

# Introducción a Programación

---

David Medina Ortiz



# Python y los lenguajes de programación

---

## Lenguajes Compilados

Los lenguajes compilados crear un ejecutable desde el código fuente.

Dicho ejecutable es el que se corre y hace las acciones.

Ejemplos de lenguajes compilados: C, C++.

Dentro de las principales ventajas, son su procesamiento de cálculo, velocidad de ejecución, entre otros

## Lenguajes Interpretados

Normalmente nacen de un lenguaje compilado

Representan simplificaciones en entender la programación.

A diferencia de los lenguajes compilados, se ejecutan en tiempo real, interpretando línea a línea.

Ejemplos clásicos son Python, Perl.

Dentro de las principales ventajas se encuentran su fácil interpretación

# Qué es un Algoritmo?

---

- Un algoritmo es un conjunto ordenado de acciones o tareas que se definen para cumplir con un objetivo.
- Ejemplos de algoritmos:
- Sumar dos números
- Ir a comprar al supermercado
- Iniciar una caldera a vapor

# Python como lenguaje de programación

---

- Lenguaje interpretado
- Lenguaje indentado
- Palabras reservadas propias del lenguaje: for, while, if, else, not, in, entre otras
- No es necesario definir variables, pero es posible hacerlo. Tomando en consideración que las variables representan el nombre del valor que queremos almacenar.
- Se aprenderá a definir variables, trabajar con condiciones, ciclos, implementar nuestras propias funciones, entre otros temas.

# Variables

---

Las variables son espacios en memoria que sirven para almacenar un valor que nosotros deseamos. Siempre se declaran como

***variable = valor de la variable***

Existen diferentes tipos,

int, float, double para numéricos

str, char para palabras y letras

listas, diccionarios, para tipos de datos complejos

Objetos, para tipos de datos abstractos

# Imprimiendo variables en python

---

Imprimir una variable en pantalla se define como *“mostrar por salida estándar”*, para ello, Python tiene implementado la función **print**, la cual se usa como sigue

`print(conjunto de caracteres a imprimir)`, donde lo que va entre paréntesis va entre `“”` si es una cadena de texto o el nombre de la variable simplemente.

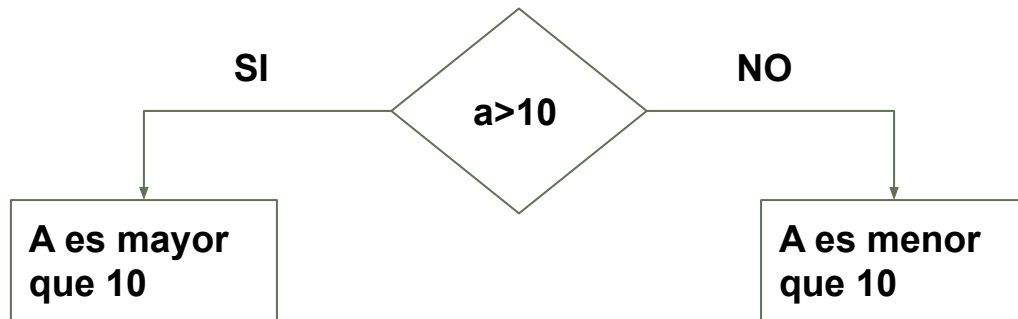
Dado a que recibe un str o cadena de caracteres, estos se pueden formatear ocupando la función `format`

*`“{} {} {} {}”.format(valor1, valor2, “valor2”, valor3)` donde los {} se reemplazan en orden por la variable o valor en los paréntesis.*

# Estructuras de control

---

Las estructuras de control representan bloques de código que se asocian a una condición o conjunto de condiciones, definidas por *if* (si en inglés)



