

PROGRAMACION I

CLASE 11 – DICCIONARIOS

PROF. ING. VERÓNICA GALATI

DICCIONARIOS

Son estructuras de datos para relacionar
Clave - Valor

También se los conoce como arreglos asociativos o
tablas de Hash



DICCIONARIOS

No son secuencias, por lo tanto no están ordenados.

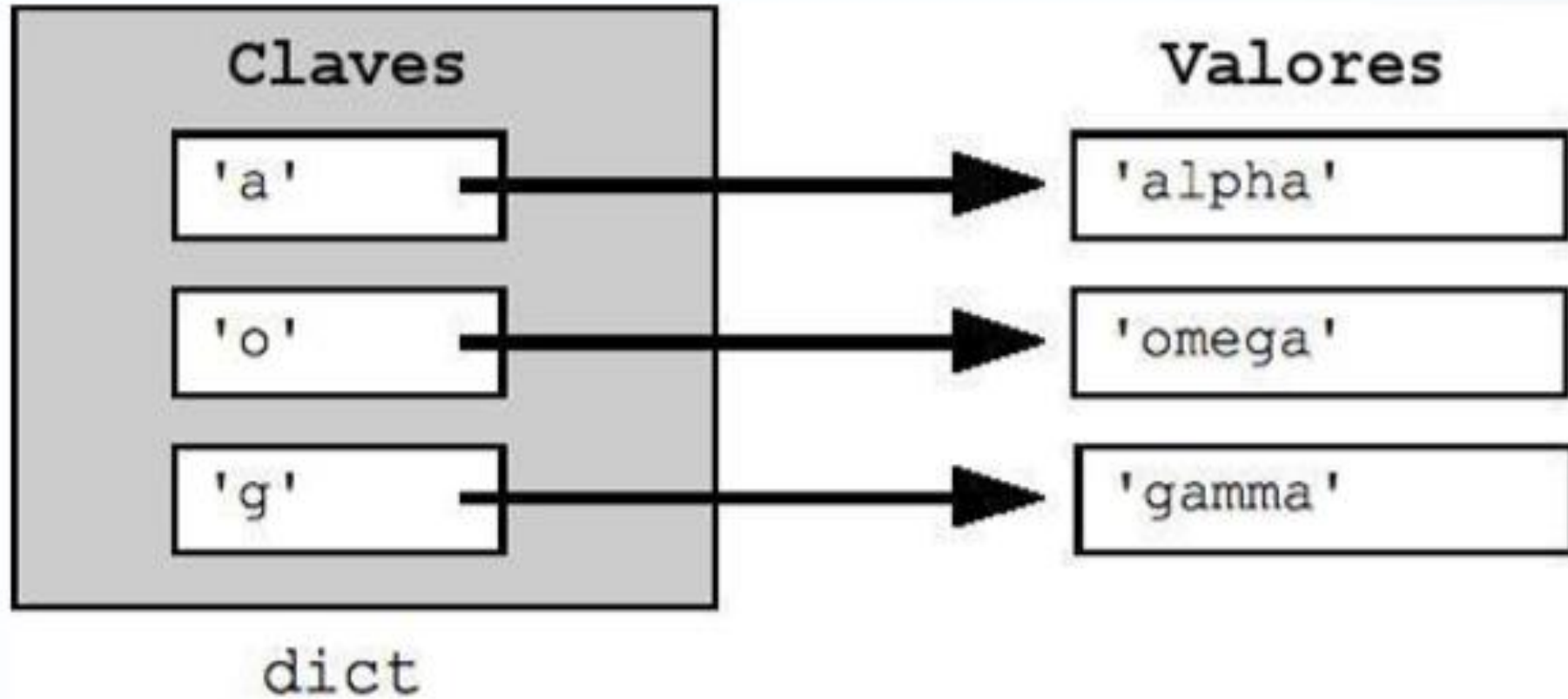
No se puede utilizar índice para acceder a elementos

Las **rebanadas** No son aplicables a los diccionarios
(carecen de orden interno)

Los elementos se acceden mediante la clave.



DICCIONARIOS



DICCIONARIOS

Cada elemento de un diccionario se representa mediante una **dupla** clave-valor

Se crean encerrando sus duplas entre llaves y separando por comas

```
edades={"Juan":23, "Maria":18, "Marcelo":30}
```

DICCIONARIOS

Las **claves** deben pertenecer a un tipo **inmutable**
(números, cadenas, tuplas)

Los **valores** asociados a cada clave pueden ser de
cualquier tipo incluso listas u otro diccionario

```
colores={"rojo":[255,0,0],"verde":[0,255,0],"azul":[0,0,255]}
```

DICCIONARIOS

Los **diccionarios** inicializados, suelen escribirse con un formato más claro y legible, colocando una dupla debajo de la otra

```
colores={ "rojo" : [255,0,0],  
          "verde": [0,255,0],  
          "azul"  : [0,0,255]  
        }
```

DICCIONARIOS

Acceder a un **valor** de una clave

```
edades={"Juan":23, "Maria":18, "Marcelo":30, "Pedro":23}  
print(edades["Marcelo"])
```



30

DICCIONARIOS

Un mismo **valor** puede estar asociado
a más de una **clave**

```
edades={"Juan":23, "Maria":18, "Marcelo":30, "Pedro":23}
```



```
{'Juan': 23, 'Maria': 18, 'Marcelo': 30, 'Pedro': 23}
```

DICCIONARIOS

Las **claves** deben ser únicas
(No admite claves duplicadas)

```
edades={"Juan":23, "María":18, "Marcelo":30, "Pedro":23, "María":25}
```



```
{'Juan': 23, 'Maria': 25, 'Marcelo': 30, 'Pedro': 23}
```

