

Diccionarios y Conjuntos

Programación

<http://progra.usm.cl>

UTFSM



Departamento de Informática
Universidad Técnica Federico Santa María

Diccionarios

Conceptos

Llave

Valor

```
>>> telefonos = {  
...     'Pepito': 5552437,  
...     'Jaimito': 5551428,  
...     'Yayita': 5550012,  
... }  
>>>  
>>> telefonos['Jaimito']  
5551428
```

Agregar llaves y valores

```
>>> d = {}
>>> d['a'] = 5
>>> d['x'] = 12
>>> d['m'] = 8
>>> d
{'a': 5, 'm': 8, 'x': 12}
>>> d['f'] = 5
>>> d['x'] = -10
>>> d
{'a': 5, 'f': 5, 'm': 8, 'x': -10}
```

Operaciones sobre diccionarios

```
>>> patas = {'humano': 2, 'pulpo': 8,
...          'perro': 4, 'gato': 4, 'ciempies': 100}
>>> patas.values()
[2, 8, 4, 4, 100]
>>> patas.items()
[('humano', 2), ('pulpo', 8), ('gato', 4),
 ('perro', 4), ('ciempies', 100)]
>>> patas.keys()
['humano', 'pulpo', 'gato', 'perro', 'ciempies']
>>> list(patas)
['humano', 'pulpo', 'gato', 'perro', 'ciempies']
>>> 'perro' in patas
True
>>> 8 in patas
False
>>> len(patas)
5
```

Ejercicio: contar letras

Escriba una función `contar_letras` (`palabra`) que reciba un string y retorne un diccionario que indique cuántas veces aparece cada letra en el string:

```
>>> contar_letras('entretener')
{'e': 4, 'n': 2, 'r': 2, 't': 2}
>>> contar_letras('lapiz')
{'a': 1, 'i': 1, 'l': 1, 'p': 1, 'z': 1}
```

Solución

```
def contar_letras(palabra):  
    cuentas = {}  
    for letra in palabra:  
        if letra not in cuentas:  
            cuentas[letra] = 0  
        cuentas[letra] = cuentas[letra] + 1  
    return cuentas
```

Recorrer diccionarios

```
capitales = {  
    'Chile': 'Santiago',  
    'Peru': 'Lima',  
    'Ecuador': 'Quito',  
}  
  
for pais in capitales:  
    print 'La capital de', pais, 'es', capitales[pais]  
  
for capital in capitales.values():  
    print capital, 'es una linda ciudad'  
  
for p, c in capitales.items():  
    print c, 'es la capital de', p
```

Conjuntos

```
>>> colores = {'rojo', 'gris', 'azul', 'gris'}
>>> colores
set(['azul', 'gris', 'rojo'])

>>> set('abracadabra')
set(['a', 'b', 'c', 'd', 'r'])
```


Agregar y sacar cosas de conjuntos

```
>>> a = set()
>>> a.add(19)
>>> a.add(12)
>>> a.add(8)
>>> a.add(12)
>>> a
set([8, 12, 19])
>>> a.remove(19)
>>> a
set([8, 12])
>>> 12 in a
True
```

Operaciones sobre conjuntos

```
>>> a = {1, 2, 3, 4}
>>> b = {2, 4, 6, 8}
>>> a & b
set ([2, 4])
>>> a | b
set ([1, 2, 3, 4, 6, 8])
>>> b - a
set ([6, 8])
>>> {1, 3} < {1, 2, 3}
True
```

Ejercicio: letras en común

Escriba una función `letras_en_comun(p, r)` que retorne la cantidad de letras en común que tienen las palabras `p` y `r`:

```
>>> letras_en_comun('murcielago', 'gominola')
6
>>> letras_en_comun('protozoo', 'agasajada')
0
```

Solución

```
def letras_en_comun(p, r):  
    return len(set(p) & set(r))
```