

Diccionarios

October 24, 2020

1 Diccionarios

Los diccionarios se encuentran a veces en otros lenguajes como «memorias asociativas» o «arreglos asociativos». A diferencia de las secuencias, que se indexan mediante un rango numérico, los diccionarios se indexan con claves, que pueden ser cualquier tipo inmutable. Las tuplas pueden usarse como claves si solamente contienen cadenas, números o tuplas; si una tupla contiene cualquier objeto mutable directa o indirectamente, no puede usarse como clave. Se pueden usar listas como claves, ya que las listas pueden modificarse usando asignación por índice, asignación por sección, o métodos como `append()` y `extend()`.

Es un conjunto de pares clave:valor con el requerimiento de que las claves sean únicas (dentro de un diccionario). Un par de llaves crea un diccionario vacío: `{}`. Colocar una lista de pares clave:valor separada por comas dentro de las llaves agrega, de inicio, pares clave:valor al diccionario; esta es, también, la forma en que los diccionarios se muestran en la salida.

```
[5]: # Espacio para mostrar  
  
variable = {"key1": 12, "key2":{"key1":23}, "key1": 'chau'}
```

```
[6]: variable
```

```
[6]: {'key1': 'chau', 'key2': {'key1': 23}}
```

```
[8]: variable['key2']
```

```
[8]: {'key1': 23}
```

```
[9]: variable['hola'] = [1,2,3,"hola"]
```

```
[10]: variable
```

```
[10]: {'key1': 'chau', 'key2': {'key1': 23}, 'hola': [1, 2, 3, 'hola']}
```

```
[11]: variable[123] = (1,8,8)
```

```
[14]: variable
```

```
[14]: {'key1': 'chau',  
      'key2': {'key1': 23},  
      'hola': [1, 2, 3, 'hola'],  
      123: (1, 8, 8)}
```

```
[15]: variable[123]
```

```
[15]: (1, 8, 8)
```

```
[16]: variable[456]
```

```
      □  
↳ -----  
  
      KeyError                                Traceback (most recent call□  
↳ last)  
  
      <ipython-input-16-19a3beaabb9> in <module>  
      ----> 1 variable[456]  
  
      KeyError: 456
```

```
[17]: variable[4.5] = (1,8)
```

```
[18]: variable
```

```
[18]: {'key1': 'chau',  
      'key2': {'key1': 23},  
      'hola': [1, 2, 3, 'hola'],  
      123: (1, 8, 8),  
      4.5: (1, 8)}
```

```
[19]: variable[[1, 8]] = "holas"
```

```
      □  
↳ -----  
  
      TypeError                                Traceback (most recent call□  
↳ last)  
  
      <ipython-input-19-3c25f1ad842f> in <module>  
      ----> 1 variable[[1, 8]] = "holas"
```

```
TypeError: unhashable type: 'list'
```

1.1 del

Para eliminar un un elemento del diccionario usamos la palabra del

```
[20]: variable
```

```
[20]: {'key1': 'chau',  
      'key2': {'key1': 23},  
      'hola': [1, 2, 3, 'hola'],  
      123: (1, 8, 8),  
      4.5: (1, 8)}
```

```
[21]: del variable[4.5]
```

```
[22]: variable
```

```
[22]: {'key1': 'chau',  
      'key2': {'key1': 23},  
      'hola': [1, 2, 3, 'hola'],  
      123: (1, 8, 8)}
```

1.2 Iterar en diccionarios

Cuando iteramos sobre diccionarios, se pueden obtener al mismo tiempo la clave y su valor correspondiente usando el método `items()`.

```
[30]: countries = {'Argentina': 'Buenos Aires', 'Bolivia': 'La Paz', 'Peru':  
    ↪{'capital': 'Lima', 'hab': 10000000}}  
for key, value in countries.items():  
    if isinstance(value, dict):  
        # diccionart  
        # for  
        #print(value)  
    pass  
else:  
    print(key, value)
```

```
Argentina Buenos Aires  
Bolivia La Paz
```

1.3 Dict Comprehension

```
[36]: fruits = ['apple', 'mango', 'banana', 'cherry']
      f_len = {}

      for f in fruits:
          f {'Bolivia': 'La Paz',
            'Peru': {'capital': 'Lima', 'hab': 10000000}}

      _len[f] = len(f)

      # con dict comp.
      # f_len = {f:len(f) for f in fruits}
```

```
[37]: f_len
```

```
[37]: {'apple': 5, 'mango': 5, 'banana': 6, 'cherry': 6}
```

```
[3]: # ¿Cómo sería con los países?
```

```
países = ["Argentina", "Bolivia", "Peru"]
capitales = ["BsAs", "La PAZ", "Lima"]
```

```
[ ]: # países=["Argentina","Bolivia","Peru"] capitales = ["Bs As","La Paz", "Lima"]
      ↪ff={p:c for p in países for c in capitales} print (ff)
```

