

# Conociendo Python

Este cuaderno forma parte del curso de [Iniciación a la programación con Python](#) del programa de Open Course Ware (OCW) de la Universidad de La Laguna.

## Sintaxis de Python

La sintaxis es el conjunto de reglas que determinan como se escribe el código en Python. Hay que tener en cuenta:

### Comentarios

Comienzan con un `#`. Por ejemplo:

```
# Esto es un comentario
```

Podemos sacar información por pantalla utilizando la función `print()`. Por ejemplo:

```
print("Este texto sale por pantalla al ejecutar el código")
```

### Indentación

Permiten definir los bloques de código y controlar la estructura y el flujo del programa. Estructuras como los bucles, condicionales o funciones lo necesitan. Normalmente se suelen utilizar 4 espacios para cada nivel. Por ejemplo:

```
x=9
if x>5:
    print(x,"es mayor que 5")
print("Esta línea está fuera del condicional")
```

Puedes probar este tema en la siguiente celda de código.

```
x=9 #Variable x que vale 9
if x>5: #Si el dato de x es mayor que 5, se ejecuta la siguiente línea
    print(x,"es mayor que 5") #Imprime por pantalla el texto: 9 es mayor
que 5
print("Esta línea está fuera del condicional") #El texto 'Esta línea
está fuera del condicional' siempre se imprime por pantalla
```

## Tipos de datos

Categorías que representan diferentes tipos de información que pueden ser manejados por Python. En concreto se verá:

- Enteros, `int`: números sin parte decimal.

- Flotantes, `float`: números con parte decimal.
- Cadena de texto, `str`: secuencias de caracteres.
- Booleano, `bool`: representa un valor binario (verdadero o falso).
- None, `NoneType`: representa la ausencia de valor.

Python, por defecto, asignará a nuestros datos el tipo de manera automática. Por ejemplo:

```
3 #Entero
3.5 #Flotante
"¡Hola, Mundo!" #Cadena de texto
True #Booleano
None #None
```

Si queremos forzar el tipo de datos, se puede ejecutar las siguientes funciones:

```
int(3) #Entero
float(3.5) #Flotante
str("¡Hola, Mundo!") #Cadena de texto
bool(1) #Booleano
```

Para conocer de que tipo es un dato, se puede ejecutar la función `type()`. Por ejemplo:

```
type(3.5)
```

Estos tipos de datos podrían considerarse básicos. En siguientes temas trabajaremos con aún más tipos de datos. Puedes probar este tema en la siguiente celda de código.

```
x=3
y=3.0
print(type(x)) #Saldrá por pantalla que el tipo de dato es un entero
print(type(y)) #Saldrá por pantalla que el tipo de dato es un flotante
y=int(y)
print(type(y)) #Saldrá por pantalla que el tipo de dato es un entero
y=str(y)
print(type(y)) #Saldrá por pantalla que el tipo de dato es una cadena de texto
```

## Variables

Las variables son espacios de memoria en los que se almacenan datos. Se le tiene que asignar un nombre a cada una para poder acceder a ella. Para asignar un valor se utiliza el símbolo `=`. Por ejemplo:

```
x="¡Hola, Mundo!"
y=3.0
print(x) #Imprime ¡Hola, Mundo! por pantalla
print(y) #Imprime el número 3.0 por pantalla
```

Las cadenas de texto pueden representarse tanto con comillas simples (' '), como dobles (" ").

Los nombres de las variables son sensibles a mayúsculas y no es posible utilizar espacios (tampoco el símbolo -), no puede comenzar por números y no puede ser una palabra reservada, como `True` o `False`.

## Operadores

Permiten realizar diferentes acciones con datos, como cálculos matemáticos o manipular los propios datos. Se tratarán 4 tipos: aritméticos, de asignación, de comparación y lógicos.

### Operadores aritméticos

Sirven para realizar cálculos matemáticos, por ejemplo:

- Suma, `+`.
- Resta, `-`.
- Multiplicación, `*`.
- División, `/`.
- Módulo (resto de una división), `%`.
- Potencia, `**`.
- División (parte entera), `//`.

Es posible utilizar estos operadores combinando enteros con números flotantes. Por ejemplo:

```
print(3+4) #Suma. El resultado es 7
print(8-3.4) #Resta. El resultado es 4.6
print(6*7) #Multiplicación. El resultado es 42
print(32/4.7) #División. El resultado es 6.8085106...
print(45%10) #Módulo. El resultado es 5
print(8**2) #Potencia. El resultado es 64
print(32//4.7) #División parte entera. El resultado es 6
```

Además de números, es posible utilizar los operadores de suma, `+`, y de multiplicación, `*` con cadenas de texto. Por ejemplo:

