

Creado por:

Isabel Maniega

Diccionarios

Los diccionarios se usan en:

- Machine learning
- Base de datos: NoSQL como es MongoDB
- en JSON, etc

Parte 1

```
In [1]: # clave-valor  
# "key": "value"  
# { "key": "value"  
# { "key": "value", "key2": "value2", "key3": "value3", ...}
```

```
In [2]: diccionario = {"A": 10, "B": 2, "C": 35}  
diccionario
```

```
Out[2]: {'A': 10, 'B': 2, 'C': 35}
```

```
In [3]: diccionario["A"]
```

```
Out[3]: 10
```

```
In [4]: diccionario["B"]
```

```
Out[4]: 2
```

```
In [5]: diccionario["C"]
```

```
Out[5]: 35
```

```
In [6]: len(diccionario)
```

```
Out[6]: 3
```

Parte 2

```
In [7]: diccionario1 = {"clave1": 1, "clave2": 2, "clave3": 3}  
diccionario1
```

```
Out[7]: {'clave1': 1, 'clave2': 2, 'clave3': 3}
```

```
In [8]: diccionario1.keys()
```

```
Out[8]: dict_keys(['clave1', 'clave2', 'clave3'])
```

```
In [9]: diccionario1.values()
```

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

```
In [10]: type(diccionario1.values())
```

```
Out[10]: dict_values
```

```
In [11]: # Obtener los valores de Keys/claves  
for key in diccionario1.keys():  
    print(key)
```

```
clave1  
clave2  
clave3
```

```
In [12]: # Obtener los valores de Keys/claves en forma de listado  
listado_keys = [key for key in diccionario1.keys()]  
listado_keys
```

```
Out[12]: ['clave1', 'clave2', 'clave3']
```

```
In [13]: # Obtener los valores de values/valores  
for value in diccionario1.values():  
    print(value)
```

```
1  
2  
3
```

```
In [14]: # Obtener listado  
listado_values = [value for value in diccionario1.values()]  
listado_values
```

```
Out[14]: [1, 2, 3]
```

```
In [15]: diccionario1.items()
```

```
Out[15]: dict_items([('clave1', 1), ('clave2', 2), ('clave3', 3)])
```

```
In [16]: # obtención de clave/valor mediante for:  
for key, value in diccionario1.items():  
    print("clave: ", key)  
    print("valor: ", value)
```

```
clave: clave1  
valor: 1  
clave: clave2  
valor: 2  
clave: clave3  
valor: 3
```

```
In [17]: diccionario1["clave1"]
```

```
Out[17]: 1
```

```
In [18]: # Modificación de valores en el diccionario  
diccionario1["clave1"] = 5  
diccionario1
```

```
Out[18]: {'clave1': 5, 'clave2': 2, 'clave3': 3}
```

```
In [19]: len(diccionario1)
```

```
Out[19]: 3
```

```
In [20]: # Eliminar un campo del diccionario  
del diccionario1["clave3"]  
diccionario1
```

```
Out[20]: {'clave1': 5, 'clave2': 2}
```

```
In [21]: len(diccionario1)
```

```
Out[21]: 2
```

```
In [22]: # otra forma de eliminar...  
diccionario1.pop("clave2")  
diccionario1
```

```
Out[22]: {'clave1': 5}
```

```
In [23]: len(diccionario1)
```

```
Out[23]: 1
```

```
In [24]: # Borrar todos los elementos del diccionario .clear()  
diccionario1.clear()
```

```
In [25]: diccionario1
```

```
Out[25]: {}
```

```
In [26]: len(diccionario1)
```

```
Out[26]: 0
```

