Introduccion Python

### Primer Programa De Python

El primer programa que reaizamos en cualquier lenguaje de programacion es nuestro “Hello World”

print('Hello World!')

Hello World!

Generamos un error a proposito, para ver que nos dice

print('Hello World!)

SyntaxError: unterminated string literal (detected at line 1) (3571950873.py, line 1)

Asignar una variable el saludo y la vamos a imprimir por consola

saludo = 'Hola mundo!'  
print(saludo)

Hola mundo!

### Variables

Definicion de una variable: - En Python, una variable es una etiqueta a la que podemos asignar valores y hace referencia a un valor determinado.

Reglas para nombrar variables: - Solo pueden contener letras, numeros y guiones bajos. - No estan permitidos los espacios (es util seguir el metodo **Camelcase**). - Evitar el uso de palabras clave y nombres de funcion propios del lenguaje - Usar nombres cortos, pero descriptivos - Python es **Case Sensitive**, distingue minusculas de mayusculas

# Ejemplo de declaracion de variable  
nombre = 'Manu'  
print(nombre)  
  
mensaje = 'Bienvenidos al curso de Python en CIFO Barcelona La Violeta'  
print(mensaje)

Manu  
Bienvenidos a curs de Python en CIFO Barcelona La Violeta

### Tipos de datos primitivos

Tenemos cuatro tipos de datos primitivos - Numericos enteros - Numericos decimales o coma flotante - Cadenas de caracteres - Logicos o booleanos

# Ejemplos de tipos de datos  
numeroEntero = 50  
numeroFlotante = 2.7  
cadena = 'hola'  
valorLogico = True  
  
print(type(numeroEntero))  
print(type(numeroFlotante))  
print(type(cadena))  
print(type(valorLogico))

<class 'int'>  
<class 'float'>  
<class 'str'>  
<class 'bool'>

### Datos alfanumericos (strings)

Formas de declarar **strings** - ‘Hola hola caracola - “Hola hola caracola” - ’Hola “hola” caracola’ - “Hola ‘hola’”caracola” - ‘Texto ’entre comillas’ y no pasa nada’

print('Hola hola caracola')  
print("Hola hola caracola")  
print('hola "hola" caracola')  
print("Hola 'hola' caracola")  
print("Hola \'hola\' caracola")

Hola hola caracola  
Hola hola caracola  
hola "hola" caracola  
Hola 'hola' caracola  
Hola 'hola' caracola

cadena = 'Cadena partida \  
en varias lineas \  
de codigo para facilitar su lectura'  
print(cadena)

Cadena partida en varias lineas de codigo para facilitar su lectura

Cadena con saltos entre ellas

cadena1 = 'Cadena con \nsaltos de linea \nentre palabras'  
print(cadena1)

Cadena con   
saltos de linea   
entre palabras

### Otra forma de escribir cadenas largas

cadena2 = ''' Otra forma de escribir  
cadenas largas  
de varias lineas '''  
print(cadena2)

Otra forma de escribir  
cadenas largas  
de varias lineas

## Metodos con strings

Un metodo es una accion que Python puede realizar con los datos. Cada metodo va seguido de un parentesis porque suelen requerir informacion adiciona para realizar su trabajo. En la sintaxis vamos a usar a metodologia de punto: **variable.metodo()**

### Cambio de mayusculas y minusculas con metodos

El metodo **tittle()** cambia cada palabra a formato titulo, con la inicial en mayusculas en todas las palabras

name = 'don pollo'  
print(name)  
print(name.title())

don pollo  
Don Pollo

Otros metodos: - capitalize() - upper(), transforma en mayusculas la cadena - lower(), transforma en minusculas la cadena

print(name.capitalize())

Don pollo

print(name.upper())  
print(name.lower())