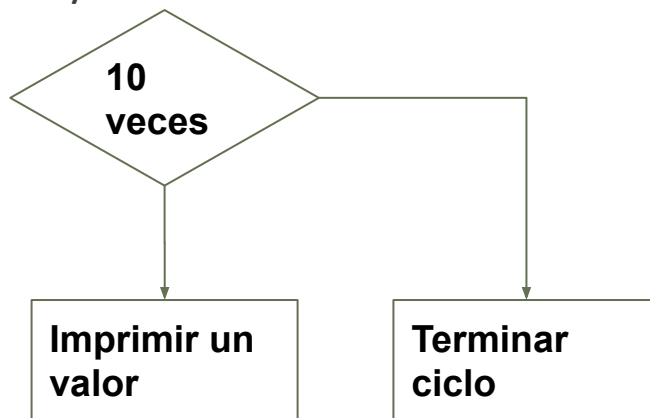
***Siempre las estructuras de control se pueden observar como el esquema previamente. También se pueden anidar, incluir más condiciones en una misma pregunta, etc.***

# Estructuras de Ciclo

---

Los ciclos, son segmentos de código que se repiten hasta que se cumpla una condición, la condición puede ser con un número de ciclos definidos, por ejemplo “ejecutar acción X por 10 veces” o puede ser con un número variable, por ejemplo “ejecutar acción X hasta que Z cumpla la condición Y”.

Existen dos sentencias claves: ***For*** y ***While***.

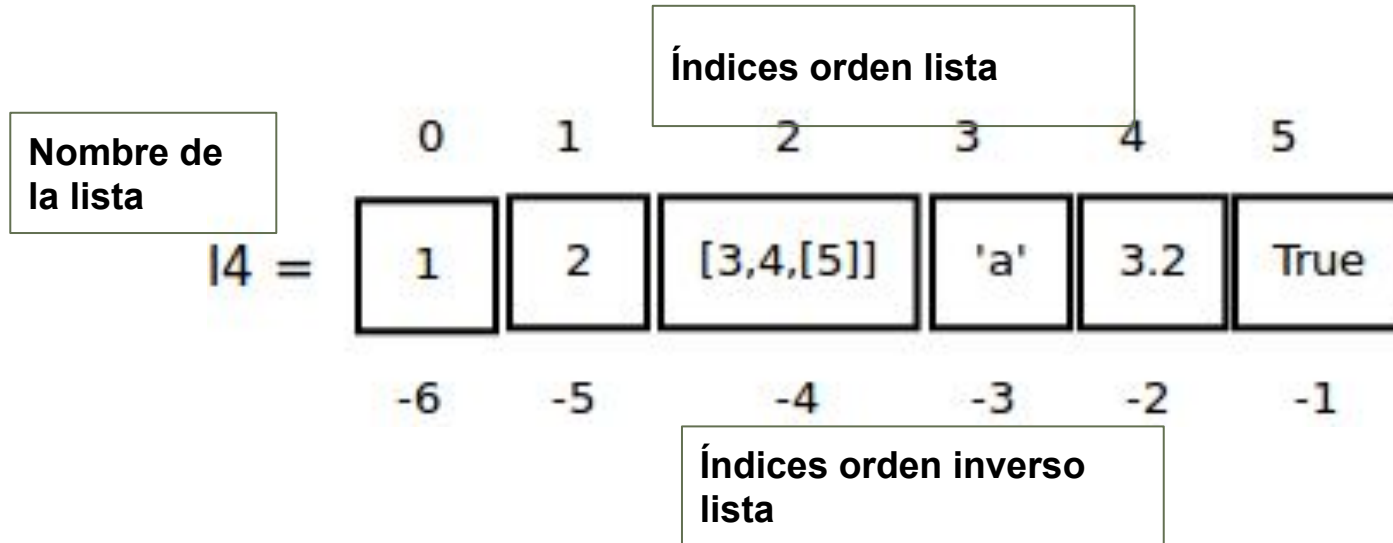




# Listas

---

Las listas son estructuras de datos que permiten almacenar variables en un determinado orden.



# Diccionarios

---

Los diccionarios, representan un tipo de dato que permite asociar una clave con un valor, donde dicho valor puede ser cualquier cosa

***diccionario = {"clave" : variable}***

La variable puede ser de todos los tipos posibles nombrados previamente.

# Funciones

---

Adicionales a las funciones existentes en python, también es posible implementar nuestras propias funciones.

***def funcion (parametros):***

***bloque de código***

***return variable***

La estructura previamente expuesta facilita la implementación de una función, **def** indica que se nombrará una función **función** representa el nombre (**parámetros**) representan variables que ingresarán a una función y **return** hace referencia a si la función devolverá algún valor