

Creado por:

Isabel Maniega

Estructura de datos de almacenamiento

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

Incluye:

- Tuplas,
- Listas,
- Diccionarios,
- etc

Tuplas

```
In [1]: # 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]: A[2]
```

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

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

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

```
In [4]: B = (1, 2)  
B
```

```
Out[4]: (1, 2)
```

```
In [5]: C = A + B  
C
```

```
Out[5]: (10, 20, 30, 40, 1, 2)
```

```
In [6]: A = A + B  
A
```

```
Out[6]: (10, 20, 30, 40, 1, 2)
```

```
In [7]: # No se puede ni modificar  
# No se puede apendizar
```

Arrays

```
In [8]: import numpy as np
```

```
In [9]: B = np.array([1, 2, 3, 4])  
B
```

```
Out[9]: array([1, 2, 3, 4])
```

```
In [10]: B[0]
```

```
Out[10]: 1
```

```
In [11]: B[0] = 20  
B
```

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

```
In [12]: B1 = np.append(B, 50)  
B1
```

```
Out[12]: array([20,  2,  3,  4, 50])
```

```
In [13]: # se puede modificar  
# se pueden añadir/ apendizar
```

o concatenar...

```
In [14]: B2 = np.array([60])  
B2
```

```
Out[14]: array([60])
```

```
In [15]: # unir dos arrays  
B3 = np.concatenate((B1, B2))  
B3
```

```
Out[15]: array([20,  2,  3,  4, 50, 60])
```

Listas

```
In [16]: C = list((1, 2, 3, 4))  
C
```

```
Out[16]: [1, 2, 3, 4]
```

```
In [17]: D = [0, 2, 3, 5, 7, 8]  
D
```

Out[17]: [0, 2, 3, 5, 7, 8]

In [18]: `D[2]`

Out[18]: 3

In [19]: `D[2] = 100`
`D`

Out[19]: [0, 2, 100, 5, 7, 8]

In [20]: `D.append(200)`
`D`

Out[20]: [0, 2, 100, 5, 7, 8, 200]

In [21]: *# Eliminar por valor concreta*
`D.remove(200)`
`D`

Out[21]: [0, 2, 100, 5, 7, 8]

In [22]: *# Eliminar por posición concreta*
`D.remove(D[2])`
`D`

Out[22]: [0, 2, 5, 7, 8]

In [23]: `L = [1, 1, 2, 3, 6, 3, 3, 8, 10, 10, 5, 1]`
`L`

Out[23]: [1, 1, 2, 3, 6, 3, 3, 8, 10, 10, 5, 1]

In [24]: *# ordenar los valores*
`L.sort()`
`L`

Out[24]: [1, 1, 1, 2, 3, 3, 3, 5, 6, 8, 10, 10]

In [25]: *# Eliminar valores duplicados y colocarlos*
`L = set(L)`
`L`

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

In [26]: `type(L)`

Out[26]: set

In [27]: `L[0] = 100`

```
-----  
-  
TypeError                                Traceback (most recent call las  
t)  
Cell In[27], line 1  
----> 1 L[0] = 100  
  
TypeError: 'set' object does not support item assignment
```

```
In [28]: L.append(50)
```

```
-----  
-  
AttributeError                          Traceback (most recent call las  
t)  
Cell In[28], line 1  
----> 1 L.append(50)  
  
AttributeError: 'set' object has no attribute 'append'
```

```
In [29]: L1 = list(L)  
L1
```

```
Out[29]: [1, 2, 3, 5, 6, 8, 10]
```

```
In [30]: L1[0] = 100  
L1
```

```
Out[30]: [100, 2, 3, 5, 6, 8, 10]
```

```
In [31]: L1.append(500)  
L1
```

```
Out[31]: [100, 2, 3, 5, 6, 8, 10, 500]
```

