



Repaso conciso, aplicativo e intenso de Python

Técnicas de programación II



Ciclo 2022-II, UNALM
04/10/2022

Tipos de datos

¿Qué tipos de datos identificas?

¿Falta alguno?

	Fecha	Día	PP [mm]	TMAX [°C]	TMIN [°C]	HR [%]
0	2022-01-31	Lunes	0.0	21.2	16.3	98
1	2022-02-28	Lunes	0.1	21.3	18.2	99
2	2022-03-31	Jueves	0.0	20.6	19.8	100
3	2022-04-30	Sábado	0.0	20.0	20.1	96
4	2022-05-31	Martes	NaN	21.0	NaN	95
5	2022-06-30	Jueves	1.1	20.8	17.5	96
6	2022-07-31	Domingo	2.0	20.9	16.4	98
7	2022-08-31	Miércoles	2.4	20.4	18.9	94

String

Float

Integer

Tipos de datos

¿Qué tipos de datos identificas?

¿Qué día llovió?

	Fecha	Día	PP [mm]	TMAX [°C]	TMIN [°C]	HR [%]	Llovió
0	2022-01-31	Lunes	0.0	21.2	16.3	98	False
1	2022-02-28	Lunes	0.1	21.3	18.2	99	True
2	2022-03-31	Jueves	0.0	20.6	19.8	100	False
3	2022-04-30	Sábado	0.0	20.0	20.1	96	False
4	2022-05-31	Martes	NaN	21.0	NaN	95	False
5	2022-06-30	Jueves	1.1	20.8	17.5	96	True
6	2022-07-31	Domingo	2.0	20.9	16.4	98	True
7	2022-08-31	Miércoles	2.4	20.4	18.9	94	True

PP > 0

Boolean

Tipos de datos

¿Qué tipos de datos identificas?

	Fecha	Día	PP [mm]	TMAX [°C]	TMIN [°C]	HR [%]	Coherencia
0	2022-01-31	Lunes	0.0	21.2	16.3	98	True
1	2022-02-28	Lunes	0.1	21.3	18.2	99	True
2	2022-03-31	Jueves	0.0	20.6	19.8	100	True
3	2022-04-30	Sábado	0.0	20.0	20.1	96	False
4	2022-05-31	Martes	NaN	21.0	NaN	95	False
5	2022-06-30	Jueves	1.1	20.8	17.5	96	True
6	2022-07-31	Domingo	2.0	20.9	16.4	98	True
7	2022-08-31	Miércoles	2.4	20.4	18.9	94	True

¿Notas algo raro entre la TMAX y TMIN?

TMAX > TMIN

Boolean

Tipos de datos

¿Cómo convertir a otro tipo?

¿Cuál es el resultado?

```
1  # Calculadora
2  num_1 = input('Inserte número 1: ')
3  num_2 = input('Inserte número 2: ')
4
5  num_3 = num_1 + num_2
6  print(num_3)
```

Inserte número 1: 7

Inserte número 2: 9

⇒ “79”

Conversión a:

- float: float()
- integer: int()
- string: str()

```
1  # Calculadora
2  num_1 = input('Inserte número 1: ')
3  num_2 = input('Inserte número 2: ')
4
5  num_3 = int(num_1) + int(num_2)
6  print(num_3)
```

Inserte número 1: 7

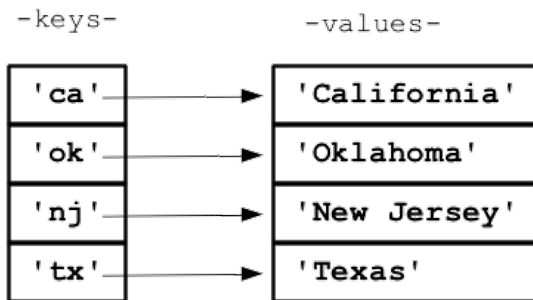
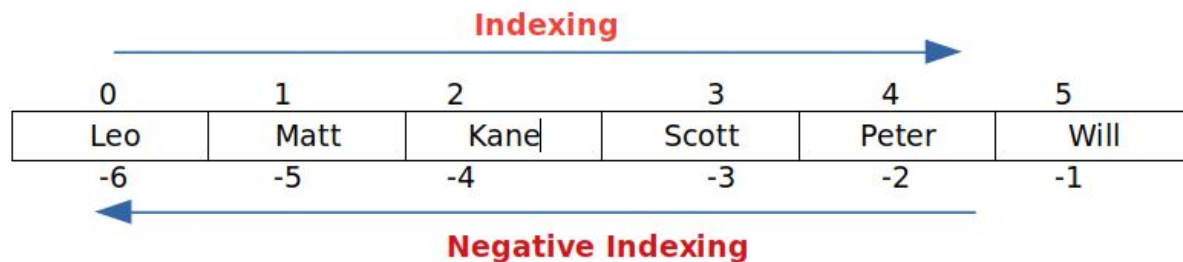
Inserte número 2: 9

