

# Diccionarios

CS1100 - Introducción a Ciencia de la Computación  
UTEC

# Logro de la Sesión

Al finalizar esta sesión, estarás en la capacidad de:

- Comprender como funciona un diccionario en Python y en que tipo de problemas puede ser utilizado.

## Logro de la Sesión

Al finalizar esta sesión, estarás en la capacidad de:

- Comprender como funciona un diccionario en Python y en que tipo de problemas puede ser utilizado.
- Crear y utilizar diccionarios en Python par resolver problemas.

# Logro de la Sesión

Al finalizar esta sesión, estarás en la capacidad de:

- Comprender como funciona un diccionario en Python y en que tipo de problemas puede ser utilizado.
- Crear y utilizar diccionarios en Python par resolver problemas.
- Video: [▶ Link](#)

# ¿Qué es un diccionario?

- En Python, un diccionario es una colección de datos que, a diferencia de otras colecciones que almacenan simples valores, permite guardar valores similares a un mapa con la forma **clave:valor**.
- La clave permite realizar búsquedas de manera más eficiente.
- Las claves de un diccionario deben ser únicas y de un tipo de dato inmutable (integers, strings, tuplas).
- Los valores pueden repetirse y pueden ser cualquier tipo de dato.

# ¿Cómo crear un diccionario?

## ■ Existen dos formas de crear un diccionario:

```
1 #La primera forma, haciendo uso de la palabra dict
2 d = dict()
3
4 #La segunda forma, haciendo uso de {}
5 d = {}
```

Cabe resaltar que en ambos casos el diccionario se creará vacío.

## ■ Para crear un diccionario con valores iniciales, se puede hacer de distintas formas:

```
1 # Creando un diccionario con clave entera
2 d = {1: 'Geeks', 2: 'For', 3: 'Geeks'}
3 print("\nDiccionario con clave entera: ")
4 print(d)
5
6 # Creando un diccionario con claves mixtas
7 d = {'name': 'Geeks', 1: [1, 2, 3, 4]}
8 print("\nDiccionario usando clave mixtas: ")
9 print(d)
10
11 # Creando un diccionario con dict()
12 d = dict({1: 'Geeks', 2: 'For', 3: 'Geeks'})
13 print("\nDiccionario haciendo uso de dict(): ")
14 print(d)
15
16 # Creando un diccionario con cada elemento como un par
17 d = dict([(1, 'Geeks'), (2, 'For')])
18 print("\nDiccionario con cada elemento como un par: ")
19 print(d)
```

# ¿Cómo acceder a los datos de un diccionario?

- Para acceder los elementos del diccionario a través de la clave, se utiliza []:

```
1 d = {'name': 'Zara', 'edad': 7, 'clase': 'First'}
2 print ("d['name']: ", d['name'])
3 print ("d['edad']: ", d['edad'])
```

- ¿Que pasaría si accedemos a través de una clave que no existe?

```
1 d = {'name': 'Zara', 'edad': 7, 'clase': 'First'};
2 print ("d['Alice']: ", d['Alice'])
```

- En caso se quiera obtener únicamente las claves de un diccionario se puede hacer por medio de la función *keys()*. Para acceder a los valores, se debe utilizar la función *values()*.

```
1 d = {'nombre': 'Zara', 'edad': 7}
2 print("Claves : %s" % d.keys())
3 print("Valores : %s" % d.values())
```



# ¿Cómo actualizar un diccionario?

- Un diccionario se puede actualizar de tres formas: añadiendo un nuevo par (clave-valor), modificando el valor para una clave o eliminando un par. El siguiente código muestra como añadir un nuevo par y modificar uno existente:

```
1 d = {'nombre': 'Zara', 'edad': 7, 'clase': 'First'}  
2 d['edad'] = 8; # actualizando una clave que ya existe  
3 d['colegio'] = "DPS colegio" # añadir un nuevo par  
4  
5 print ("d['edad']: ", d['edad'])  
6 print ("d['colegio']: ", d['colegio'])
```



# ¿Cómo eliminar datos de un diccionario?

- Python permite eliminar datos de tres maneras: a través de la clave, eliminando únicamente un par. Eliminando toda la información del diccionario y eliminando completamente el diccionario (como variable).

```
1 d = {'nombre': 'Zara', 'edad': 7, 'grado': 'First'}
2
3 del d['nombre'] # elimina par con clave 'nombre'
4 d.clear()      # elimina todos los datos del diccionario
5 del d          # elimina el diccionario
6
7 print ("d['edad']: ", d['edad'])
8 print ("d['colegio']: ", d['colegio'])
```

# Ejercicios

- 1 Implemente una función que retorne el mayor valor de un diccionario.
- 2 Implemente una función que reciba clave y valor desde el teclado y las añada a un diccionario. Debe ser posible ingresar valores hasta que se ingrese “x” como clave.
- 3 Implemente una función que permita mapear dos listas (una con las claves y otra con los valores) dentro de un diccionario.
- 4 Implemente una función que reciba como parámetro un número y retorne el mes que corresponde a ese número. Si se recibe un valor menor a 1 y mayor a 12 retornar “Mes no válido”.
- 5 Implemente una función que reciba como parámetro un mes y retorne el número que corresponde a ese mes. Si se recibe un mes inválido retornar -1.
- 6 Implemente una función que reciba como parámetro un string y retorne un diccionario la cantidad de veces que se repite cada palabra en el string.
- 7 Implemente una función que reciba como parámetro un string y retorne un diccionario la cantidad de veces que se repite cada carácter en el string.

Las soluciones se encuentran aquí.

## No olvidar...

- Un diccionario es una colección que permite **indexar** la información haciendo que las búsquedas sean más eficientes.
- Cada elemento tiene dos partes: clave y valor.
- Las claves deben ser únicas.

