6_Más_sobre_Listas,tuplas,matrices

June 18, 2025

Creado por:

Isabel Maniega

1 Estructura de datos de almacenamiento

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

Incluye:

- Tuplas,
- Listas,
- Diccionarios,
- etc

1.1 Tuplas

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

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

[2]:
```
A[2]
```

[2]:
```
30
```

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

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

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

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

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

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

[6]:
```
D = (400,)
E = B + D
E
```

[6]:
```
(1, 2, 400)
```

[7]:
```
A = A + B
A
```

[7]:
```
(10, 20, 30, 40, 1, 2)
```

[8]:
```
# Eliminación de toda la tupla:

del A

A # NameError: name 'A' is not defined
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[8], line 5
      1 # Eliminación de toda la tupla:
      3 del A
----> 5 A # NameError: name 'A' is not defined

NameError: name 'A' is not defined
```

[9]:
```
# No se puede ni modificar
# No se puede apendizar
```

1.2 Arrays

[10]:
```
import numpy as np
```

[11]:
```
B = np.array([1, 2, 3, 4])
B
```

[11]:
```
array([1, 2, 3, 4])
```

[12]:
```
B[0]
```

[12]:
```
np.int64(1)
```

[13]:
```
B[0] = 20
B
```

[13]:
```
array([20,  2,  3,  4])
```

[14]:
```
B1 = np.append(B, 50)
B1
```

[14]:
```
array([20,  2,  3,  4, 50])
```

[15]:
```
# se puede modificar
# se pueder añadir/ apendizar
```

o concatenar…

[16]:
```
B2 = np.array([60])
B2
```

[16]:
```
array([60])
```

[17]:
```
# unir dos arrays
B3 = np.concatenate((B1, B2))
B3
```

[17]:
```
array([20,  2,  3,  4, 50, 60])
```

1.3 Listas

[18]:
```
C = list((1, 2, 3, 4))
C
```

[18]:
```
[1, 2, 3, 4]
```

[19]:
```
D = [0, 2, 3, 5, 7, 8]
D
```

[19]:
```
[0, 2, 3, 5, 7, 8]
```

[20]:
```
D[2]
```

[20]:
```
3
```

[21]:
```
D[2] = 100
D
```

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

[22]:
```
D
```

```
[22]: [0, 2, 100, 5, 7, 8]
```

```
[23]: # Eliminar por valor concreta
      D.remove(100)
      D
```

```
[23]: [0, 2, 5, 7, 8]
```

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

```
[24]: [0, 2, 7, 8]
```

```
[25]: # Otra opción más:
      del D[3]
      D
```

```
[25]: [0, 2, 7]
```

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

```
[26]: [1, 1, 2, 3, 6, 3, 3, 8, 10, 10, 5, 1]
```

```
[27]: # ordenar los valores
      L.sort()
      L
```

```
[27]: [1, 1, 1, 2, 3, 3, 3, 5, 6, 8, 10, 10]
```

```
[28]: # Eliminar valores duplicados y colocarlos
      L = set(L)
      L
```

```
[28]: {1, 2, 3, 5, 6, 8, 10}
```

```
[29]: type(L)
```

```
[29]: set
```

```
[30]: L[0] = 100
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[30], line 1
----> 1 L[0] = 100
```

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

[31]:

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

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[31], line 1
----> 1 L.append(50)

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

[32]:

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

[32]:

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

[33]:

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

[33]:

```
[100, 2, 3, 5, 6, 8, 10]
```

[34]:

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

[34]:

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

1.4 Otras cosas importantes

[35]:

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

[35]:

```
[3, 7, 5, 1]
```

[36]:

```
min(listado)
```

[36]:

```
1
```

[37]:

```
max(listado)
```

[37]:

```
7
```

[38]:

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

```
[38]: [1, 23, 26, 45, 48, 1, 3, 6, 8]
```

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

```
[39]: [1, 23, 26, 45, 48, 1, 3, 6, 8]
```

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

```
[40]: [1, 2, 3, 4, 5, 6]
```

```
[41]: test.reverse()
      test
```

```
[41]: [6, 5, 4, 3, 2, 1]
```

```
[42]: test1 = enumerate(test)
      test1
```

```
[42]: <enumerate at 0x76b1e0394310>
```

```
[43]: list(test1)
```

```
[43]: [(0, 6), (1, 5), (2, 4), (3, 3), (4, 2), (5, 1)]
```

1.5 Transformación de entre ellas (Array/Lista)

Numpy array a lista

```
[44]: import numpy as np
```

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

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

```
[46]: E1 = E.tolist()
      E1
```

```
[46]: [1, 3, 4, 25, 1, 6, 7]
```

Lista a numpy Array

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

```
[47]: [2, 3, 6, 8, 4, 12, 25]
```

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

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

```
[49]: print(F1)
```

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

```
[50]: type(E1), type(F1)
```

```
[50]: (list, numpy.ndarray)
```

1.6 Matrices

1.6.1 Ejemplo 1: Teoría

```
[51]: import numpy as np
```

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

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

1.6.2 Ejemplo 2: Suma de matrices

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

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

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

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

```
[55]: matriz_suma = matriz1 + matriz2
      matriz_suma
```

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

1.6.3 Ejemplo 3: Multiplicación y división de matrices

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

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

```
[57]: matriz1[1][0]
```

```
[57]: np.int64(3)
```

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

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

```
[59]: # Multiplicación
      matriz_multiplicacion = 2 * matriz1
      matriz_multiplicacion
```

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

```
[60]: # División
      matriz_division = matriz_multiplicacion / 2
      matriz_division
```

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

```
[61]: matriz_division = matriz2 / 2
      matriz_division
```

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

Creado por:

Isabel Maniega