Parte 3

```
In [27]: dic = {"clave1": 10, "clave2": 20, "clave3": 30}  
dic
```

```
Out[27]: {'clave1': 10, 'clave2': 20, 'clave3': 30}
```

```
In [28]: from collections import Counter
```

```
In [29]: Counter(dic)
```

```
Out[29]: Counter({'clave1': 10, 'clave2': 20, 'clave3': 30})
```

```
In [30]: Counter(dic).most_common()
```

```
Out[30]: [('clave3', 30), ('clave2', 20), ('clave1', 10)]
```

```
In [31]: Counter(dic).most_common()[0]
```

```
Out[31]: ('clave3', 30)
```

```
In [32]: Counter(dic).most_common()[1]
```

```
Out[32]: ('clave2', 20)
```

```
In [33]: Counter(dic).most_common()[2]
```

```
Out[33]: ('clave1', 10)
```

```
In [34]: Counter(dic).most_common()[-1]
```

```
Out[34]: ('clave1', 10)
```

```
In [35]: Counter(dic).most_common()[-2]
```

```
Out[35]: ('clave2', 20)
```

```
In [36]: Counter(dic).most_common()[-3]
```

```
Out[36]: ('clave3', 30)
```

```
In [37]: # Si queremos seleccionar unos elementos del diccionario,  
# en este caso los primeros valores  
Counter(dic).most_common()[:2]
```

```
Out[37]: [('clave3', 30), ('clave2', 20)]
```

```
In [38]: Counter(dic).most_common()[1:]
```

```
Out[38]: [('clave2', 20), ('clave1', 10)]
```

Parte 4

```
In [39]: diccionario2 = {"clave1": 10, "clave2": 20, "clave3": 30}  
diccionario2
```

```
Out[39]: {'clave1': 10, 'clave2': 20, 'clave3': 30}
```

```
In [40]: diccionario2.keys()
```

```
Out[40]: dict_keys(['clave1', 'clave2', 'clave3'])
```

```
In [41]: listado_keys = []  
for key in diccionario2.keys():  
    listado_keys.append(key)  
listado_keys
```

```
Out[41]: ['clave1', 'clave2', 'clave3']
```

```
In [42]: listado_values = []
        for value in diccionario2.values():
            listado_values.append(value)
        listado_values
```

```
Out[42]: [10, 20, 30]
```

```
In [43]: # pip install pandas
        import pandas as pd
```

```
In [44]: df_diccionario = pd.DataFrame(listado_keys, columns=["claves"])
        df_diccionario
```

```
Out[44]:
```

	claves
0	clave1
1	clave2
2	clave3

```
In [45]: df_diccionario["Valores"] = listado_values
        df_diccionario
```

```
Out[45]:
```

	claves	Valores
0	clave1	10
1	clave2	20
2	clave3	30

Parte 5

```
In [46]: # Como ordenar un dataframe
        df_diccionario.sort_values(by='Valores')
        # ascendente
```

```
Out[46]:
```

	claves	Valores
0	clave1	10
1	clave2	20
2	clave3	30

```
In [47]: # si no se especifica ascending (caso anterior) por defecto ascending = True
        df_diccionario.sort_values(by='Valores', ascending=True)
```

```
Out[47]:
```

	claves	Valores
0	clave1	10
1	clave2	20
2	clave3	30

```
In [48]: df_diccionario.sort_values(by='Valores', ascending=False)
# descendente
```

```
Out[48]:
```

	claves	Valores
2	clave3	30
1	clave2	20
0	clave1	10

-5.1- Strings

Importancia de los strings en AI

Strings como introducción al Procesamiento de Lenguaje Natural (NLP- Natural Language Processing)

Proyectos típicos de Inteligencia Artificial con NLP:

- Chatbots,
- analítica de textos,
- análisis de sentimientos en redes sociales,
- Etc.

Index en los Strings

```
In [49]: s1 = "Hi, How are you?"
s1
```

```
Out[49]: 'Hi, How are you?'
```

```
In [50]: s1[0]
```

```
Out[50]: 'H'
```

```
In [51]: s1[0], s1[1], s1[2]
```

```
Out[51]: ('H', 'i', ',')
```

```
In [52]: s1[-1]
```

```
Out[52]: '?'
```

