

Tipos de datos avanzados

Listas

Las listas en Python, representadas por el tipo *list*, son colecciones ordenadas y **mutables** de elementos que pueden ser de diferentes tipos de datos. Se definen utilizando corchetes [...] y permiten almacenar, acceder, modificar y eliminar elementos de forma flexible.

```
a = [1,2,3]
b = [4,5,6]
c = a + b
print(f"El valor de a es {a}")
print(f"El valor de b es {b}")
print(f"El valor de c es {c}")
print(f"El tipo de a es: {type(a)}")
print(f"El tipo de b es: {type(b)}")
print(f"El tipo de c es: {type(c)}")
```

```
El valor de a es [1, 2, 3]
El valor de b es [4, 5, 6]
El valor de c es [1, 2, 3, 4, 5, 6]
El tipo de a es: <class 'list'>
El tipo de b es: <class 'list'>
El tipo de c es: <class 'list'>
```

Ejemplo: Crear una lista que contenga un elemento de cada tipo básico

```
lista = [5, 5.34, 2+2j, True, "Manzana"]
print(f"El tipo de la lista es: {type(lista)}")
print(f"El primer elemento de la lista es: {lista[0]} y su tipo es: \
{type(lista[0])}")
print(f"El segundo elemento de la lista es: {lista[1]} y su tipo es: \
{type(lista[1])}")
```

```
print(f"El tercer elemento de la lista es: {lista[2]} y su tipo es: \
{type(lista[2])}")
print(f"El cuarto elemento de la lista es: {lista[3]} y su tipo es: \
{type(lista[3])}")
print(f"El quinto elemento de la lista es: {lista[4]} y su tipo es: \
{type(lista[4])}")
```

```
El tipo de la lista es: <class 'list'>
El primer elemento de la lista es: 5 y su tipo es: <class 'int'>
El segundo elemento de la lista es: 5.34 y su tipo es: <class 'float'>
El tercer elemento de la lista es: (2+2j) y su tipo es: <class 'complex'>
El cuarto elemento de la lista es: True y su tipo es: <class 'bool'>
El quinto elemento de la lista es: Manzana y su tipo es: <class 'str'>
```

Tuplas

Las tuplas en Python, representadas por el tipo *tuple*, son colecciones ordenadas e **inmutables** de elementos que pueden ser de diferentes tipos de datos. Se definen usando paréntesis (...) y, una vez creadas, sus elementos no pueden ser modificados ni eliminados.

Ejemplo de uso de variables de tipo tuple

```
a = (1,2,3)
b = (4,5,6)
c = a + b
print(f"El valor de a es: {a}")
print(f"El valor de b es: {b}")
print(f"El valor de c es: {c}")
print(f"El tipo de a es: {type(a)}")
print(f"El tipo de b es: {type(b)}")
print(f"El tipo de c es: {type(c)}")
```

```
El valor de a es: (1, 2, 3)
El valor de b es: (4, 5, 6)
El valor de c es: (1, 2, 3, 4, 5, 6)
El tipo de a es: <class 'tuple'>
El tipo de b es: <class 'tuple'>
El tipo de c es: <class 'tuple'>
```

Ejemplo: crear una tupla que contenga un elemento de cada tipo básico

```

tupla = (5, 5.34, 2+2j, True, "Manzana")
print(f"El tipo de tupla es: {type(tupla)}")
print(f"La primera posición de la tupla es {tupla[0]} y su tipo es: \
{type(tupla[0])}")
print(f"La segunda posición de la tupla es {tupla[1]} y su tipo es: \
{type(tupla[1])}")
print(f"La tercera posición de la tupla es {tupla[2]} y su tipo es: \
{type(tupla[2])}")
print(f"La cuarta posición de la tupla es {tupla[3]} y su tipo es: \
{type(tupla[3])}")
print(f"La quinta posición de la tupla es {tupla[4]} y su tipo es: \
{type(tupla[4])}")

```

```

El tipo de tupla es: <class 'tuple'>
La primera posición de la tupla es 5 y su tipo es: <class 'int'>
La segunda posición de la tupla es 5.34 y su tipo es: <class 'float'>
La tercera posición de la tupla es (2+2j) y su tipo es: <class 'complex'>
La cuarta posición de la tupla es True y su tipo es: <class 'bool'>
La quinta posición de la tupla es Manzana y su tipo es: <class 'str'>

```

Conjuntos (sets)

Los conjuntos en Python, representados por el tipo *set*, son colecciones desordenadas de elementos únicos y **mutables**. Se definen utilizando llaves {...} y permiten realizar operaciones matemáticas como unión, intersección y diferencia, además de eliminar elementos duplicados.

