Creado por:

Isabel Maniega

Estructura de datos de almacenamiento

En el presente documento se han incluído varios de los archivos de clase.

Incluye:

- Tuplas,
- Listas,
- Diccionarios,
- etc

Tuplas

In [1]:
```python
# 0, 1, 2, 3 ... Izquierda a derecha
# -1, -2, -3, ... Derecha a Izquierda
A = (10, 20, 30, 40)
A
```

Out[1]:
```
(10, 20, 30, 40)
```

In [2]:
```python
A[2]
```

Out[2]:
```
30
```

In [3]:
```python
A[0], A[1], A[2], A[3]
```

Out[3]:
```
(10, 20, 30, 40)
```

In [ ]:
```python
# No se puede ni modificar
# No se puede apendizar
```

Arrays

In [4]:
```python
import numpy as np
```

In [5]:
```python
B = np.array([1, 2, 3, 4])
B
```

Out[5]:
```
array([1, 2, 3, 4])
```

In [6]:
```python
B[0]
```

```
Out[6]: 1
```

```
In [7]: B[0] = 20
        B
```

```
Out[7]: array([20,  2,  3,  4])
```

```
In [8]: B1 = np.append(B, 50)
        B1
```

```
Out[8]: array([20,  2,  3,  4, 50])
```

```
In [9]: # se puede modificar
        # se pueder añadir/ apendizar
```

o concatenar...

```
In [10]: B2 = np.array([60])
         B2
```

```
Out[10]: array([60])
```

```
In [11]: # unir dos arrays
         B3 = np.concatenate((B1, B2))
         B3
```

```
Out[11]: array([20,  2,  3,  4, 50, 60])
```

Listas

```
In [12]: C = list((1, 2, 3, 4))
         C
```

```
Out[12]: [1, 2, 3, 4]
```

```
In [13]: D = [0, 2, 3, 5, 7, 8]
         D
```

```
Out[13]: [0, 2, 3, 5, 7, 8]
```

```
In [14]: D[2]
```

```
Out[14]: 3
```

```
In [15]: D[2] = 100
         D
```

```
Out[15]: [0, 2, 100, 5, 7, 8]
```

```
In [16]: D.append(200)
         D
```

```
Out[16]: [0, 2, 100, 5, 7, 8, 200]
```

In [17]:

```
# Eliminar por valor concreta
D.remove(200)
D
```

Out[17]:

```
[0, 2, 100, 5, 7, 8]
```

In [18]:

```
# Eliminar por posición concreta
D.remove(D[2])
D
```

Out[18]:

```
[0, 2, 5, 7, 8]
```

In [19]:

```
L = [1, 1, 2, 3, 6, 3, 3, 8, 10, 10, 5, 1]
L
```

Out[19]:

```
[1, 1, 2, 3, 6, 3, 3, 8, 10, 10, 5, 1]
```

In [20]:

```
# ordenar los valores
L.sort()
L
```

Out[20]:

```
[1, 1, 1, 2, 3, 3, 3, 5, 6, 8, 10, 10]
```

In [21]:

```
# Eliminar valores duplicados y colocarlos
L = set(L)
L
```

Out[21]:

```
{1, 2, 3, 5, 6, 8, 10}
```

In [22]:

```
type(L)
```

Out[22]:

```
set
```

In [23]:

```
L[0] = 100
```

```
---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
Cell In[23], line 1
----> 1 L[0] = 100

TypeError: 'set' object does not support item assignment
```

In [24]:

```
L.append(50)
```

```
---------------------------------------------------------------------------
-
AttributeError                            Traceback (most recent call las
t)
Cell In[24], line 1
----> 1 L.append(50)

AttributeError: 'set' object has no attribute 'append'
```

In [25]:

```
L1 = list(L)
L1
```

```
Out[25]: [1, 2, 3, 5, 6, 8, 10]
```

In [26]:

```python
L1[0] = 100
L1
```

```
Out[26]: [100, 2, 3, 5, 6, 8, 10]
```

In [27]:

```python
L1.append(500)
L1
```

```
Out[27]: [100, 2, 3, 5, 6, 8, 10, 500]
```