DON POLLO  
don pollo

Otros metodos tambien son: - .swapcase(), invierte minusculas y mayusculas - .islower() y .isupper() devuelve **True** o **False**

personaje = 'Alberto Instantaneo'  
print(personaje.swapcase())

aLBERTO iNSTANTANEO

mayuscula = 'A'  
print(mayuscula.isupper())  
print(mayuscula.islower())

True  
False

### Metodos de longitud y acceso a una string

frase = 'Erase una vez'

Metodo len(), nos permite calcular la longitud de la cadena de caracteres

print(len(frase))

13

Para acceder a cada una de las posiciones de una string. Podemos usar su indice. Ejemplo: accedemos al primer caracter con la posicion 0

print(frase[0])

E

Para acceder a la ultima posicion, cuando no conocemos la longitud de la cadena, podemos usar dos versiones.

print(frase[len(frase)-1])  
print(frase[-1])

z  
z

### Trabajando con plantillas (strings y variables)

Para insertar el valor de una variable en una cadena, colocamos la letra **“f”** justo antes de abrir las comillas, y ponemos entre llaves e nombre o nombres de las variables. Python reemplazara por el valor de las variables **“f”** correponde aa formato y estas cadenas se llaman **“cadenas f”**

firstname = 'don'  
lastname = 'pollo'  
fullname = f'{firstname} {lastname}'  
print(fullname)

don pollo

print(f'Hello, {fullname}')

Hello, don pollo

print(f'Hello, {fullname.title()}')

Hello, Don Pollo

message = f'Hello, {fullname.title()}'  
print(message)

Hello, Don Pollo

### Añadir espacipos de tabulacion y nuevas lineas

* \*\* añade un espacio de tabulacion
* \*\*\* añade una nueva linea

print('Python')  
print('\tPython')

Python  
 Python

print('Lenguajes: \nPython\nJavascript\nC#')  
print('Lenguajes: \n\tPython\n\tJavascript\n\tc#')

Lenguajes:   
Python  
Javascript  
C#  
Lenguajes:   
 Python  
 Javascript  
 c#

### Funcion str()

Ejemplo que nos va a generar un error

edad = 25  
nombre = 'Pepe'  
frase1 = 'El alumno' + nombre + 'tiene' + edad + 'años'  
print(frase1)

TypeError: can only concatenate str (not "int") to str

frase1 = f'El alumno {nombre} tiene {edad} años'  
print(frase1)

El alumno Pepe tiene 25 años

frase2 = 'El alumno ' + nombre + ' tiene ' + str(edad) + ' años'  
print(frase2)

El alumno Pepe tiene 25 años

### Otras operaciones

Repetir una cadena **‘n’** veces

repite = 'ola '  
cancion = repite \* 5  
print(cancion)

ola ola ola ola ola

### Otra forma de realizar plantillas con cadenas y variables

Metodo **“.format()”**

presentacion = 'Hola, soy {}'.format('Pepe')  
print(presentacion)

Hola, soy Pepe

nombre = 'Pepe'  
presentacion = 'Hola, soy {}'.format(nombre)  
print(presentacion)

Hola, soy Pepe

mascota = 'Tommy'  
plantilla = 'Hola, soy {} y mi mascota se llama {}'  
print(plantilla.format(nombre, mascota))

Hola, soy Pepe y mi mascota se llama Tommy

plantillaAlternativa = 'Hola, soy {1} y mi mascota se llama {0}'  
print(plantillaAlternativa.format(nombre, mascota))

Hola, soy Tommy y mi mascota se llama Pepe

## Slicing en las cadenas

El **“slicing”** es la capacidad de las cadenas que devuelve un subconjunto **“(substring o subcadena)”** usando los indices **“[inicio:fin]”** - El primer indice **“(inicio)”** es donde empieza el substring **(se incuye ese caracter)**. - El segundo indice **(fin)** es donde acaba la **“substring”** **(se excluye ese caracter)**.

Las cadenas son inmutables, no se pueden modificar. Ejemplo, vamos a intentar cambiar un caracter de la cadena anterior.