
```

In []:

```
type(lista)
```

Los nombres de las variables pueden contener letras, números y guión bajo (`_`)

- Los nombres de las variables no pueden empezar con números.
- No se pueden usar palabras reservadas como nombre de las variables.

In []:

```
lugar_de_nacimiento = 'Bolivia'
numero2 = 2
esVisible = False
```

In []:

```
lvariable = 'una variable'
```

In []:

```
person@ = 'persona'
```

In []:

```
class [= 'esto no puede ser una variable'
```

Palabras reservadas de python <https://docs.python.org/2.5/ref/keywords.html>
(<https://docs.python.org/2.5/ref/keywords.html>)

Tip: Hacer uso de nombres de variables mnemotécnicas para ayudarnos a recordar por qué creamos la variable en primer lugar.

Sentencias

In []:

```
print(1)
n = 2
print(n)
```

Operadores y Operandos

Los operadores son +, -, *, /, **

In []:

```
print('suma 1 + 2 = {}'.format(1 + 2))
print('resta 4 - 2 = {}'.format(4 - 2))
print('multiplicación 3 * 2 = {}'.format(3 * 2))
# En python 2.0 la división te devolvía un entero, python 3.0 te devuelve un typ
o flotante
print('división 10 / 3 = {}'.format(10 / 3))
# Para obtener un entero se usa //
print('división 10 // 3 = {}'.format(10 // 3))
print('potencia 5 ** 3 = {}'.format(5 ** 3))
```

Operador Módulo

In []:

```
print('Módulo 5 % 3 = {}'.format(5 % 3))
```

expresiones

Es una combinación de Valores, Variables y Operadores

```
13
x
x + 13
```

In []:

```
(3 + 2) ** 3
```

Orden de las operaciones

python utiliza **PEMDAS**

1. **Parentheses first:** parentesis
2. **Exponents:** Potencias
3. **Multiplication and Divisio:** Multiplicación y División
4. **Addition and Subtraction:** Suma y Resta

In []:

```
# (10 + 2) + 3 * 4 - ((2 ** 2) * (2 + 3))
# 12 + 12 - 2 ** 2 * 5
# 12 + 12 - 4 * 5
# 12 + 12 - 20
# 4

(10 + 2) + 3 * 4 - 2 ** 2 * (2 + 3)
```

Operaciones en Cadenas

In []:

```
# el operador + concatena cadenas
a = '1024'
b = '512'
a + b
```

In []:

```
# el operador * multiplica el contenido de la cadena
texto = 'texto '
n = 3
texto * n
```

Input

In []:

```
nombre = input('Cómo te llamas?\n> ')\nprint('Hola {}'.format(nombre))
```

In []:

```
edad = input('Qué edad tienes?\n> ')\nhoras_de_vida = int(edad) * 365 * 24\nprint('Usted tiene aproximadamente {} horas de vida.'.format(horas_de_vida))
```

Comentarios

In []:

```
"""\nla formula para obtener la aceleracion es la siguiente:\na = dv / dt = (vf - vi)/(tf - ti)\n"""\n\nv0 = 12.3  # velocidad inicial en m/s\nv1 = 41.2  # velocidad final en m/s\nt0 = 0     # Tiempo inicial en segundos\nt1 = 10    # Tiempo Final en segundos\n\n# Aceleración se representa en m/s2\na = (v1 - v0)/t\n\nprint('la aceleración es {} m/s2'.format(a))
```

Debugging

In []:

```
nombre mal = 1
```

In []:

```
mes = 09
```

In []:

```
var1 = 2\nvar2 = 3 + var
```

In []:

```
numeroUno = 1\nnumeroTres = numerouno + 2
```