

Programación con python

Dictionaries

Dictionaries

Un diccionario, como su nombre lo sugiere, te permite almacenar y buscar elementos formateados como pares clave-valor, de manera similar a cómo un diccionario físico te permite buscar palabras y sus definiciones asociadas.

En Python, un diccionario (o dict) es una colección que asocia claves inmutables con valores de cualquier tipo.

Dictionaries

El siguiente código crea y accede a un diccionario que almacena la edad de cada persona de acuerdo con su nombre:

```
ages = {  
    "Luis": 30,  
    "Alex": 33,  
    "Juan": 30  
}  
  
print("Alex tiene", ages["Alex"], "años")
```

Dictionaries

Acceder a los elementos del diccionario.

```
x = {2: "hello", "1": 5}
```

```
print(x[2])
```

```
print(x["1"])
```

Dictionaries

Modificar el valor de una clave existente

```
x = {"key": 1}
```

```
x["key"] = "value"
```

```
print(x["key"])
```

Dictionaries

Agregar una nueva clave al diccionario

```
x = {"key": 1}
```

```
x["key2"] = "value"
```

```
print(x)
```

Dictionaries

Agregar un elemento a un diccionario vacío:

```
x = {}
```

```
x["key"] = "value"
```

```
print(x)
```

```
print(x["key"])
```

Dictionaries

Si intentamos acceder a algo que no existe, al igual que en otras estructuras tenemos un error:

```
Traceback (most recent call last):  
  File "c:\ProgrammingExpert\ProgrammingFundamentals\17_dictionaries.py", line 5, in <module>  
    print(x[2])  
KeyError: 2
```


Dictionaries

Crear un diccionario usando dict():

```
x = dict()
```

```
x["key2"] = "value"
```

```
print(x["key2"])
```

Dictionaries

Eliminar un elemento de un diccionario

```
x = {1: 1, 2: 2, 3: 3}
```

```
del x[1]
```

```
print(x)
```

Dictionaries

Verificar si una clave existe en un diccionario:

```
x = {1: 1, 2: 2, 3: 3}
```

```
contains = 1 in x
```

```
print(contains)
```

Dictionaries

No encontrada:

```
x = {2: 1, 3: 3}
```

```
contains = 1 in x
```

```
print(contains)
```

Aquí la clave 1 ya no existe dentro del diccionario, por lo tanto, la expresión devuelve False.

Dictionaries

Obtener todos los valores de un diccionario:

```
x = {2: 1, 3: 3}
```

```
values = x.values()
```

```
print(values)
```

Salida → dict_values([1, 3])

El método `.values()` devuelve una vista con todos los valores del diccionario, pero no una lista como tal.

Dictionaries

Obtener todos los valores de un diccionario:

```
x = {2: 1, 3: 3}
```

```
values = x.values()
```

```
print(values)
```

Salida → dict_values([1, 3])

El método `.values()` devuelve una vista con todos los valores del diccionario, pero no una lista como tal.

Dictionaries

Convertir los valores del diccionario a una lista

```
x = {2: 1, 3: 3}
```

```
values = list(x.values())
```

```
print(values[0])
```

Aquí los valores se convierten a una lista, lo que permite acceder a ellos mediante su índice, cómo `values[0]`.

Dictionaries

Obtener las claves de un diccionario:

```
x = {2: 1, 3: 3}
```

```
keys = list(x.keys())
```

```
print(keys[0])
```


Dictionaries

Obtener las claves y valores con `.items()`:

```
x = {2: 1, 3: 3}
```

```
items = list(x.items())
```

```
print(items)
```

- El método `.items()` devuelve pares clave-valor del diccionario.
- Al convertirlo con `list()`, obtenemos una lista de tuplas donde cada tupla representa una pareja (clave, valor).
- Luego, `print(items)` muestra todas las parejas.

Dictionaries

Recorrer un diccionario:

```
x = {2: 1, 3: 3}
```

```
for key in x:  
    value = x[key]  
    print(key, value)
```

Dictionaries

```
x = {2: 1, 3: 3}
```

```
for key, value in x.items():  
    print(key, value)
```

- El método `.items()` devuelve una colección de pares (clave, valor).
- En cada iteración del `for`, Python desempaqueta esos pares en las variables `key` y `value`.
- Luego, se imprime cada par en una línea.

