

Diccionarios

Un Diccionario es un tipo de datos en Python que permite organizar la información de manera semejante a una lista. Sin embargo, en una lista cada elemento tiene un índice asociado; en un diccionario, cada elemento tiene una llave asociada. Esta llave es el elemento que identifica de manera única a cada elemento. Esto significa que aunque los datos se puede repetir en una diccionario, las llaves son únicas ya que identifican a cada elemento (así como en una lista los índices también son únicos).

Un diccionario se especifica con `{ }` (como si fuera un libro) y tiene la forma `{llave: dato}`.

```
In [8]:  
alumno = {'nombre': 'Elmer', 'edad': 20, 'curso': 'fisica'}  
  
# Se imprimen los elementos del alumno1 utilizando las llaves de identificación de cada dato:  
print("alumno['nombre'] =", alumno['nombre'])  
print("alumno['edad'] =", alumno['edad'])  
print("alumno['curso'] =", alumno['curso'])  
  
alumno['nombre'] = Elmer  
alumno['edad'] = 20  
alumno['curso'] = fisica
```

Considere lo siguiente:

- Las llaves son cadenas. Pueden ser números y tuplas, pero no listas.
- Los datos se separan de las llaves con :
- Las llaves se especifican entre [], como sucede con los índices en otras secuencias en Python

Cuando se especifica una llave que no exista, Python retornará una excepción (*KeyError*):

```
In [15]:  
print(alumno['apellido'])  
  
Curio
```

Se puede editar un dato así como agregar un nuevo registro en un diccionario (cuidado que la llave no exista en el diccionario):

```
In [16]:  
alumno['edad'] = 24  
alumno['apellido'] = 'Curio'  
print(alumno)  
  
{'nombre': 'Elmer', 'curso': 'fisica', 'apellido': 'Curio', 'edad': 24}
```

E incluso se puede eliminar un dato, especificando la llave:

```
In [17]:  
del alumno['edad']  
print(alumno)  
  
{'nombre': 'Elmer', 'curso': 'fisica', 'apellido': 'Curio'}
```

Hay dos instrucciones que pueden resultar utiles:

```
diccionario.clear()    #Elimina todos los elementos de un diccionario, dejando este en blanco
del diccionario        #Elimina el diccionario completamente
```

Para interactuar con un diccionario es importante conocer cuales son sus especificaciones:

```
In [34]:
print(alumno, '\n')

print("Numero de elementos:", len(alumno), '\n')

print("Cuales son?", alumno.values(), '\n')

print("Que llaves tiene?", alumno.keys(), '\n')

print("Y como estan organizados?", alumno.items(), '\n')

print("Que dato tiene como nombre?", alumno.get('nombre' 'no existe'), '\n')

print("Que dato tiene como edad?", alumno.get('edad', 'no existe'), '\n')

{'nombre': 'Elmer', 'curso': 'fisica', 'apellido': 'Curio'}

Numero de elementos: 3

Cuales son? dict_values(['Elmer', 'fisica', 'Curio'])

Que llaves tiene? dict_keys(['nombre', 'curso', 'apellido'])

Y como estan organizados? dict_items([('nombre', 'Elmer'), ('curso', 'fisica'), ('apellido', 'Curio')])

Que dato tiene como nombre? None

Que dato tiene como edad? no existe
```

Un diccionario es un iterable, aunque no se puede utilizar directamente en un lazo for: las llaves y valores debe de ser desempaquetados previamente. Si no se separan, solo se iterará sobre las llaves:

```
In [36]:
for data in alumno:
    print(data)

nombre
curso
apellido
```

```
In [39]:
for key, value in alumno.items():
    print(key, "->", value)

nombre -> Elmer
curso -> fisica
apellido -> Curio
```

Listas y diccionarios

Como puede verse, los diccionarios son ideales para guardar información como si fueran las filas de una tabla. Se puede hacer una lista de diccionarios para crear una tabla (en lugar de hacer una lista de listas, en donde los datos no son tan fáciles de visualizar):

```
In [12]:  
listado_alumnos = [{'nombre': 20},  
                  {'nombre': 12},  
                  {'nombre': 14},  
                  {'nombre': 34},  
                  {'nombre': 50}]  
  
for registro in listado_alumnos:  
    print(registro['nombre'])  
  
20  
12  
14  
34  
50
```

Así también, se puede tener una lista dentro de un diccionario:

```
In [2]:  
alumnos = {  
    'Elmer': ['Fisica I', 'Programacion I', 'Lenguaje I'],  
    'Elba': ['Fisica I', 'Lenguaje II'],  
    'Mario': ['Lenguaje II', 'Matematica II', 'Lenguaje I']  
}  
  
for nombre, cursos in alumnos.items():  
    print("Cursos de", nombre, ":")  
    for i in cursos:  
        print("  -", i)  
    print()  
  
Cursos de Elmer :  
  - Fisica I  
  - Programacion I  
  - Lenguaje I  
  
Cursos de Elba :  
  - Fisica I  
  - Lenguaje II  
  
Cursos de Mario :  
  - Lenguaje II  
  - Matematica II  
  - Lenguaje I
```

O inclusive un diccionario dentro de un diccionario...

```
In [9]:
alumnos ={
    'Elmer':{
        'Curso1':{
            'Nombre': 'Programacion I',
            'Codigo': 'EL129',
            'Creditos': 3
        },
        'Curso2':{
            'Nombre': 'Lenguaje II',
            'Codigo': 'HU101',
            'Creditos': 2
        },
    },
    'Elba':{
        'Curso1':{
            'Nombre': 'Programacion II',
            'Codigo': 'EL139',
            'Creditos': 4
        },
        'Curso2':{
            'Nombre': 'Fisica I',
            'Codigo': 'CI121',
            'Creditos': 4
        },
    }
}

for i, cursos in alumnos.items():
    for j, datos in cursos.items():
        print("{:8} {:20} Creditos: {}".format(datos['Codigo'], datos['Nombre'], datos['Creditos']))
```

EL129	Programacion I	Creditos: 3
HU101	Lenguaje II	Creditos: 2
EL139	Programacion II	Creditos: 4
CI121	Fisica I	Creditos: 4

Un diccionario, por lo tanto, es un tipo de datos que resulta útil como repositorio de información organizada con etiquetas y es muy útil para definir tablas de datos.