16

Estructura de datos

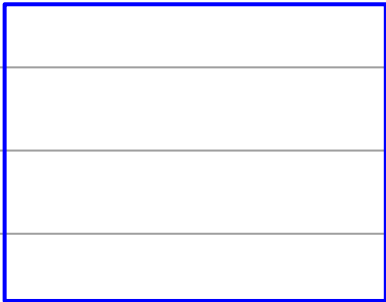
Almacenan datos en una variable

Lista	Tupla	Diccionario
<ul style="list-style-type: none">• Corchetes: <code>[True, 'hola', 1, 1.5]</code>• Mutable• Orden dado por la indexación	<ul style="list-style-type: none">• Paréntesis: <code>(True, 'hola', 1, 1.5)</code>• <u>I</u>nmutable• Orden dado por la indexación	<ul style="list-style-type: none">• Llaves: <code>{'a': True, 2: 'hola'}</code>• Mutable• No tiene orden predefinido
[1] Cuando necesites almacenar datos [2] que vayas a reemplazar o agregar posteriormente.	[1] Cuando necesites almacenar datos.	[1] Cuando necesites almacenar datos [2] relacionados con otro dato.



Estructura de datos

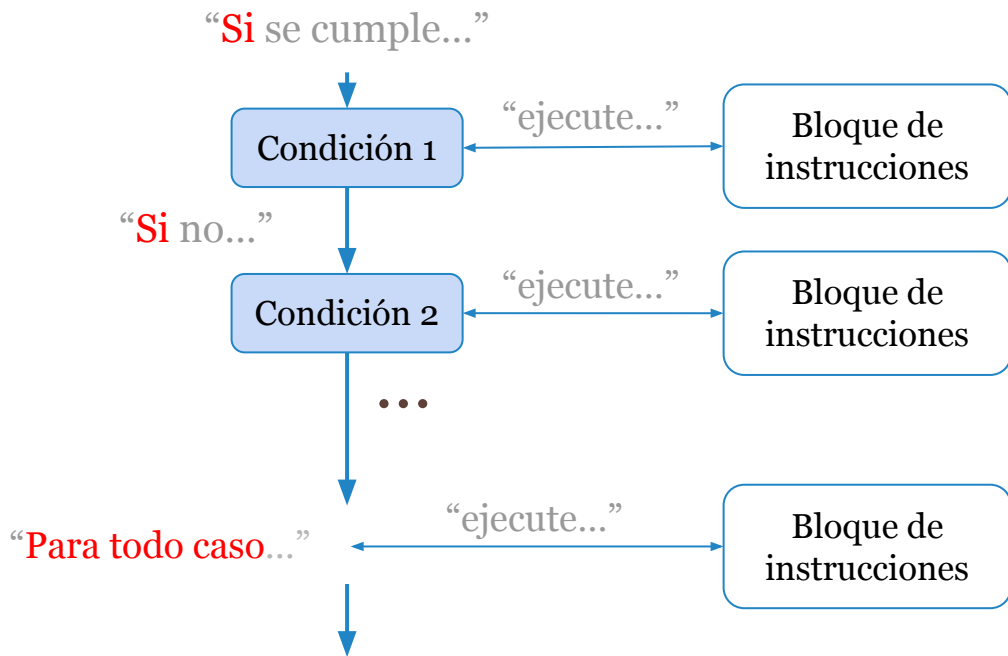
Manipulación básica

	Lista	Tupla	Diccionario
Crear	Corchetes []	Paréntesis ()	Llaves { }
Selección	list[Index]	tuple[Index]	dict[Key]
Agregar	list.append()		dict[new_Key] = new_value
Remover	list.remove()		del dict[Key]
Ordenar	list.sort()		sorted(dict)
Unir	list_1 + list_2		dict_1.update(dict_2)

Inmutabilidad

Condicionales

Pseudocódigo

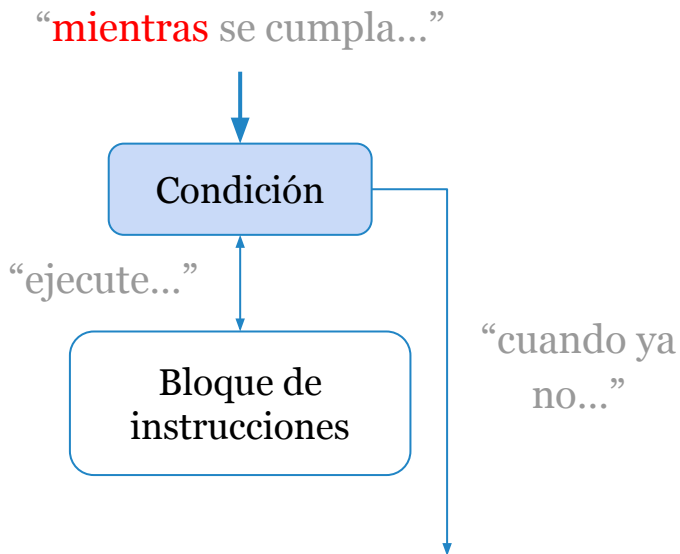


Código

```
if (condición 1):  
    Bloque de instrucciones  
elif (condición 2):  
    Bloque de instrucciones  
else:  
    Bloque de instrucciones
```


Bucle while

Pseudocódigo

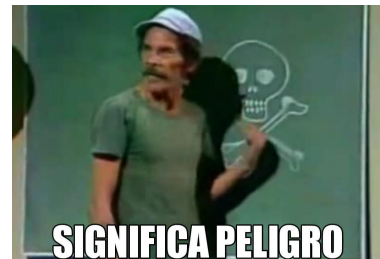


Código

```
while (condición):
```

```
    Bloque de instrucciones
```




¡Cuidado con los bucles
infinitos!