Dictionaries

Recorrer un diccionario con for usando .items()

```
x = {2: 1, 3: 3}
```

```
for key, value in x.items():  
    print("Clave:", key, "→ Valor:", value)
```

Clave: 2 → Valor: 1

Clave: 3 → Valor: 3

Dictionaries

Contar elementos en un diccionario:

```
x = {2: 1, 3: 3}
```

```
print(len(x))
```

Dictionaries

Este código muestra cómo usar el método `.get()` de los diccionarios para acceder a un valor con una clave específica y manejar el caso en que la clave no exista.

```
x = {2: 1, 3: 3}
```

```
x[4] = x.get(4, 0) + 1
```

```
print(x)
```

Dictionaries

- La instrucción `x.get(4, 0)` busca la clave 4 dentro del diccionario:
- Si la clave existe, devuelve su valor.
- Si no existe, devuelve el valor por defecto que se indica (en este caso 0).
- Como 4 no existe aún, `x.get(4, 0)` devuelve 0.
- Luego se le suma 1 $\rightarrow 0 + 1 = 1$.
- Finalmente se asigna ese resultado a `x[4]`, creando una nueva clave:

`{2: 1, 3: 3, 4: 1}`

Dictionaries

Si ejecutas de nuevo:

```
x[4] = x.get(4, 0) + 1  
print(x)
```

Salida:

```
{2: 1, 3: 3, 4: 2}
```

Ahora la clave 4 ya existía, así que `x.get(4, 0)` devolvió 1, se sumó 1, y el nuevo valor se actualizó a 2.

Dictionaries

Esto es útil para contadores, acumuladores o frecuencias en diccionarios.

Dictionaries

Ejemplo real, Contar caracteres en una cadena con un diccionario:

```
characters = {}
```

```
string = "hello world"
```

```
for char in string:
```

```
    characters[char] = characters.get(char, 0) + 1
```

```
print(characters)
```

Resultado/Salida → {'h': 1, 'e': 1, 'l': 3, 'o': 2, ' ': 1, 'w': 1, 'r': 1, 'd': 1}

Dictionaries

1. Se crea un diccionario vacío llamado `characters` para guardar los resultados.
2. La variable `string` contiene el texto "hello world".
3. El bucle `for char in string`: recorre cada carácter de la cadena (letras y espacios).

En cada iteración:

- Se usa `.get(char, 0)` para obtener cuántas veces ha aparecido ese carácter hasta ahora (si no ha aparecido, devuelve 0).
- Luego se suma 1 y se guarda de nuevo en el diccionario bajo la clave `char`.
- Al final, el diccionario contiene cada carácter como clave y su número de apariciones como valor.

Dictionaries

Contar números ingresados por el usuario:

```
counts = {}
```

```
while True:
```

```
    num = input("Number (enter q to quit): ")
```

```
    if num == "q":  
        break
```

```
    num = int(num)  
    counts[num] = counts.get(num, 0) + 1
```

```
print(counts)
```

Dictionaries

Ejemplo de ejecución:

Number (enter q to quit): 5

Number (enter q to quit): 2

Number (enter q to quit): 5

Number (enter q to quit): q

{5: 2, 2: 1}

Actividad

Crea un diccionario llamado student con las siguientes claves y valores:

"name": tu nombre

"age": tu edad

"grade": tu promedio actual

Después:

Imprime todo el diccionario.

Imprime solo el valor asociado a "name".

Cambia el valor de "grade" por un número diferente y vuelve a imprimir el diccionario.

**NO USE INTELIGENCIA ARTIFICIAL PARA SU TRABAJO
ENTREGUELA EN LA SECCION DE TAREAS**

Ing. Luis Francisco Contreras González

Actividad

Crea un diccionario con tres países como claves y sus capitales como valores.

Luego:

Agrega un nuevo país y su capital.

Elimina uno de los países existentes.

Imprime el resultado final.

