



UAX

UNIVERSIDAD ALFONSO X EL SABIO

Tipos de datos y estructuras

GRADO EN BIOMEDICINA

Bioinformática

idelhgar@uax.es

2023

Números

- Python tiene varios tipos de números. Principalmente dos:
- **Enteros o integers:** números positivos o negativos, enteros.
- **Floating point:** números con decimales. También los exponenciales como 4E2

Examples	Number "Type"
1,2,-5,1000	Integers
1.2,-0.5,2e2,3E2	Floating-point numbers

Variables

- Hay una serie de reglas para nombrar variables:
 1. **No empezar por números**
 2. **No pueden tener espacios en el nombre, es mejor usar “_”**
 3. **No se pueden usar símbolos especiales como: ”’.,<>/?|()!@#\$~%&-+**
 4. **Siempre se considera mejor que se usen las variables en minúsculas**
 5. **Evitar usar los caracteres ‘l’, ‘O’ o ‘I’ (l minúscula, i mayúscula y o mayúscula)**
 6. **Evitar usar palabras con significado especial en Python como list o str**

Strings

- Guardan piezas de texto, secuencias etc. Son secuencias de caracteres.
- Se emplean las comillas.
- Son inmutables, una vez creada los elementos ya no se pueden modificar o cambiar.
- Sí que se pueden concatenar strings.
- Hay funciones útiles como **upper()**, **lower()** o **split()**
- **Las strings se pueden imprimir con print(), format() o el nuevo f-strings**
- [2. Lexical analysis — Python 3.11.5 documentation](#)

Listas

- Las listas son como las strings, secuencias de caracteres PERO a diferencia de las strings, podemos cambiar sus elementos. **Son mutables**
- **Van entre corchetes con los elementos separados por comas**
- Pueden albergar distintas clases de elementos
- Funciones útiles como **append()**, **pop()**, **sort()**, **reverse()**
- Podemos también hacer matrices con ellas uniéndolas entre sí
- También podemos usar las **list_comprehensions** para crear listas rápidamente. Aquí va unido el uso de loops.

Diccionarios

- Los diccionarios también se les llama **hash**
- Son estructuras que nos permiten almacenar información mediante una **key** y un **valor**
- **El valor puede ser cualquier objeto de Python, incluso otro diccionario**
- **Las keys tienen que tener un valor único**
- Las funciones más empleadas son **keys()** y **items()**

Tuplas

- Son como las listas pero **INMUTABLES**
- En lugar de corchetes llevan paréntesis
- Podemos usar las funciones **index()** y **count()**
- Se usan cuando queremos crear un objeto que NO pueda ser cambiado

Sets

- Colección ordenada de elementos **ÚNICOS**
- Los construimos con la función **.set()**
- Podemos añadir elementos con la función **add()**
- Los usamos cuando queremos eliminar duplicados y tener solamente elementos únicos, como las keys de un diccionario.

Booleanos

- Elementos de verdadero o falso

- **TRUE**

- **FALSE**

- **NONE**

Operator	Description	Example
==	If the values of two operands are equal, then the condition becomes true.	(a == b) is not true.
!=	If values of two operands are not equal, then condition becomes true.	(a != b) is true.
>	If the value of left operand is greater than the value of right operand, then condition becomes true.	(a > b) is not true.
<	If the value of left operand is less than the value of right operand, then condition becomes true.	(a < b) is true.
>=	If the value of left operand is greater than or equal to the value of right operand, then condition becomes true.	(a >= b) is not true.
<=	If the value of left operand is less than or equal to the value of right operand, then condition becomes true.	(a <= b) is true.