

# Introducción a la programación

## Usando Python

Mateo Suster  
mateosuster@gmail.com

Matemática para Economistas III  
Instituto de Industria  
Universidad Nacional de General Sarmiento

6 de mayo de 2022

# Tipos de Datos y Comparaciones

- ▶ Igualdad: `i == k`

# Tipos de Datos y Comparaciones

- ▶ Igualdad: `i == k`
  - ▶ Probar `2 == 3`, `4 == 4`, `'a' == 'a'`

# Tipos de Datos y Comparaciones

- ▶ Igualdad: `i == k`
  - ▶ Probar `2 == 3`, `4 == 4`, `'a' == 'a'`
- ▶ Distinto: `i != k`

# Tipos de Datos y Comparaciones

- ▶ Igualdad: `i == k`
  - ▶ Probar `2 == 3`, `4 == 4`, `'a' == 'a'`
- ▶ Distinto: `i != k`
  - ▶ Probar `2 != 3`

# Tipos de Datos y Comparaciones

- ▶ Igualdad:  $i == k$ 
  - ▶ Probar  $2 == 3$ ,  $4 == 4$ ,  $'a' == 'a'$
- ▶ Distinto:  $i != k$ 
  - ▶ Probar  $2 != 3$
- ▶ Menor:  $i < k$

# Tipos de Datos y Comparaciones

- ▶ Igualdad:  $i == k$ 
  - ▶ Probar  $2 == 3$ ,  $4 == 4$ ,  $'a' == 'a'$
- ▶ Distinto:  $i != k$ 
  - ▶ Probar  $2 != 3$
- ▶ Menor:  $i < k$
- ▶ Mayor:  $i > k$

# Tipos de Datos y Comparaciones

- ▶ Igualdad:  $i == k$ 
  - ▶ Probar  $2 == 3$ ,  $4 == 4$ ,  $'a' == 'a'$
- ▶ Distinto:  $i != k$ 
  - ▶ Probar  $2 != 3$
- ▶ Menor:  $i < k$
- ▶ Mayor:  $i > k$
- ▶ Menor o igual:  $i \leq k$



# Tipos de Datos y Comparaciones

- ▶ Igualdad:  $i == k$ 
  - ▶ Probar  $2 == 3$ ,  $4 == 4$ ,  $'a' == 'a'$
- ▶ Distinto:  $i != k$ 
  - ▶ Probar  $2 != 3$
- ▶ Menor:  $i < k$
- ▶ Mayor:  $i > k$
- ▶ Menor o igual:  $i \leq k$
- ▶ Mayor o igual:  $i \geq k$

## Tipos de Datos: Listas

Una *lista* es una colección de valores (**definida entre corchetes**) que se acceden mediante un índice:

## Tipos de Datos: Listas

Una *lista* es una colección de valores (**definida entre corchetes**) que se acceden mediante un índice:

-index	-6	-5	-4	-3	-2	-1
list1	88	99	4.12	199	993	9999
index	0	1	2	3	4	5

## Tipos de Datos: Listas

Una *lista* es una colección de valores (**definida entre corchetes**) que se acceden mediante un índice:

-index	-6	-5	-4	-3	-2	-1
list1	88	99	4.12	199	993	9999
index	0	1	2	3	4	5

Ojo, el primer elemento tiene índice 0

# Tipos de Datos: Listas

Una *lista* es una colección de valores (**definida entre corchetes**) que se acceden mediante un índice:

-index	-6	-5	-4	-3	-2	-1
list1	88	99	4.12	199	993	9999
index	0	1	2	3	4	5

Ojo, el primer elemento tiene índice 0

Y el último elemento tiene índice -1

# Más sobre listas

Algunos ejemplos de listas y operaciones:

- ▶ [2, 3, 5]

# Más sobre listas

Algunos ejemplos de listas y operaciones:

- ▶ `[2, 3, 5]` ¿Cuál es el elemento con índice 1?

# Más sobre listas

Algunos ejemplos de listas y operaciones:

- ▶ `[2, 3, 5]` ¿Cuál es el elemento con índice 1?
- ▶ `[2, 3.5, 'cosita', 8]`



## Más sobre listas

Algunos ejemplos de listas y operaciones:

- ▶ `[2, 3, 5]` ¿Cuál es el elemento con índice 1?
- ▶ `[2, 3.5, 'cosita', 8]`
- ▶ `[]` lista vacía.

## Más sobre listas

Algunos ejemplos de listas y operaciones:

- ▶ `[2, 3, 5]` ¿Cuál es el elemento con índice 1?
- ▶ `[2, 3.5, 'cosita', 8]`
- ▶ `[]` lista vacía.
- ▶ `c = [2, 3, 5]` define una lista de nombre `c`.

## Más sobre listas

Algunos ejemplos de listas y operaciones:

- ▶ `[2, 3, 5]` ¿Cuál es el elemento con índice 1?
- ▶ `[2, 3.5, 'cosita', 8]`
- ▶ `[]` lista vacía.
- ▶ `c = [2, 3, 5]` define una lista de nombre `c`.
- ▶ `c[i]` accede o devuelve el elemento con índice `i` de la lista.

## Más sobre listas

Algunos ejemplos de listas y operaciones:

- ▶ `[2, 3, 5]` ¿Cuál es el elemento con índice 1?
- ▶ `[2, 3.5, 'cosita', 8]`
- ▶ `[]` lista vacía.
- ▶ `c = [2, 3, 5]` define una lista de nombre `c`.
- ▶ `c[i]` accede o devuelve el elemento con índice `i` de la lista.
- ▶ `len(c)` devuelve la longitud de la lista.

# Más sobre listas

Algunos ejemplos de listas y operaciones:

- ▶ `[2, 3, 5]` ¿Cuál es el elemento con índice 1?
- ▶ `[2, 3.5, 'cosita', 8]`
- ▶ `[]` lista vacía.
- ▶ `c = [2, 3, 5]` define una lista de nombre `c`.
- ▶ `c[i]` accede o devuelve el elemento con índice `i` de la lista.
- ▶ `len(c)` devuelve la longitud de la lista.
- ▶ `c.append(x)` agrega el elemento `x` al final de la lista `c`.
- ▶ Y muchas operaciones más que iremos viendo...

En resumen, ¿qué es una lista?

# En resumen, ¿qué es una lista?

PYnative.com

## List in Python

```
L = [ 20, 'Jessa', 35.75, [30, 60, 90] ]
```

↑  
L[0]

↑  
L[1]

↑  
L[2]

↑  
L[3]

- ✓ **Ordered**: Maintain the order of the data insertion.
- ✓ **Changeable**: List is mutable and we can modify items.
- ✓ **Heterogeneous**: List can contain data of different types
- ✓ **Contains duplicate**: Allows duplicates data