**NO USE INTELIGENCIA ARTIFICIAL PARA SU TRABAJO
ENTREGUELA EN LA SECCION DE TAREAS**

Ing. Luis Francisco Contreras González

Actividad

Crea un diccionario con nombres de frutas como claves y sus precios como valores.
Muestra cuántas frutas contiene.

**NO USE INTELIGENCIA ARTIFICIAL PARA SU TRABAJO
ENTREGUELA EN LA SECCION DE TAREAS**

Ing. Luis Francisco Contreras González

Actividad

Crea un diccionario con nombres de alumnos como claves y sus calificaciones como valores.

Luego imprime algo como:

Alumno: Ana → Calificación: 9.5

Alumno: Luis → Calificación: 8.7

**NO USE INTELIGENCIA ARTIFICIAL PARA SU TRABAJO
ENTREGUELA EN LA SECCION DE TAREAS**

Ing. Luis Francisco Contreras González

Actividad

Usa el mismo diccionario del ejercicio anterior y:

Muestra todas las claves.

Muestra todos los valores.

Convierte `.items()` en una lista e imprime el primer elemento.

**NO USE INTELIGENCIA ARTIFICIAL PARA SU TRABAJO
ENTREGUELA EN LA SECCION DE TAREAS**

Ing. Luis Francisco Contreras González

Actividad

Crea un diccionario con tres materias y sus calificaciones.

Luego pide al usuario que ingrese el nombre de una materia.

Si existe, muestra su calificación.

Si no, muestra un mensaje indicando que no está registrada.

**NO USE INTELIGENCIA ARTIFICIAL PARA SU TRABAJO
ENTREGUELA EN LA SECCION DE TAREAS**

Ing. Luis Francisco Contreras González

Actividad

Pide al usuario una frase.

Usa un diccionario para contar cuántas veces aparece cada palabra.

**NO USE INTELIGENCIA ARTIFICIAL PARA SU TRABAJO
ENTREGUELA EN LA SECCION DE TAREAS**

Ing. Luis Francisco Contreras González

Actividad

Pide al usuario una palabra o frase e imprime cuántas veces aparece cada letra (como en "hello world").

Ignora los espacios.

**NO USE INTELIGENCIA ARTIFICIAL PARA SU TRABAJO
ENTREGUELA EN LA SECCION DE TAREAS**

Ing. Luis Francisco Contreras González

Actividad

Crea un programa que guarde en un diccionario las calificaciones de tres materias:

matematicas, historia, programacion.

Pide al usuario que ingrese la calificación de cada una.

Guarda las materias como claves y las calificaciones como valores.

Calcula el promedio de todas las calificaciones.

Muestra el resultado con un mensaje como:

Promedio general: 8.7

**NO USE INTELIGENCIA ARTIFICIAL PARA SU TRABAJO
ENTREGUELA EN LA SECCION DE TAREAS**

Ing. Luis Francisco Contreras González

Actividad

Crea un pequeño diccionario bilingüe (español → inglés) con al menos 5 palabras.

Ejemplo inicial:

```
diccionario = {"gato": "cat", "perro": "dog",  
"casa": "house", "rojo": "red", "verde": "green"}
```

**NO USE INTELIGENCIA ARTIFICIAL PARA SU TRABAJO
ENTREGUELA EN LA SECCION DE TAREAS**

Ing. Luis Francisco Contreras González

Actividad

Pide al usuario que escriba una palabra en español.

Si la palabra existe, muestra su traducción.

Si no está, muestra el mensaje:

"Esa palabra no está en el diccionario."

Permite que el usuario siga buscando palabras hasta que escriba "salir".

**NO USE INTELIGENCIA ARTIFICIAL PARA SU TRABAJO
ENTREGUELA EN LA SECCION DE TAREAS**

Ing. Luis Francisco Contreras González

Actividad

Dada una lista de números, encuentra cuál número aparece con más frecuencia y cuántas veces aparece, usando un diccionario.

ejemplo de salida:

El número 2 aparece 3 veces.

**NO USE INTELIGENCIA ARTIFICIAL PARA SU TRABAJO
ENTREGUELA EN LA SECCION DE TAREAS**

Ing. Luis Francisco Contreras González