

Creado por:
Isabel Maniega

-5- 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", "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 [8]: diccionario1 = {"clave1": 1, "clave2": 2, "clave3": 3}
diccionario1

Out[8]: {'clave1': 1, 'clave2': 2, 'clave3': 3}

In [10]: diccionario1.keys()

Out[10]: dict_keys(['clave1', 'clave2', 'clave3'])

In [11]: diccionario1.values()

Out[11]: dict_values([1, 2, 3])

In [12]: type(diccionario1.values())

Out[12]: dict_values

In [13]: # Obtener los valores de Keys/Claves
for key in diccionario1.keys():
    print(key)

clave1
clave2
clave3

In [14]: # Obtener los valores de Keys/Claves en forma de listado
listado_keys = [key for key in diccionario1.keys()]
listado_keys

Out[14]: ['clave1', 'clave2', 'clave3']

In [15]: # Obtener los valores de valores/valores
for value in diccionario1.values():
    print(value)

1
2
3

In [16]: # Obtener listado
listado_valores = [value for value in diccionario1.values()]
listado_valores

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

In [17]: diccionario1.items()

Out[17]: dict_items([('clave1', 1), ('clave2', 2), ('clave3', 3)])

In [18]: # 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 [19]: diccionario1["clave1"]

Out[19]: 1

In [21]: # Modificación de valores en el diccionario
diccionario1["clave1"] = 5
diccionario1

Out[21]: {'clave1': 5, 'clave2': 2, 'clave3': 3}

In [22]: len(diccionario1)

Out[22]: 3

In [23]: # Eliminar un campo del diccionario
del diccionario1["clave3"]
diccionario1

Out[23]: {'clave1': 5, 'clave2': 2}

In [24]: len(diccionario1)

Out[24]: 2

In [25]: # otra forma de eliminar...
diccionario1.pop("clave2")
diccionario1

Out[25]: {'clave1': 5}

In [26]: len(diccionario1)

Out[26]: 1

In [27]: # Borrar todos los elementos del diccionario .clear()
diccionario1.clear()

In [28]: diccionario1

Out[28]: {}

In [29]: len(diccionario1)

Out[29]: 0
```

Parte 3

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

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

In [31]: from collections import Counter

In [32]: Counter(dic)

Out[32]: Counter({'clave1': 10, 'clave2': 20, 'clave3': 30})

In [36]: Counter(dic).most_common()

Out[36]: [('clave3', 30), ('clave2', 20), ('clave1', 10)]

In [34]: Counter(dic).most_common()[0]

Out[34]: ('clave3', 30)

In [35]: Counter(dic).most_common()[1]

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

In [37]: Counter(dic).most_common()[2]

Out[37]: ('clave1', 10)

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

Out[38]: ('clave1', 10)

In [39]: Counter(dic).most_common()[-2]

Out[39]: ('clave2', 20)

In [40]: Counter(dic).most_common()[-3]

Out[40]: ('clave3', 30)

In [41]: # Si queremos seleccionar unos elementos del diccionario,
# en este caso los primeros valores
Counter(dic).most_common()[1:2]

Out[41]: [('clave3', 30), ('clave2', 20)]

In [42]: Counter(dic).most_common()[1:]

Out[42]: [('clave2', 20), ('clave1', 10)]
```

Parte 4

```
In [57]: diccionario2 = {"clave1": 10, "clave2": 20, "clave3": 30}
diccionario2

Out[57]: {'clave1': 10, 'clave2': 20, 'clave3': 30}

In [58]: diccionario2.keys()

Out[58]: dict_keys(['clave1', 'clave2', 'clave3'])

In [59]: listado_keys = []
for key in diccionario2.keys():
    listado_keys.append(key)
listado_keys

Out[59]: ['clave1', 'clave2', 'clave3']

In [60]: listado_valores = []
for value in diccionario2.values():
    listado_valores.append(value)
listado_valores

Out[60]: [10, 20, 30]

In [61]: # pip install pandas
import pandas as pd

In [62]: df_diccionario = pd.DataFrame(listado_keys, columns=["claves"])
df_diccionario

Out[62]:
  claves
0  clave1
1  clave2
2  clave3

In [63]: df_diccionario["Valores"] = listado_valores
df_diccionario

Out[63]:
  claves  Valores
0  clave1      10
1  clave2      20
2  clave3      30
```

