

# Tipo de datos

Programación y Laboratorio I

# Tipos de datos

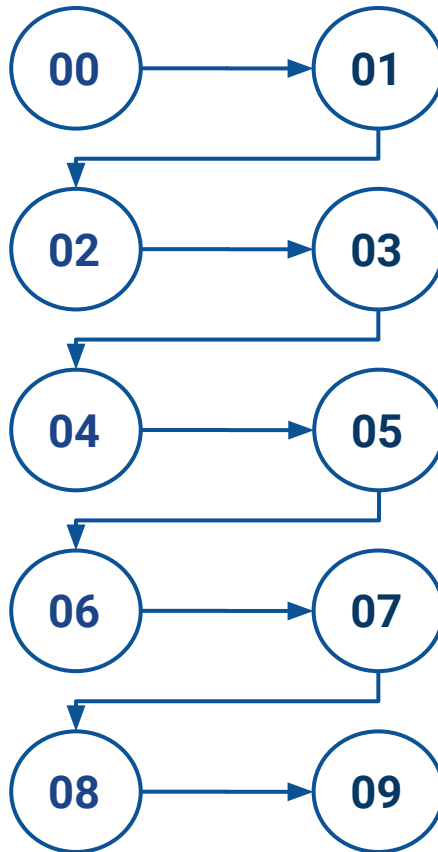
¿Qué es un tipo de dato?

Flotante

Booleano

Listas

Diccionarios



Entero

Complejos

Cadena de caracteres

Tuplas

Conjuntos

# ¿Qué es un tipo de dato?

Todos los valores que aparecen en un programa tienen un tipo. Cuando los programas almacenan datos en variables, a cada variable se le debe asignar un tipo de datos específico.

Diferentes tipos de datos son adecuados para diferentes tipos de usos.

# Entero: **int**

Permite representar números enteros, es decir, positivos y negativos no decimales.

```
numero = 23  
print(numero)           # 23  
print(type(numero))     #<class 'int'>
```

# Flotante: float

Permite representar un números positivo o negativo con decimales

```
numero = 3.14  
print(numero) # 3.14  
print(type(numero)) #<class 'float'>
```

# Complejos: **complex**

Los números complejos son aquellos que tienen dos partes: una parte real y otra imaginaria.

```
c = 3 + 5j
print(c)           # (3+5j)
print(c.real)      # 3.0
print(c.imag)      # 5.0
print(type(c))     # <class 'complex'>
```

# Booleano: True, False

Es un tipo de dato que permite almacenar dos valores **True** o **False**.

```
x = True
```

```
y = False
```

# Cadena de caracteres: **str**

Los strings son un tipo **inmutable** que permite almacenar secuencias de caracteres.

Para crear una, es necesario incluir el texto entre comillas.

```
nombre = 'Veronica'
```

```
apellido = "Perez"
```

```
print(type(nombre))      # <class 'str'>
```

```
print(type(apellido))    # <class 'str'>
```



# Listas: **list**

Son uno de los tipos más versátiles del lenguaje, ya que permiten almacenar un conjunto arbitrario de datos.

```
lista = [1, "Hola", 3.67]
print(type(lista))      # <class 'list'>
print(lista[1])         # Hola
lista[1] = "Chau"
print(lista[1])         # Chau
```

# Tuplas: **tuple**

Las tuplas son muy similares a las listas, pero son inmutables, lo que significa que no pueden ser modificadas una vez declaradas.

```
lista = tuple([1, "Hola", 3.67])  
print(type(lista))           # <class 'tuple'>  
print(lista[1])              # Hola  
lista[1] = "Chau"            # object does not support item assignment
```

# Diccionarios: dict

Un diccionario es una colección de elementos, donde cada uno tiene una clave (**key**) y un valor (**value**).

```
diccionario = {'nombre' : 'Juan', 'edad' : 21}  
print(diccionario['nombre'])    # Juan  
print(diccionario['edad'])      # 21
```

# Conjuntos: **set**

Los elementos de un set son únicos, no contiene elementos duplicados.

Los set no respetan el orden que tenían al ser declarados.

Sus elementos son inmutables.

# Conjuntos: set

Un set se puede crear haciendo uso de { }

```
s = {2, 4, 7, 1, 8, 1}
print(s)           # {1, 2, 4, 7, 8}
print(type(s))     #<class 'set'>
```

# Conjuntos: **set**

Un set también se puede crear haciendo uso de la palabra reservada **set** la cual permite transformar cualquier objeto iterable en un **set**

```
lista = [1, 3, 6, 3, 2, 1]
s = set(lista)
print(s)           # {1, 2, 3, 6}
print(type(s))     # <class 'set'>
```