

Tipos de datos

Los tipos de datos definen un conjunto de valores que tienen una serie de características y propiedades determinada. un tipo de dato establece qué valores puede tomar una variable y qué operaciones se pueden realizar sobre la misma.

Tipos de datos básico:

- Numérico (enteros, punto flotante, complejos)
- Cadena de caracteres
- Booleano

Otros tipos:

- Secuencias (list, tuple, range)
- Mapas (dict)
- Conjuntos (set)
- Iteradores
- Clases
- Instancias
- Excepciones

Numéricos (Enteros)

- Tipo de dato **int**.
- Representa al conjunto de los números enteros.
- El conjunto está limitado por la memoria disponible (Python no da límites para este conjunto).
- Ejemplos de los números enteros: 1, -7, 5, 0 , -2, 3.
- Podemos representar a los números enteros en formato binario, octal o hexadecimal:
 - binario: anteponer **0b** y luego colocar valores de 0 o 1. Ej: 0b1001 -> 9
 - octal: anteponer **0o** y luego colocar valores del 0 al 7. Ej: 0o1037 -> 543
 - hexadecimal: anteponer **0x** y luego colocar valores del 0 al 9 y/o A a la F. Ej: 0x2AE -> 756137

Numéricos (Punto flotante)

- Tipo de dato **float**.
- Representa al conjunto de los números reales. *(casi)
- El conjunto está limitado por la memoria disponible (Python no da límites para este conjunto).
- Ejemplos de punto flotantes:
 - 2.4
 - 3/3
 - 3.1E4
- Estos números se representan en el hardware como fracciones de base 2 (binarios). Como existen decimales que no se pueden representar con números binarios con exactitud, en determinados casos arroja una aproximación (Aproximación suficiente para trabajar con el. Ej 2.1 + 5.2).
- Para mayor precisión se deberá trabajar con el Tipo **Decimal**

Numéricos (Complejos)

- Tipo de dato **complex**.
- Los números complejos tienen una parte real y una imaginaria y cada una de ellas se representa como un **float**.
- Su estructura es de la siguiente manera [real] + [imaginario] j.
- Para acceder a las partes se realiza a través de los atributos **real** e **imag**.
- Ejemplos de números complejos:
 - $2.4 + 1j$
 - $3 - 5.3j$
 - $4j$

Booleano

- Tipo de dato **bool**.
- Solo se le pueden asignar 2 valores, **True** (Verdadero) y **False** (Falso).
- Los objetos son por defecto considerados como verdadero con dos excepciones:
 - Al implementar `__bool__()` retorne **False**
 - Al implementar `__len__()` retorne **0**.
- Los objetos mostrados a continuación son considerado **False**.
 - None**
 - False**
 - que valga **0** cualquier objeto de tipo numérico (**0** , **0.0** , **0 j**)
 - Secuencias y colecciones vacías: " " , () , [] , { } , **set()** , **range(0)**

Cadena de caracteres

- Tipo de dato **str**.
- Secuencia inmutable de caracteres en formato Unicode.
- Para la creación se utiliza comillas simples (' ') o comillas dobles (" ").
- Ejemplos de números complejos:
 - " hola mundo! "
 - ' hola mundo! '

Nota: Si se desea utilizar comillas dobles como parte de la cadena de caracteres, se debe encerrar la cadena con comillas simples o colocar una barra invertida (\) antes de la comilla doble que se desea colocar en el texto. Funciona igual para las comillas simples.

Unicode: Estándar de
codificación de caracteres

Cadena de caracteres

- Colocar barra invertida (\) para indicar caracteres especiales.
- Para múltiples líneas, se debe utilizar `""" ... """` o `''' ... '''`.
- Para concatenar cadenas se puede utilizar el operador `+`.
- Para repetir cadenas de caracteres se puede utilizar el operador `*`.
- Dos cadenas literales una al lado de la otra se concatenan automáticamente (solo funciona con dos literales).
- Las cadenas se pueden indexar, retornando otra cadena de caracteres.

Conversión de tipos

- `str(parámetro)`: devuelve un **str** (cadena de caracteres) lo que se pase como parámetro.
- `int(parámetro)`: devuelve un **int** lo que se pase como parámetro.
- `float(parámetro)`: devuelve un **float** lo que se pase como parámetro
- `complex(parámetro)`: devuelve un **complex** lo que se pase como parámetro

Nota: Para valores de parámetros no validos, el intérprete lanzará un error.