

A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light greenish-blue. They are positioned diagonally, with the blue one partially covering the green one.

Objetos y estructuras de datos básicas



Números (int)

Todos los números enteros, tales como:

- 0, 1, 2, -31, 11, 22, 1000000



Números (float)

Números con decimales, tales como:

- 1.3
- 0.2
- 0.00002
- 11.11
- 09.11
- 22.22
- 213123.14123
- 3.14159265359



Strings (str)

Representan a las cadenas de texto, tales como:

- "Hola mundo"
- "Mi mamá me mima mucho"
- ""
- "Una cadenaaaaa muy laaaaaaaargaaaaa"
- 'También van con comillas simples'
- ''
- """Triples comillas dobles se usan para textos largooooooss, docstrings"""""



Listas (list)

- Se pueden definir con sintaxis propia del lenguaje
 - `a = [1, 2, 3, 4]`
- ó usando el builtin “list”
 - `a = list()`
- Soportan operaciones como suma (resta no)
 - `[1, 2, 3] + [4, 5, 6]`
- Soporta múltiples tipos de datos
 - `[1, 2, “hello”, 12.12, Author(name=“Gabriel”)]`
- Las listas son ordenadas
 - `[1, 2, 3, 4] != [4, 1, 3, 2]`
- Se puede acceder por índices (también índices negativos :-S)
 - `a[0], a[2], a[-1]`
- Son mutables



Diccionarios (dict)

- Se pueden definir con sintaxis propia del lenguaje
 - `a = {"key1": 1, "key2": 2, "key3": [1, 2, 3]}`
- ó usando el builtin "dict"
 - `a = dict(key1=1, key2=2, key3=[1, 2, 3])`



Tuplas (tuple)

- Se pueden definir con sintaxis propia del lenguaje
 - `a = (1, 2, 3, 4)`
- ó usando el builtin “tuple”
 - `a = tuple()`
- Soportan operaciones como suma (resta no)
 - `(1, 2, 3) + (4, 5, 6)`
- Soporta múltiples tipos de datos
 - `(1, 2, “hello”, 12.12, Author(name=“Gabriel”))`
- Las tuplas son ordenadas
 - `(1, 2, 3, 4) != (4, 1, 3, 2)`
- Se puede acceder por índices (también índices negativos :-S)
 - `a[0], a[2], a[-1]`
- Son inmutables



Conjuntos (set)

- Se pueden definir con sintaxis propia del lenguaje
 - `a = {1, 2, 3, 4}`
- ó usando el builtin “set”
 - `a = set()`
- Soporta múltiples tipos de datos, siempre y cuando sean hashtables
 - `{1, 2, “hello”, 12.12}`
- Son inmutables



Booleanos (bool)

- Los valores booleanos se componen de dos constantes True y False
- Se usan operadores lógicos “and” y “or”
 - True and True -> True
 - True and False -> False
 - True or False -> True
 - False or False -> False