1.4 list(d)

```
[45]: countries = {'Argentina': 'Buenos Aires', 'Bolivia': 'La Paz', 'Peru':
      ↪{'capital': 'Lima', 'hab': 10000000}}
      c = list(countries)
      c[0]
```

```
[45]: 'Argentina'
```

```
[46]: for c in list(countries):
      print(c)
```

```
Argentina
Bolivia
Peru
```

1.5 len(d)

```
[47]: len(countries)
```

```
[47]: 3
```

1.6 key in d - key not in d

```
[52]: try:
        countries['Brasil']
    except KeyError:
        print("pais no registrado")
```

pais no registrado

```
[61]: co = input()
        if co not in countries['Peru']:
            print("pais no registrado")
            #
        else:
            print(countries['Peru'][co])
```

capital
Lima

1.7 clear()

```
[62]: countries.clear()
```

```
[63]: countries
```

```
[63]: {}
```

1.8 copy()

```
[64]: countries = {'Argentina': 'Buenos Aires', 'Bolivia': 'La Paz', 'Peru': 'Lima'}
        ↪ {'capital': 'Lima', 'hab': 10000000}
        qwerty = countries.copy()
```

```
[66]: qwerty
```

```
[66]: {'Argentina': 'Buenos Aires',
        'Bolivia': 'La Paz',
        'Peru': {'capital': 'Lima', 'hab': 10000000}}
```

1.9 items()

```
[ ]: # ya lo vimos más arriba
countries = {'Argentina': 'Buenos Aires', 'Bolivia': 'La Paz', 'Peru': {}
            ↪ {'capital': 'Lima', 'hab': 10000000}}
for key, value in countries.items():
    if isinstance(value, dict):
        # diccionart
        # for
        # print(value)
        pass
    else:
        print(key, value)
```

1.10 keys()

```
[67]: countries = {'Argentina': 'Buenos Aires', 'Bolivia': 'La Paz', 'Peru': {}
            ↪ {'capital': 'Lima', 'hab': 10000000}}

for k in countries.keys():
    print(k)
```

Argentina
Bolivia
Peru

```
[68]: 'Argentina' in countries.keys()
```

[68]: True

1.11 update()

```
[70]: countries = {'Argentina': 'Buenos Aires', 'Bolivia': 'La Paz', 'Peru': {}
            ↪ {'capital': 'Lima', 'hab': 10000000}}

countries.update(Argentina='La Rioja')
```

```
[71]: countries
```

```
[71]: {'Argentina': 'La Rioja',
      'Bolivia': 'La Paz',
      'Peru': {'capital': 'Lima', 'hab': 10000000}}
```

```
[72]: countries.update(Chile='Santiago')
```

```
[73]: countries
```

```
[73]: {'Argentina': 'La Rioja',  
      'Bolivia': 'La Paz',  
      'Peru': {'capital': 'Lima', 'hab': 10000000},  
      'Chile': 'Santiago'}
```

```
[74]: countries.update(Chile=123)
```

```
[75]: countries
```

```
[75]: {'Argentina': 'La Rioja',  
      'Bolivia': 'La Paz',  
      'Peru': {'capital': 'Lima', 'hab': 10000000},  
      'Chile': 123}
```

```
[76]: countries.update(Chile='Santiago', Argentina='Bs As')
```

```
[79]: d = {'China': 'Pekin', 'Japón': 'Tokio'}
```

```
[80]: countries.update(d)
```

```
[81]: countries
```

```
[81]: {'Argentina': 'Bs As',  
      'Bolivia': 'La Paz',  
      'Peru': {'capital': 'Lima', 'hab': 10000000},  
      'Chile': 'Santiago',  
      'China': 'Pekin',  
      'Japón': 'Tokio'}
```

1.12 values()

```
[88]: for value in countries.values():  
      print(value)
```

```
Bs As  
La Paz  
{'capital': 'Lima', 'hab': 10000000}  
Santiago  
Pekin  
Tokio
```

1.13 Crear un diccionario a partir de dos iterables

```
[90]: paises = ["Argentina", "Bolivia", "Peru"]  
      capitales = ["Bs As", "La Paz", "Lima"]  
  
      ff= dict(zip(paises, capitales))
```

```
print(ff)
```

```
{'Argentina': 'Bs As', 'Bolivia': 'La Paz', 'Peru': 'Lima'}
```

1.14 Ejercicios

- Eliminar un items None de un diccionario.
- Filtrar las personas mayores o igual a 170cm en -> alturas = {'John': 175, 'Jane': 150, 'Jim': 155, 'Matt': 170}

```
[91]: d = {'Argentina': 'Bs As',  
          'Bolivia': 'La Paz',  
          'Peru': None,  
          'Chile': 'Santiago',  
          'China': None,  
          'Japón': 'Tokio'}
```

```
[92]: d
```

```
[92]: {'Argentina': 'Bs As',  
       'Bolivia': 'La Paz',  
       'Peru': None,  
       'Chile': 'Santiago',  
       'China': None,  
       'Japón': 'Tokio'}
```

```
[ ]: alturas = {'John': 175, 'Jane': 150, 'Jim': 155, 'Matt': 170, 'Yamila': 180,  
               ↪ 'Pedro': 169.99, 'Fede': '170'}
```