- Concatenación, `+`: une dos cadenas de texto.
- Repetición, `*`: repite la misma cadena de texto tantas veces como se especifique.

```
x="¡Hola, "
y="Mundo!"
z="¡Adiós! "
print(x+y) #Concatenación. El resultado es '¡Hola, Mundo!'
print(z*3) #Repetición. El resultado es '¡Adiós! ¡Adiós! ¡Adiós! '
```

### Operadores de asignación

Sirven para asignar valores a las variables, siempre el valor de la derecha del operador a la variable de la izquierda. Por ejemplo:

- Igual, `=`: asigna un valor.
- Suma y asignación, `+=`: suma el valor de la derecha a la variable de la izquierda y asigna ese resultado a la variable de la izquierda.
- Resta y asignación, `-=`: resta el valor de la derecha a la variable de la izquierda y asigna ese resultado a la variable de la izquierda.
- Multiplicación y asignación, `*=`: multiplica el valor de la derecha por el de la variable de la izquierda y asigna ese resultado a la variable de la izquierda.
- División y asignación, `/=`: divide el valor de la derecha por el de la variable de la izquierda y asigna ese resultado a la variable de la izquierda.
- Módulo y asignación, `%=`: calcula el módulo determinado en el valor de la derecha sobre la variable de la izquierda y asigna ese resultado a la variable de la izquierda.
- Potencia y asignación, `**=`: calcula la potencia del valor de la derecha sobre la variable de la izquierda y asigna ese resultado a la variable de la izquierda.
- División entera y asignación, `//=`: divide (la parte entera) el valor de la derecha por el de la variable de la izquierda y asigna ese resultado a la variable de la izquierda.

```
x=4
x+=4
print(x) #Suma. La variable x vale 8
x-=3.4
print(x) #Resta. La variable x vale 4.6
x*=7
print(x) #Multiplicación. La variable x vale 32.199999...
x/=4.7
print(x) #División. La variable x vale 6.851063
x=13
x%=10
print(x) #Módulo. La variable x vale 3
x**=2
print(x) #Potencia. La variable x vale 9
x//=4.7
print(x) #División parte entera. La variable x vale 1
```

## Operadores de comparación

Nos permiten comparar los valores de dos elementos, dando un resultado de `True` o `False`.  
Por ejemplo:

- Igualdad, `==`: comprueba si dos valores son iguales.
- Diferente, `!=`: comprueba si dos valores son diferentes.
- Menor, `<`: comprueba si el valor de la izquierda es menor que el de la derecha.
- Mayor, `>`: comprueba si el valor de la izquierda es mayor que el de la derecha.
- Menor o igual, `<=`: comprueba si el valor de la izquierda es menor o igual que el de la derecha.
- Mayor o igual, `>=`: comprueba si el valor de la izquierda es mayor o igual que el de la derecha.

```

x=9
y=11.2
print(x==y) #Igualdad. El resultado es False
print(x!=y) #Diferente. El resultado es True
print(x<y) #Menor. El resultado es True
print(x>y) #Mayor. El resultado es False
print(x<=y) #Menor o igual. El resultado es True
print(x>=y) #Mayor o igual. El resultado es False

```

## Operadores lógicos

Es posible combinar o modificar expresiones booleanas utilizando operadores lógicos. Por ejemplo:

- Y lógico, `and`: devuelve `True` si AMBAS expresiones son verdaderas. Si al menos una es falsa, devuelve `False`.
- O lógico, `or`: devuelve `True` si al menos una de las expresiones es verdadera. Si ambas son falsas, devuelve `False`.
- No lógico, `not`: devuelve el valor inverso de la expresión. Si la expresión era `True`, devuelve `False`, y viceversa.

```

x=8.2
y=7
print(x>y and x!=y) #Y lógico. El resultado es True
print(x>y and x==y) #Y lógico. El resultado es False
print(x>y or x==y) #O lógico. El resultado es True
print(x<y and x==y) #O lógico. El resultado es False
print(not(x!=y)) #No lógico. El resultado es False

```

Además de estos operadores, existen más, e incluso algunos que trabajan a un nivel de bits.

Puedes probar este tema en la siguiente celda de código.

```

#Operadores aritméticos
x = 65
y = 9.8

print("La suma de x e y es",x+y)
print("La resta de x e y es",x-y)
print("La multiplicación de x por x es",x*x)
print("La potencia de x elevado a 3 es",x**3)

#Operadores de asignación
x=30
y=10
print("x vale",x)
y+=x
print("x+y vale",y)
y*=y

```

```
print("y*y vale",y)

#Operadores de comparación
if x==y:
    print("x e y son iguales") #Esto se ejecutará si se cumple la
    condición anterior

if x!=y:
    print("x e y no son iguales")

if x>y:
    print(x,"es mayor que",y)

#Operadores lógicos
print(x<y and x!=y)
print(x<y or x==y)
print(not(x<y) and x!=y)
```