Otras cosas importantes

In [28]:

```python
listado = [3, 7, 5, 1]
listado
```

```
Out[28]: [3, 7, 5, 1]
```

In [29]:

```python
min(listado)
```

```
Out[29]: 1
```

In [30]:

```python
max(listado)
```

```
Out[30]: 7
```

In [31]:

```python
# Recomendación listas:
L = [1, 23, 26, 45, 48, 1, 3, 6, 8]
L
```

```
Out[31]: [1, 23, 26, 45, 48, 1, 3, 6, 8]
```

In [32]:

```python
# separar en grupos de 3 en 3:
L = [
    1, 23, 26,
    45, 48, 1,
    3, 6, 8
]
L
```

```
Out[32]: [1, 23, 26, 45, 48, 1, 3, 6, 8]
```

In [33]:

```python
test = [1, 2, 3, 4, 5, 6]
test
```

```
Out[33]: [1, 2, 3, 4, 5, 6]
```

In [34]:

```python
test.reverse()
test
```

```
Out[34]: [6, 5, 4, 3, 2, 1]
```

In [35]:

```python
test1 = enumerate(test)
test1
```

```
Out[35]: <enumerate at 0x7fdda029d680>
```

```
In [36]: list(test1)
```

```
Out[36]: [(0, 6), (1, 5), (2, 4), (3, 3), (4, 2), (5, 1)]
```

Transformación de entre ellas (Array/Lista)

Numpy array a lista

```
In [37]: import numpy as np
```

```
In [38]: E = np.array([1, 3, 4, 25, 1, 6, 7])
         E
```

```
Out[38]: array([ 1,  3,  4, 25,  1,  6,  7])
```

```
In [39]: E1 = E.tolist()
         E1
```

```
Out[39]: [1, 3, 4, 25, 1, 6, 7]
```

Lista a numpy Array

```
In [40]: F = [2, 3, 6, 8, 4, 12, 25]
         F
```

```
Out[40]: [2, 3, 6, 8, 4, 12, 25]
```

```
In [41]: F1 = np.array(F)
         F1
```

```
Out[41]: array([ 2,  3,  6,  8,  4, 12, 25])
```

```
In [42]: type(E1), type(F1)
```

```
Out[42]: (list, numpy.ndarray)
```

Matrices

Ejemplo 1: Teoría

```
In [43]: import numpy as np
```

```
In [44]: a = np.array([
             [10, 20, 30, 40],
             [50, 60, 70, 80],
             [90, 100, 110, 120]
         ])
         a
```

```
Out[44]: array([[ 10,  20,  30,  40],
                [ 50,  60,  70,  80],
                [ 90, 100, 110, 120]])
```

Ejemplo 2: Suma de matrices

```
In [45]: # np.array([ [fila1], [fila2]])
         matriz1 = np.array([[1, 2], [3, 4]])
         matriz1
```

```
Out[45]: array([[1, 2],
                [3, 4]])
```

```
In [46]: matriz2 = np.array([[5, 6], [7, 8]])
         matriz2
```

```
Out[46]: array([[5, 6],
                [7, 8]])
```

```
In [47]: matriz_suma = matriz1 + matriz2
         matriz_suma
```

```
Out[47]: array([[ 6,  8],
                [10, 12]])
```

Ejemplo 3: Multiplicación y división de matrices

```
In [48]: # np.array([ [fila1], [fila2]])
         matriz1 = np.array([[1, 2], [3, 4]])
         matriz1
```

```
Out[48]: array([[1, 2],
                [3, 4]])
```

```
In [49]: matriz2 = np.array([[5, 6], [7, 8]])
         matriz2
```

```
Out[49]: array([[5, 6],
                [7, 8]])
```

```
In [50]: # Multiplicación
         matriz_multiplicacion = 2 * matriz1
         matriz_multiplicacion
```

```
Out[50]: array([[2, 4],
                [6, 8]])
```

```
In [51]: # División
         matriz_division = matriz_multiplicacion / 2
         matriz_division
```

```
Out[51]: array([[1., 2.],
                [3., 4.]])
```

```
In [52]: matriz_division = matriz2 / 2
         matriz_division
```

```
Out[52]: array([[2.5, 3. ],
                [3.5, 4. ]])
```

Creado por:

Isabel Maniega