Bucle while

Identifiquemos bucles infinitos

```
1  # ¿Es un bucle infinito?
2
3  while 1 <= 10:
4      print('ABCD')
5
6  while True:
7      print('ABCD')
8
9  i = 0
10 while i != 10:
11     print('ABCD')
12
```



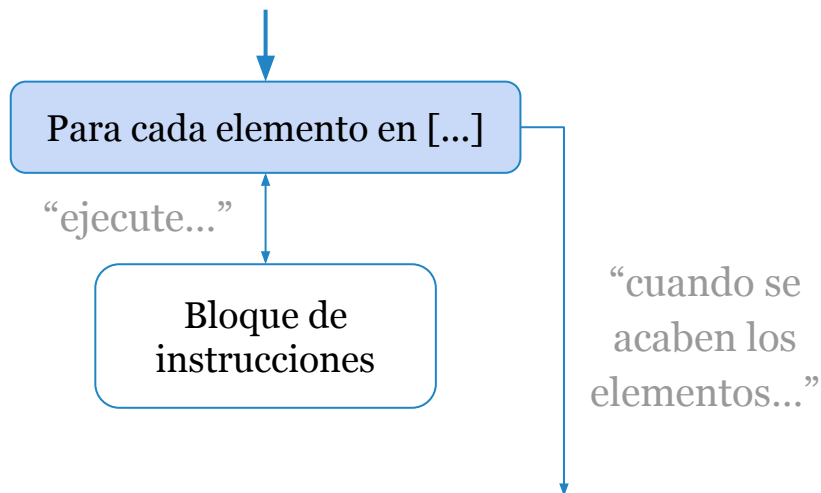
```
13 i = 0
14 while i != 10:
15     print('ABCD')
16     i = i + 2
17
18 while 10:
19     print('ABCD')
20
21 Soy_un_bucle_infinito = None
22 while Soy_un_bucle_infinito:
23     print('ABCD')
```



Bucle for

Pseudocódigo

Estructura de datos



Código

```
for i in Iterable:
```

```
    Bloque de instrucciones
```

Comprehension

Simplifica la creación de estructuras

[`x+1 if condición else x+2` `for x in Iterable`]

Condicional

Bucle for

Tipo de
estructura

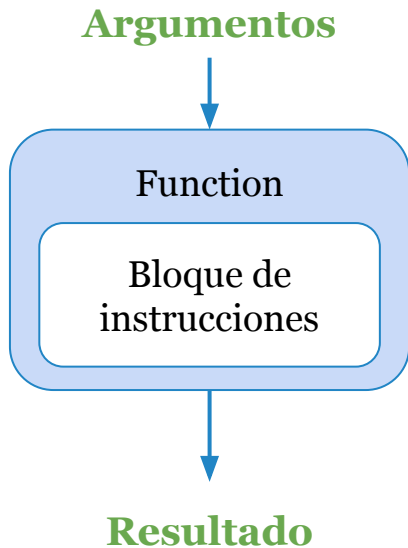
[] lista
tuple() tupla
{ : } diccionario

- Útil para:
 - crear largas series de números con una condición compleja (alternativa a Numpy).
 - modificar conjuntos de datos con una misma operación.
- Precaución: podría reducir el tiempo de ejecución del código si se usa muy seguido, especialmente en la creación de listas.
- Más información:
<https://realpython.com/list-comprehension-python/>

Funciones

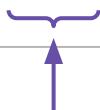
Simplifican las tareas repetitivas

Pseudocódigo



Código

```
def Nombre_de_la_funcion(argumentos):  
    ...  
    return resultado
```



Indentación

Delimita el bloque de código dentro y fuera de la función

Función lambda

Simplifica la creación de funciones

`lambda x: x + 1 if condición else x+2`

↑ ↑

Función lambda Argumento

Condicional

- Útil cuando tienes funciones cortas y sin bucles (for o while)
- Permite ingresar más de un argumento

`lambda x,y:`

- Más información: <https://realpython.com/python-lambda/>

Referencias recomendadas



- Tutoriales muy buenos y concisos.
- Ejemplos y ejercicios desde básicos a avanzados
- Comunidad, libros, etc.



- El “yahoo respuestas” para programadores
- Donde pasarán la mayor cantidad de tiempo cuando necesiten solucionar un problema.



- Biblioteca de repositorios para encontrar la “inspiración” (dile NO al plagio :v) que te falta.
- Recomendación: empieza a crear tu [portafolio](#).