Ejemplo de uso de variables de tipo set

```

a = {1,2,3}
b = {4,5,6}
c = a.union(b)
print(f"El valor de a es: {a}")
print(f"El valor de b es: {b}")
print(f"El valor de c es: {c}")
print(f"El tipo de a es: {type(a)}")
print(f"El tipo de b es: {type(b)}")
print(f"El tipo de c es: {type(c)}")

```

```

El valor de a es: {1, 2, 3}
El valor de b es: {4, 5, 6}
El valor de c es: {1, 2, 3, 4, 5, 6}

```

```
El tipo de a es: <class 'set'>
El tipo de b es: <class 'set'>
El tipo de c es: <class 'set'>
```

Ejemplo: crear un conjunto que contenga un elemento de cada tipo básico

```
conjunto = {5, 5.34, 2+2j, True, "Manzana"}
print(f"El tipo de conjunto es: {type(conjunto)}")
# Lo siguiente dará error, debido a que un conjunto no está indexado.
# No podemos acceder a él mediante índices,
print(f"El primer elemento de conjunto es: {conjunto[0]}")
```

El tipo de conjunto es: <class 'set'>

TypeError: 'set' object is not subscriptable

```
[1;31m-----[0m
[1;31mTypeError[0m                                Traceback (most recent call last)
```

```
Cell [1;32mIn[47], line 5[0m
```

```
[0;32m      2[0m [38;5;28mprint[39m([38;5;124mf[39m[38;5;124m"[39m[38;5;124mEl tipo de conjun
[0;32m      3[0m [38;5;66;03m# Lo siguiente dará error, debido a que un conjunto no está ind
[0;32m      4[0m [38;5;66;03m# No podemos acceder a él mediante índices,[39;00m
[1;32m----> 5[0m [38;5;28mprint[39m([38;5;124mf[39m[38;5;124m"[39m[38;5;124mEl primer elemen
```

```
[1;31mTypeError[0m: 'set' object is not subscriptable
```

Diccionarios

Los diccionarios en Python, representados por el tipo *dict*, son colecciones desordenadas de pares **clave-valor**. Cada clave es única y se utiliza para acceder a su valor asociado. Se definen utilizando llaves { } con pares separados por dos puntos (*clave: valor*).

Ejemplo de uso de variables de tipo dict

```
diccionario = {"nombre": "José Ortega",
               "edad": 29,
               "ciudad": "Tijuana"}
print(f"Nombre: {diccionario['nombre']}")
print(f"Edad: {diccionario['edad']}")
print(f"Ciudad: {diccionario['ciudad']}")
print(f"El tipo de diccionario es: {type(diccionario)}")
```

Nombre: José Ortega
Edad: 29
Ciudad: Tijuana
El tipo de diccionario es: <class 'dict'>

Ejemplo: Crear un diccionario con elementos de cada tipo básico

```
persona = {"nombre": "José Ortega",  
           "edad": 29,  
           "estatura": 1.74,  
           "estudiando": True}  
print(f"El tipo del valor asociado a la clave nombre es: \  
{type(persona["nombre"])}")  
print(f"El tipo del valor asociado a la clave edad es: \  
{type(persona["edad"])}")  
print(f"El tipo del valor asociado a la clave estatura es: \  
{type(persona["estatura"])}")  
print(f"El tipo del valor asociado a la clave estudiando es: \  
{type(persona["estudiando"])}")
```

El tipo del valor asociado a la clave nombre es: <class 'str'>
El tipo del valor asociado a la clave edad es: <class 'int'>
El tipo del valor asociado a la clave estatura es: <class 'float'>
El tipo del valor asociado a la clave estudiando es: <class 'bool'>