Longitud

```
In [53]: len(s1)
```

```
Out[53]: 16
```

```
In [54]: for letra in s1:
```

```
print(letra)
```

```
H  
i  
,  
  
H  
o  
w  
  
a  
r  
e  
  
y  
o  
u  
?
```

most_common()

```
In [55]: s1 = "Hi, How are you?"  
s1
```

```
Out[55]: 'Hi, How are you?'
```

```
In [56]: from collections import Counter
```

```
In [57]: # Número de veces que esta en el strings  
Counter(s1).most_common()
```

```
Out[57]: [(' ', 3),  
          ('H', 2),  
          ('o', 2),  
          ('i', 1),  
          (',', 1),  
          ('w', 1),  
          ('a', 1),  
          ('r', 1),  
          ('e', 1),  
          ('y', 1),  
          ('u', 1),  
          ('?', 1)]
```

count

```
In [58]: # Frecuencia aparece la palabra  
s1 = "Hi, How are you?"  
s1
```

```
Out[58]: 'Hi, How are you?'
```

```
In [59]: s1.count("Hi")
```

```
Out[59]: 1
```

```
In [60]: s1.count("How")
```

```
Out[60]: 1
```

```
In [61]: # debe ser exacta para que la busque, si está en H pues debe buscarse así  
s1.count("hi")
```

```
Out[61]: 0
```

upper / lower

```
In [62]: s1 = "Hi, How are you?"  
s1
```

```
Out[62]: 'Hi, How are you?'
```

```
In [63]: mayusculas = s1.upper()  
mayusculas
```

```
Out[63]: 'HI, HOW ARE YOU?'
```

```
In [64]: minuscula = s1.lower()  
minuscula
```

```
Out[64]: 'hi, how are you?'
```

find

```
In [65]: s1 = "Hi, How are you?"  
s1
```

```
Out[65]: 'Hi, How are you?'
```

```
In [66]: len(s1)
```

```
Out[66]: 16
```

```
In [67]: # Buscar la letra "o"  
s1[5], s1[13]
```

```
Out[67]: ('o', 'o')
```

```
In [68]: # búsqueda de la posición  
s1_find = s1.find("o")  
s1_find
```

```
Out[68]: 5
```

```
In [69]: # búsqueda de la posición  
s1_interrogación = s1.find("?")  
s1_interrogación
```

```
Out[69]: 15
```



```
In [70]: s1[-1]
```

```
Out[70]: '?'
```

```
In [71]: # si no encuentra la letra, en ese caso pone -1  
s1_notfound = s1.find("p")  
s1_notfound
```

```
Out[71]: -1
```

```
In [72]: # Diferencia entre mayúsculas y minúsculas  
s1 = "Hi, how are you?"  
s1
```

```
Out[72]: 'Hi, how are you?'
```

```
In [73]: s1_H = s1.find("H")  
s1_H
```

```
Out[73]: 0
```

```
In [74]: s1_h = s1.find("h")  
s1_h
```

```
Out[74]: 4
```

startswith, endswith

```
In [75]: s1 = "Hi, how are you?"  
s1
```

```
Out[75]: 'Hi, how are you?'
```

```
In [76]: s1_startswith = s1.startswith("hi")  
s1_startswith
```

```
Out[76]: False
```

```
In [77]: s1_startswith = s1.startswith("Hi")  
s1_startswith
```

```
Out[77]: True
```

```
In [78]: s1_endswith = s1.endswith("you")  
s1_endswith
```

```
Out[78]: False
```

```
In [79]: s1_endswith = s1.endswith("you?")  
s1_endswith
```

```
Out[79]: True
```

Split

```
In [80]: # hacemos el split de un string (division en substrings)  
# creando una lista de elementos que componen el string  
s1 = "Hi, how are you?"  
s1
```

Out[80]: 'Hi, how are you?'

```
In [81]: s1_split = s1.split()  
s1_split
```

Out[81]: ['Hi,', 'how', 'are', 'you?']

```
In [82]: s1 = "Hi, how are you?"  
s1
```

Out[82]: 'Hi, how are you?'

```
In [83]: s1_split = s1.split(",")  
s1_split
```

Out[83]: ['Hi', ' how are you?']

```
In [84]: s2 = "Hi , how are you?"  
s2
```

Out[84]: 'Hi , how are you?'

```
In [85]: s2_split = s2.split()  
s2_split
```

Out[85]: ['Hi', ', ', 'how', 'are', 'you?']