DICCIONARIOS

Asignar un **valor** a una **clave** reemplaza el valor existente o crea una clave nueva, dependiendo si existe o no previamente la clave.

```
colores={"rojo" : [255,0,0],  
        "verde": [0,255,0],  
        "azul"  : [0,0,255]  
        }  
  
colores["gris"]=[128,128,128]
```

DICCIONARIOS

No es posible acceder a una **clave** a través de su **valor**

Tratar de acceder a un elemento con una clave
inexistente provoca una excepción **KeyError**

```
colores={"rojo" : [255,0,0],  
         "verde": [0,255,0],  
         "azul"  : [0,0,255]  
        }  
  
print(colores["amarillo"])
```

```
print(colores["amarillo"])  
KeyError: 'amarillo'
```

DICCIONARIOS

Se puede evaluar si una clave existe con el operador **in** (o con el operador **not in**)

```
if "amarillo" in colores:  
    print(colores["amarillo"])  
else:  
    print("No existe amarillo")
```

DICCIONARIOS - MÉTODOS

Se puede utilizar el método **get(<clave>)**

Retorna el valor asociado a la clave

None si no existe la clave

```
colores={"rojo" : [255,0,0],  
        "verde": [0,255,0],  
        "azul"  : [0,0,255]  
        }
```

```
print(colores.get("rojo"))
```

```
[255, 0, 0]
```

DICCIONARIOS - MÉTODOS

El método **get(<clave>, valor)**

Admite un segundo parámetro que será devuelto si
No existe la clave

```
colores={"rojo" : [255,0,0],  
        "verde": [0,255,0],  
        "azul"  : [0,0,255]  
        }  
  
print(colores.get("amarillo"))
```

None

```
colores={"rojo" : [255,0,0],  
        "verde": [0,255,0],  
        "azul"  : [0,0,255]  
        }  
  
print(colores.get("amarillo","no existe"))
```

no existe

DICCIONARIOS - MÉTODOS

El método `items(<diccionario>)`

Retorna una lista de tuplas con clave,valor

```
colores={"rojo" : [255,0,0],  
        "verde": [0,255,0],  
        "azul"  : [0,0,255]  
        }  
for color,rgb in colores.items():  
    print(color,"->",rgb)
```

```
rojo -> [255, 0, 0]  
verde -> [0, 255, 0]  
azul -> [0, 0, 255]
```


DICCIONARIOS

Podemos recorrer un diccionario:

```
for clave in diccionario:  
    print("clave:",clave,"valor:",diccionario[clave])
```

```
for color in colores:  
    print(color,"->",colores[color])
```

```
rojo -> [255, 0, 0]  
verde -> [0, 255, 0]  
azul -> [0, 0, 255]
```

DICCIONARIOS - MÉTODOS

El método **keys(<diccionario>)**

Retorna una lista de todas las claves (iterable) y actúa como un conjunto

```
claves=colores.keys()  
print(claves)
```

```
dict_keys(['rojo', 'verde', 'azul'])
```

DICCIONARIOS - MÉTODOS

El método **values(<diccionario>)**

Retorna una lista con todos los valores

```
valores=edades.values()  
print(valores)
```

```
dict_values([23, 25, 30, 23])
```

DICCIONARIOS

Para eliminar un elemento se puede utilizar la instrucción **del**

```
colores={"rojo" : [255,0,0],  
        "verde": [0,255,0],  
        "azul"  : [0,0,255]  
        }  
del colores["rojo"]  
print(colores)
```

```
{'verde': [0, 255, 0], 'azul': [0, 0, 255]}
```

Retorna excepción **KeyError**
si no existe la clave

DICCIONARIOS

La instrucción **del**
Permite eliminar Todo el diccionario

```
colores={"rojo" : [255,0,0],  
         "verde": [0,255,0],  
         "azul"  : [0,0,255]  
        }
```

```
del colores
```

```
print(colores)  NameError: name 'colores' is not defined
```

DICCIONARIOS

El método **clear()**

Permite vaciar el contenido del diccionario

```
colores={"rojo" : [255,0,0],  
        "verde" : [0,255,0],  
        "azul"  : [0,0,255]  
}  
colores.clear()  
print(colores)
```

```
{ }
```

DICCIONARIOS

Se puede utilizar la función **len(<dic>)**

```
colores={"rojo" : [255,0,0],  
         "verde": [0,255,0],  
         "azul"  : [0,0,255]  
        }
```

```
print(len(colores))
```

3

DICCIONARIOS

Iterar por un diccionario se obtienen las claves.



Convertir un diccionario a una lista, se obtienen todas las claves en una nueva lista

```
colores={"rojo" : [255,0,0],  
         "verde": [0,255,0],  
         "azul"  : [0,0,255]  
        }  
claves= list(colores)  
print(claves)
```

```
['rojo', 'verde', 'azul']
```


DICCIONARIOS - EJEMPLO

Realizar un programa para ingresar una frase y mostrar un listado ordenado alfabéticamente con las palabras que contiene, eliminando las repetidas y añadiendo junto a cada una la cantidad de veces que se encontró.



DICCIONARIOS - EJEMPLO

```
frase=input("Ingrese una frase:")
palabras=frase.split()
dic={}
for palabra in palabras:
    if palabra not in dic:
        dic[palabra] = 1
    else:
        dic[palabra] = dic[palabra] + 1

claves=list(dic)
claves.sort()
for clave in claves:
    print(clave,"aparece :",dic[clave],"veces")
```

CONSULTAS???



Práctica 8
completa