Parte 5

```
In [65]: # Como ordenar un dataframe
df_diccionario.sort_values(by="Valores")
# ascendente

Out[65]:
  claves  Valores
0  clave1      10
1  clave2      20
2  clave3      30

In [68]: # si no se especifica ascending (caso anterior) por defecto ascending = True
df_diccionario.sort_values(by="Valores", ascending=True)

Out[68]:
  claves  Valores
2  clave3      30
1  clave2      20
0  clave1      10

In [67]: df_diccionario.sort_values(by="Valores", ascending=False)
# descendente

Out[67]:
  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 [69]: s1 = "Hi, How are you?"
s1

Out[69]: 'Hi, How are you?'

In [70]: s1[0]

Out[70]: 'H'

In [71]: s1[0], s1[1], s1[2]

Out[71]: ('H', 'i', ',')
```

```
In [72]: s1[-1]

Out[72]: '?'

Longitud
```

```
In [73]: len(s1)

Out[73]: 16

In [74]: for letra in s1:
    print(letra)

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

most_common()

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

Out[75]: 'Hi, How are you?'

In [76]: from collections import Counter

Out[77]: # Número de veces que está en el strings
Counter(s1).most_common()

Out[77]:
[(' ', 3),
 ('H', 2),
 ('o', 2),
 ('i', 1),
 ('r', 1),
 ('w', 1),
 ('a', 1),
 ('r', 1),
 ('e', 1),
 ('y', 1),
 ('u', 1),
 ('?', 1)]
```

count

```
In [78]: # Frecuencia aparece la palabra
s1 = "Hi, How are you?"
s1

Out[78]: 'Hi, How are you?'

In [79]: s1.count("Hi")

Out[79]: 1

In [80]: s1.count("How")

Out[80]: 1

In [81]: # debe ser exacta para que la busque, si está en H pues debe buscarse así
s1.count("Hi")

Out[81]: 0
```

upper / lower

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

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

In [83]: mayusculas = s1.upper()
mayusculas

Out[83]: 'HI, HOW ARE YOU?'

In [84]: minuscula = s1.lower()
minuscula

Out[84]: 'hi, how are you?'
```

find

```
In [85]: s1 = "Hi, How are you?"
s1

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

In [86]: len(s1)

Out[86]: 16

In [87]: # Buscar la letra "o"
s1[5], s1[13]

Out[87]: ('o', 'o')

In [88]: # búsqueda de la posición
s1.find("o")
s1.find

Out[88]: 5

In [89]: # búsqueda de la posición
s1.interrogación = s1.find("?")
s1.interrogación

Out[89]: 15

In [90]: s1[-1]

Out[90]: '?'

In [91]: # si no encuentra la letra, en ese caso pone -1
s1_notfound = s1.find("p")
s1_notfound

Out[91]: -1

In [92]: # Diferencia entre mayúsculas y minúsculas
s1 = "Hi, how are you?"
s1

Out[92]: 'Hi, how are you?'

In [93]: s1_H = s1.find("H")
s1_H

Out[93]: 0

In [94]: s1_h = s1.find("h")
s1_h

Out[94]: 4
```

startswith, endswith

```
In [95]: s1 = "Hi, how are you?"
s1

Out[95]: 'Hi, how are you?'

In [96]: s1.startswith("hi")

Out[96]: False

In [97]: s1.startswith("Hi")
s1.startswith

Out[97]: True

In [98]: s1.endswith("you")
s1.endswith

Out[98]: False

In [99]: s1.endswith("you?")
s1.endswith

Out[99]: True
```

Split

```
In [100]: # 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[100]: 'Hi, how are you?'