Replace

```
In [86]: # reemplazar algo  
s1 = "Hi, How are you?"  
s1
```

Out[86]: 'Hi, How are you?'

```
In [87]: # sustituir la "H" por "h"  
# el primer valor es el valor a sustituir  
# el segundo valor el valor que quiero poner  
s1.replace("H", "h")
```

Out[87]: 'hi, how are you?'

Join

```
In [ ]: # une todos los elementos del string por un simbolo específico  
# "-" en este caso
```

```
In [88]: s1 = "Hi, How are you?"  
s1
```

```
Out[88]: 'Hi, How are you?'
```

```
In [89]: s1_join = "-".join(s1)  
s1_join
```

```
Out[89]: 'H-i-,- -H-o-w- -a-r-e- -y-o-u-?'
```

```
In [90]: s1_join = "+".join(s1)  
s1_join
```

```
Out[90]: 'H+i+,+ +H+o+w+ +a+r+e+ +y+o+u+?'
```

Sleep y time

```
In [91]: from time import sleep
```

```
In [93]: %%time  
print("Hola")  
sleep(2)  
print("Mundo")
```

Hola

Mundo

CPU times: user 1.89 ms, sys: 0 ns, total: 1.89 ms

Wall time: 2 s

```
In [94]: %%time  
print("Hola")  
sleep(10)  
print("Mundo")
```

Hola

Mundo

CPU times: user 3.42 ms, sys: 0 ns, total: 3.42 ms

Wall time: 10 s

```
In [95]: import time
```

```
In [96]: %%time  
print("Hola")  
time.sleep(2)  
print("Mundo")
```

Hola

Mundo

CPU times: user 2.26 ms, sys: 0 ns, total: 2.26 ms

Wall time: 2 s

```
In [ ]: # Ejemplo 1
```

```
In [97]: %%time  
x = 2  
a = time.time()  
print(x)
```

```
b = time.time()

tiempo = a - b
tiempo
```

2

CPU times: user 38 µs, sys: 0 ns, total: 38 µs

Wall time: 40.1 µs

Out[97]: -3.0517578125e-05

```
In [98]: %%time
print("Hello")
a = time.time()
time.sleep(5)
b = time.time()
print("World")

tiempo = a - b
tiempo
```

Hello

World

CPU times: user 1.81 ms, sys: 226 µs, total: 2.04 ms

Wall time: 5.01 s

Out[98]: -5.005258560180664

In []: # Ejemplo 2

```
In [99]: %%time

import time

tiempo_inicial = time.time()

contador = 0

for numero in range(1000000):
    contador += 1

tiempo_final = time.time()

tiempo_ejecucion = tiempo_final - tiempo_inicial
print("tiempo de ejecución: t_final - t.inicial = ", tiempo_ejecucion)
```

tiempo de ejecución: t_final - t.inicial = 0.036331892013549805

CPU times: user 36.7 ms, sys: 0 ns, total: 36.7 ms

Wall time: 36.6 ms

Operaciones Elementales y algunas cosas más

In [100... 5<7

Out[100... True

In [101... 5<7, 5<5, 5<=5, 7>5, 7>=5, 5>5

Out[101... (True, False, True, True, True, False)

```
In [102... # Ojo con estos, porque van en condicionales if para testear  
6!=5, 6!=6, 5==5, 5==6
```

```
Out[102... (True, False, True, False)
```

```
In [103... # division:  
12/4
```

```
Out[103... 3.0
```

```
In [104... # cociente:  
11//4
```

```
Out[104... 2
```

```
In [105... # resto:  
11 % 4
```

```
Out[105... 3
```

```
In [106... print('División exacta: ', 12/4, "cociente:", 11//4, "resto", 11%4)
```

```
División exacta: 3.0 cociente: 2 resto 3
```

```
In [107... resultado = 12/4 + 11//4 + 11%4  
resultado
```

```
Out[107... 8.0
```

```
In [108... # Multiplicacion  
2 * 4
```

```
Out[108... 8
```

```
In [109... # Elevado:  
2**4
```

```
Out[109... 16
```

Creado por:

Isabel Maniega