Otras cosas importantes

```
In [32]: listado = [3, 7, 5, 1]  
listado
```

```
Out[32]: [3, 7, 5, 1]
```

```
In [33]: min(listado)
```

```
Out[33]: 1
```

```
In [34]: max(listado)
```

```
Out[34]: 7
```

```
In [35]: # Recomendación listas:  
L = [1, 23, 26, 45, 48, 1, 3, 6, 8]  
L
```

```
Out[35]: [1, 23, 26, 45, 48, 1, 3, 6, 8]
```

```
In [36]: # separar en grupos de 3 en 3:  
L = [  
    1, 23, 26,  
    45, 48, 1,  
    3, 6, 8  
]  
L
```

```
Out[36]: [1, 23, 26, 45, 48, 1, 3, 6, 8]
```

```
In [37]: test = [1, 2, 3, 4, 5, 6]  
test
```

```
Out[37]: [1, 2, 3, 4, 5, 6]
```

```
In [38]: test.reverse()  
test
```

```
Out[38]: [6, 5, 4, 3, 2, 1]
```

```
In [39]: test1 = enumerate(test)  
test1
```

```
Out[39]: <enumerate at 0x7f160c0e4200>
```

```
In [40]: list(test1)
```

```
Out[40]: [(0, 6), (1, 5), (2, 4), (3, 3), (4, 2), (5, 1)]
```

Transformación de entre ellas (Array/Lista)

Numpy array a lista

```
In [41]: import numpy as np
```

```
In [42]: E = np.array([1, 3, 4, 25, 1, 6, 7])  
E
```

```
Out[42]: array([ 1,  3,  4, 25,  1,  6,  7])
```

```
In [43]: E1 = E.tolist()  
E1
```

```
Out[43]: [1, 3, 4, 25, 1, 6, 7]
```

Lista a numpy Array

```
In [44]: F = [2, 3, 6, 8, 4, 12, 25]  
F
```

```
Out[44]: [2, 3, 6, 8, 4, 12, 25]
```

```
In [45]: F1 = np.array(F)  
F1
```

```
Out[45]: array([ 2,  3,  6,  8,  4, 12, 25])
```

```
In [46]: print(F1)
```

```
[ 2  3  6  8  4 12 25]
```

```
In [47]: type(E1), type(F1)
```

```
Out[47]: (list, numpy.ndarray)
```

Matrices

Ejemplo 1: Teoría

```
In [48]: import numpy as np
```

```
In [49]: a = np.array([
           [10, 20, 30, 40],
           [50, 60, 70, 80],
           [90, 100, 110, 120]
         ])
a
```

```
Out[49]: array([[ 10,  20,  30,  40],
                [ 50,  60,  70,  80],
                [ 90, 100, 110, 120]])
```

Ejemplo 2: Suma de matrices

```
In [50]: # np.array([ [fila1], [fila2]])
matriz1 = np.array([[1, 2], [3, 4]])
matriz1
```

```
Out[50]: array([[1, 2],
                [3, 4]])
```

```
In [51]: matriz2 = np.array([[5, 6], [7, 8]])
matriz2
```

```
Out[51]: array([[5, 6],
                [7, 8]])
```

```
In [52]: matriz_suma = matriz1 + matriz2
matriz_suma
```

```
Out[52]: array([[ 6,  8],
                [10, 12]])
```

Ejemplo 3: Multiplicación y división de matrices

```
In [53]: # np.array([ [fila1], [fila2]])
matriz1 = np.array([[1, 2], [3, 4]])
matriz1
```

```
Out[53]: array([[1, 2],  
               [3, 4]])
```

```
In [54]: matriz1[1][0]
```

```
Out[54]: 3
```

```
In [55]: matriz2 = np.array([[5, 6], [7, 8]])  
matriz2
```

```
Out[55]: array([[5, 6],  
               [7, 8]])
```

```
In [56]: # Multiplicación  
matriz_multiplicacion = 2 * matriz1  
matriz_multiplicacion
```

```
Out[56]: array([[2, 4],  
               [6, 8]])
```

```
In [57]: # División  
matriz_division = matriz_multiplicacion / 2  
matriz_division
```

```
Out[57]: array([[1., 2.],  
               [3., 4.]])
```

```
In [58]: matriz_division = matriz2 / 2  
matriz_division
```

```
Out[58]: array([[2.5, 3. ],  
               [3.5, 4. ]])
```

Creado por:

Isabel Maniega