

```
In [31]: from IPython.display import Image
```

DICCIONARIOS EN PYTHON

- Los diccionarios son otro tipo de estructura o colección al igual que las cadenas y las listas.
- De igual forma pueden contener más de un valor
- Sin embargo los elementos se guardan sin ningún orden.
- Los diccionario están compuestos por una estructura para guardar valores de la forma CLAVE:VALOR.
- Cada clave está separada de su valor por dos puntos ":",
- Sus elementos separados por comas y
- Todo va delimitado con llaves {}

Ejemplos

Diccionario de edades con el nombre como clave

```
In [28]: edades = {"Juan":23, "Maria": 25, "Ana":22, "Pablo":25}
```

```
In [29]: dicGenes = {"genRaton": "AACGTA", "genGato" : "GTACCCA", "genHumano": "CCGGTAACAGGG"}
```

Característica de los Diccionarios

- Tipo de datos: **dict** (Las cadenas son: **str** y las lista son **list**)
- Son mutables: se pueden cambiar el valor
- Para la creación se usa el símbolo de paréntesis **{ }**
- Para el acceso se utilizan los corchetes **[]**
- Se indexan a través de una llave **[key]**
- Diferente a las otras estructura que se indexan por posición (cadenas y listas)

Creación

```
In [ ]: ### Un diccionario vacio
d = {}
```

```
In [ ]: ### Un diccionario de notas
dicNotas = {"examen1": 3.0, "examen2": 4.3, "talleres":2.0, "quizes": 4.5}
```

```
In [ ]: ### Un diccionario de secuencias de genes
dicGenes = {"genRaton": "AACGTA", "genGato" : "GTACCCA", "genHumano": "CCGGTAACAGGG"}
```

Acceso a los elementos

- Se usa el nombre seguido de los corchetes y dentro de ellos la clave:

```
nombreDic [nombreClave]
```

Ejemplo acceso a los elementos

```
In [3]: edades = {"Juan":23, "Maria": 25, "Ana":22, "Pablo":25}
ej = edades ["Juan"]
print (ej)
```

23

```
In [4]: ep = edades ["Pablo"]
print (ep)
```

25

```
In [5]: print (edades ["Maria"])
```

25

Adiccion/Eliminación de Elementos

- Para adicionar un nuevo par elemento:clave se usa el operador **[]**
- Para eliminar una clave se utiliza la instruccion **del**

```
In [21]: # Adicion de la clave "Antonieta" con valor de edad de 55
edades = {"Juan":23, "Maria": 25, "Ana":22, "Pablo":25}
edades ["Antonieta"] = 55
print (edades)
```

```
{'Juan': 23, 'Maria': 25, 'Ana': 22, 'Pablo': 25, 'Antonieta': 55}
```

```
In [22]: # Eliminación de la clave y valor de "Ana"
del edades ["Ana"]
print (edades)
```

```
{'Juan': 23, 'Maria': 25, 'Pablo': 25, 'Antonieta': 55}
```

Métodos principales Integradas de Diccionarios

- **keys ()** : retorna las llaves
- **values ()**: retorna los valores
- **items ()** : retorna una pareja (llave,valor)

Otras métodos de Diccionariso

- 'clear',
- 'copy',
- 'fromkeys',
- 'get',
- 'items',
- 'keys',
- 'pop',
- 'popitem',
- 'setdefault',
- 'update',
- 'values'

Iteración en los diccionarios

Iteración general por valores

```
In [3]: edades = {"Juan":23, "Maria": 25, "Ana":22, "Pablo":25}
for k in edades:
    print (k)
```

```
Juan
Maria
Ana
Pablo
```

Iteración por llaves usando el método keys ()

```
In [7]: edades = {"Juan":23, "Maria": 25, "Ana":22, "Pablo":25}
for k in edades.keys():
    print ("Llave:", k)
    print ("Valor: ", edades [k])
```

```
Llave: Juan
Valor:  23
Llave: Maria
Valor:  25
Llave: Ana
Valor:  22
Llave: Pablo
Valor:  25
```

Iteración por valores usando el método values ()

```
In [9]: edades = {"Juan":23, "Maria": 25, "Ana":22, "Pablo":25}
for v in edades.values ():
    print (v)
```

```
23
25
22
25
```

Iteración por pareja (llave,valor) usando el método items()

```
In [27]: edades = {"Juan":23, "Maria":25, "Ana":22, "Pablo":25}
for i in edades.items ():
    print (i)
```

```
('Juan', 23)
('Maria', 25)
('Ana', 22)
('Pablo', 25)
```