

Clase-1

December 17, 2018

1 Características de Python

```
In [4]: dos_mas_dos = 2 + 2
```

```
In [5]: numero_flotante = 3.16
```

```
In [6]: cadena = "Lorem ipsum dolor sit amet"
```

```
In [7]: dos_mas_dos + numero_flotante
```

```
Out[7]: 7.16
```

```
In [8]: dos_mas_dos + cadena
```

```
-----  
TypeErrorTraceback (most recent call last)
```

```
<ipython-input-8-050e4683ce0b> in <module>()  
----> 1 dos_mas_dos + cadena
```

```
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

```
In [11]: string_4 = str(dos_mas_dos)
```

```
In [13]: cadena + str(dos_mas_dos)
```

```
Out[13]: 'Lorem ipsum dolor sit amet4'
```

```
In [15]: # En esta linea voy a guardar el resultado de pi  
         # para ello, necesito importar el módulo math  
         import math
```

```
pi_guardado = math.pi
```

```

In [23]: """
        Resulta que existe otra forma de exportar la funcion pi
        Ahora la vamos a ver

        """

        from math import pi, sqrt

In [46]: cadena_1 = "tapir, ornitorrinco, perro"
        cadena_2  = "gato, tortuga, hurón"

In [43]: cadena_3 = "Perro"

In [48]: cadena_3.

Out[48]: 'Perro'

In [57]:

Out[57]: True

```

2 Elementos básicos de Python

2.1 Comentarios

```

In [ ]: # Comentarios simples

        """
        Comentarios múltiples
        """

```

2.2 Variables

Toda operación que asigne mediante el operación = se guardará en un nombre de la variable

2.3 Ingreso de datos

Podemos ingresar datos con `sys.argv` para leer desde la consola

Podemos ingresar datos del usuario con un prompt mediante `input`

Todo valor ingresado por éstos métodos devolverá un string. Es necesario coercionar su tipo de dato,

2.4 Introducción a Objetos

2.4.1 Objetos y métodos

- Todo objeto creado en Python tendrá una clase asociada, y una serie de funciones y acciones que podrá realizar

2.4.2 Tipos de datos

```
In [63]: 5 == 6
```

```
Out[63]: False
```

```
In [64]: lista_1 = list(range(10))
```

Evalua de izquierda a derecha, salvo que definamos operaciones dentro de paréntesis.

```
In [70]: 'Gato' in lista_1 == 5 == 5
```

```
Out[70]: False
```

2.4.3 Distintas formas de interpolación

```
In [ ]: nombre = input("Escribe tu nombre")
        edad = int(input("Escribe tu edad"))

        if edad > 18:
            # Forma uno de Python 3.x
            print("{} -> {}".format(nombre, edad))
            # Forma dos, de python 3.5
            print(f'{nombre} -> {edad}')
            # forma antigua, python 2.x
            print("%s -> %s" % (nombre, edad))
```

2.4.4 Usos de if, elif y else

- If permite evaluar una condición.
- Elif permite evaluar una condición secundaria **siempre y cuando la primera no se cumpla**.
- Else genera una condición para todos aquellos valores que no satisfagan la condicieron

```
In [ ]: nombre = input("Escribe tu nombre")

        if (nombre not in "Bernarda") or (nombre not in "Magdalena"):
            print("No eres Bernarda, vuelve a intentar")
        elif (nombre in "Bernarda") or (nombre in "Magdalena"):
            edad = int(input("Y tu edad?"))
            if edad < 50:
                print("Estás en el mercado laboral")
            else:
                print("Deberías jubilar")
```

2.5 Importancia de saber con qué tipo de dato trabajamos

```
In [5]: lista_heterogenea = ['a', 5.32, True, 'gato', 5, [1, 2, 3]]
```

```
In [6]: lista_heterogenea
```

```
Out[6]: ['a', 5.32, True, 'gato', 5, [1, 2, 3]]
```

Mediante el tipo de dato podemos realizar una serie de func

```
In [4]: for i in lista_heterogenea:
        if type(i) is str:
            print(i + " paralelepipedo", " es cadena y genera una concatenación")
        if type(i) is int:
            print(i + 1, " es entero y devuelve una suma de entero")
        if type(i) is float:
            print(i + 5, " es flotante y devuelve una serie con parte decimal")
        if type(i) is list:
            print(i, " es una lista y por tanto puedo seguir iterando en ella")
            for j in i:
                print(j, " estoy iterando en una lista y cada uno de sus elementos")
```

```
a paralelepipedo es cadena y genera una concatenación
10.32 es flotante y devuelve una serie con parte decimal
gato paralelepipedo es cadena y genera una concatenación
6 es entero y devuelve una suma de entero
[1, 2, 3] es una lista y por tanto puedo seguir iterando en ella
1 estoy iterando en una lista y cada uno de sus elementos
2 estoy iterando en una lista y cada uno de sus elementos
3 estoy iterando en una lista y cada uno de sus elementos
```