In [101]: s1_split = s1.split()
s1_split

Out[101]: ['Hi,', 'how', 'are', 'you?']

In [102]: s1 = "Hi, how are you?"
s1

Out[102]: 'Hi, how are you?'

In [103]: s1_split = s1.split(",")
s1_split

Out[103]: ['Hi', ' how are you?']

In [104]: s2 = "Hi, how are you?"
s2

Out[104]: 'Hi , how are you?'

In [106]: s2_split = s2.split()
s2_split

Out[106]: ['Hi', ' ', 'how', 'are', 'you?']
```

Replace

```
In [109]: # reemplazar algo
s1 = "Hi, How are you?"
s1

Out[109]: 'Hi, How are you?'

In [110]: # sustituir la "H" por "h"
# el primer valor es el valor a sustituir
# el segundo valor el valor que quiera poner
s1.replace("H", "h")

Out[110]: 'hi, how are you?'

Join
```

```
In [111]: # une todos los elementos del string por un símbolo específico
# "-" en este caso

In [112]: s1 = "Hi, How are you?"
s1

Out[112]: 'Hi, How are you?'

In [113]: s1_join = "-".join(s1)
s1_join

Out[113]: 'H-i,- -H-o-w- -a-r-e- -y-o-u-?'

In [114]: s1_join = "+".join(s1)
s1_join

Out[114]: 'Hi,r,+ +H+o+w+ +a+r+e+ +y+o+u+?'
```

Sleep y time

```
In [115]: from time import sleep

In [117]: %time
print("Hola")
sleep(2)
print("Mundo")

Hola
Mundo
CPU times: user 4.4 ms, sys: 0 ns, total: 4.4 ms
Wall time: 2 s

In [118]: %time
print("Hola")
sleep(10)
print("Mundo")

Hola
Mundo
CPU times: user 849 µs, sys: 3.62 ms, total: 4.47 ms
Wall time: 10 s

In [119]: import time

In [120]: %time
print("Hola")
time.sleep(2)
print("Mundo")

Hola
Mundo
CPU times: user 3.4 ms, sys: 865 µs, total: 4.26 ms
Wall time: 2 s

In [121]: # Ejemplo 1

In [122]: %time
x = 2
a = time.time()
print(x)
b = time.time()

tiempo = a - b
tiempo

CPU times: user 422 µs, sys: 10 µs, total: 432 µs
Wall time: 371 µs

Out[122]: -0.000315264315185547

In [123]: %time
print("Hello")
a = time.time()
time.sleep(5)
b = time.time()
print("World")

tiempo = a - b
tiempo

Hello
World
CPU times: user 4.44 ms, sys: 0 ns, total: 4.44 ms
Wall time: 5 s

Out[123]: -5.001991510391235

In [124]: # Ejemplo 2

In [126]: %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 = 0.24107742309570312
CPU times: user 199 ms, sys: 0 ns, total: 199 ms
Wall time: 242 ms
```

Operaciones Elementales y algunas cosas más

```
In [129]: 5<7

Out[129]: True

In [128]: 5<7, 5<=5, 7>5, 7==5, 5==5

Out[128]: (True, False, True, True, True, False)

In [130]: # Ojo con estos, porque van en condicionales if para testear
6!=5, 6!=6, 5==5, 5==6

Out[130]: (True, False, True, False)

In [131]: # division:
12/4

Out[131]: 3.0

In [132]: # cociente:
11/4

Out[132]: 2.75

In [133]: # resto:
11%4

Out[133]: 3

In [134]: print('División exacta: ', 12/4, "cociente:", 11//4, "resto", 11%4)

División exacta: 3.0 cociente: 2 resto 3

In [135]: # Multiplicacion
2 * 4

Out[135]: 8

In [136]: # Elevado:
2**4

Out[136]: 16
```

Creado por:
Isabel Maniega