

CURSO DE PYTHON3



```
print("Hello, world!")
```



ffmancera



IEEE

INDICE – SESIÓN 1

- Introducción
- Tipo de datos y como tratarlos
- Control de flujo

INTRODUCCIÓN

- Ventajas de usar Python:
 - Sintaxis sencilla y curva de aprendizaje rápida
 - Potencia en el lenguaje
 - Gran variedad de módulos y funciones
 - Multiplataforma
 - Una comunidad consolidada y en continuo crecimiento
- Desventajas principales:
 - Menos rendimiento que otros lenguajes
 - No recomendado para sistemas críticos en tiempo real

TIPOS DE DATOS

- **¿Como declaramos un dato?**

Boolean: Dato binario (True o False)

int(Integer): número entero

float: número en coma flotante

str(String): Cadena de caracteres (Texto)

tuple(T, K): Par de valores

list(List): Lista de datos

dict(Dictionary): Conjunto clave-valor

*** Ver tiposdedatos.py**

OPERADORES

OPERADOR	OPERACIÓN	EJEMPLO
**	Exponente	$2^{**}4 = 16$
%	Modulo	$12 \% 5 = 2$
//	División con numero entero	$3 // 2 = 1$
/	División	$3 / 2 = 1.5$
*	Multiplicación	$3 * 2 = 6$
—	Resta	$3 - 2 = 1$
+	Suma	$3 + 2 = 5$

EJERCICIO 1

- Se quiere realizar una lista de la compra, para ello vamos a tener una tupla o par (x, y) que contenga como valor “x” el nombre del elemento y como valor “y” la cantidad de elementos.
- Datos:
 - Manzanas = 2
 - Caja de Cereales = 1
 - Caja de Leche = 1
 - Huevos = 12
 - Bollos de pan = 3
 - Botellines de cerveza = 48

COMPARADORES

OPERADOR	SIGNIFICADO	EJEMPLO
==	Igual a	7 == 7 True
!=	Diferente de	7 != 7 False
<	Menor que	7 < 6 False
>	Mayor que	7 > 6 True
<=	Menor o igual que	7 <= 7 True
>=	Mayor o igual que	7 >= 7 True
& and	AND Lógico	True and False False
 or	OR Lógico	True or False True

CONDICIONALES (IF, ELIF, ELSE)

- **“IF”** pregunta por un Booleano o en su defecto por una expresión que devuelva un Booleano. Si el resultado es True se ejecutará la acción.
- **“ELIF”** pregunta por un Booleano si la condición anterior no se ha cumplido, si es True se ejecutará la acción.
- **“ELSE”** Si no se cumple ninguna de las condiciones anteriores entonces se ejecutará lo que esté indicado.

* Ver condicionales.py

BUCLES (WHILE, FOR)

- “**WHILE**” mientras se cumpla la condición inicial se ejecutará la acción indicada luego hay que actualizar la condición en cada iteración o nos quedaremos en un bucle infinito.
- “**FOR**” se le marca un numero de iteraciones ya sea por elementos de una estructura de datos o un simple integer.
- “**BREAK**” es el elemento que finaliza el bucle en caso de que se cumpla alguna condición.

* Ver bucles.py

EJERCICIO 2

- Con los datos del ejercicio anterior, vamos a sustituir en dict anterior, la cantidad por una tupla donde tengamos el la cantidad y el precio por unidad (cantidad, precio) y usaremos un bucle for para calcular el precio total de nuestra lista de la compra
- Datos:
 - Manzana: 0.20€
 - Caja de cereales: 1.20€
 - Caja de leche: 0.40€
 - Huevo: 0.25€
 - Bollo de pan: 0.15€
 - Botellin de cerveza: 0.40€