**Python Dictionary**

A Python dictionary is a collection of items, similar to lists and tuples. However, unlike lists and tuples, each item in a dictionary is a **key-value** pair (consisting of a key and a value).

**Create a Dictionary**

We create a dictionary by placing key: value pairs inside curly brackets {}, separated by commas. For example,

# creating a dictionary

country\_capitals = {

"Germany": "Berlin",

"Canada": "Ottawa",

"England": "London"

}

# printing the dictionary

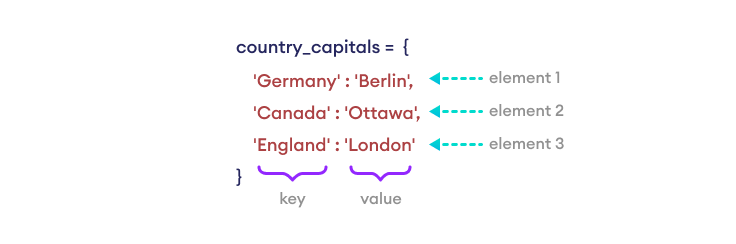
print(country\_capitals)

[Run Code](https://www.programiz.com/python-programming/online-compiler)

**Output**

{'Germany': 'Berlin', 'Canada': 'Ottawa', 'England': 'London'}

The country\_capitals dictionary has three elements (key-value pairs), where 'Germany' is the key and 'Berlin' is the value assigned to it and so on.

Python Dictionary

**Notes**:

* Dictionary keys must be immutable, such as tuples, strings, integers, etc. We cannot use mutable (changeable) objects such as lists as keys.
* We can also create a dictionary using a Python built-in function dict(). To learn more, visit [Python dict()](https://www.programiz.com/python-programming/methods/built-in/dict).

**Valid and Invalid Dictionaries**

Keys of a dictionary must be immutable

Keys of a dictionary must be unique

**Access Dictionary Items**

We can access the value of a dictionary item by placing the key inside square brackets.

country\_capitals = {

"Germany": "Berlin",

"Canada": "Ottawa",

"England": "London"

}

# access the value of keys

print(country\_capitals["Germany"]) # Output: Berlin

print(country\_capitals["England"]) # Output: London

[Run Code](https://www.programiz.com/python-programming/online-compiler)

**Note:** We can also use the [get()](https://www.programiz.com/python-programming/methods/dictionary/get) method to access dictionary items.

**Add Items to a Dictionary**

We can add an item to a dictionary by assigning a value to a new key. For example,

country\_capitals = {

"Germany": "Berlin",

"Canada": "Ottawa",

}

# add an item with "Italy" as key and "Rome" as its value

country\_capitals["Italy"] = "Rome"

print(country\_capitals)

[Run Code](https://www.programiz.com/python-programming/online-compiler)

**Output**

{'Germany': 'Berlin', 'Canada': 'Ottawa', 'Italy': 'Rome'}

**Remove Dictionary Items**

We can use the [del](https://www.programiz.com/python-programming/del) statement to remove an element from a dictionary. For example,

country\_capitals = {

"Germany": "Berlin",

"Canada": "Ottawa",

}

# delete item having "Germany" key

del country\_capitals["Germany"]

print(country\_capitals)

[Run Code](https://www.programiz.com/python-programming/online-compiler)

**Output**

{'Canada': 'Ottawa'}

**Note**: We can also use the [pop()](https://www.programiz.com/python-programming/methods/dictionary/pop) method to remove an item from a dictionary.

If we need to remove all items from a dictionary at once, we can use the [clear()](https://www.programiz.com/python-programming/methods/dictionary/clear) method.

country\_capitals = {

"Germany": "Berlin",

"Canada": "Ottawa",

}

# clear the dictionary

country\_capitals.clear()

print(country\_capitals)

[Run Code](https://www.programiz.com/python-programming/online-compiler)

**Output**

{}

**Change Dictionary Items**

Python dictionaries are mutable (changeable). We can change the value of a dictionary element by referring to its key. For example,

country\_capitals = {

"Germany": "Berlin",

"Italy": "Naples",

"England": "London"

}

# change the value of "Italy" key to "Rome"

country\_capitals["Italy"] = "Rome"

print(country\_capitals)

[Run Code](https://www.programiz.com/python-programming/online-compiler)

**Output**

{'Germany': 'Berlin', 'Italy': 'Rome', 'England': 'London'}

**Note**: We can also use the [update()](https://www.programiz.com/python-programming/methods/dictionary/update) method to add or change dictionary items.

**Iterate Through a Dictionary**

A dictionary is an ordered collection of items (starting from Python 3.7), therefore it maintains the order of its items.

We can iterate through dictionary keys one by one using a [for loop](https://www.programiz.com/python-programming/for-loop).

country\_capitals = {

"United States": "Washington D.C.",

"Italy": "Rome"

}

# print dictionary keys one by one

for country in country\_capitals:

print(country)

print()

# print dictionary values one by one

for country in country\_capitals:

capital = country\_capitals[country]

print(capital)

[Run Code](https://www.programiz.com/python-programming/online-compiler)

**Output**

United States

Italy

Washington D.C.

Rome

**Find Dictionary Length**

We can find the length of a dictionary by using the [len()](https://www.programiz.com/python-programming/methods/built-in/len) function.

country\_capitals = {"England": "London", "Italy": "Rome"}

# get dictionary's length

print(len(country\_capitals)) # Output: 2

numbers = {10: "ten", 20: "twenty", 30: "thirty"}

# get dictionary's length

print(len(numbers)) # Output: 3

countries = {}

# get dictionary's length

print(len(countries)) # Output: 0

[Run Code](https://www.programiz.com/python-programming/online-compiler)

**Python Dictionary Methods**

Here are some of the commonly used [dictionary methods](https://www.programiz.com/python-programming/methods/dictionary).

|  |  |
| --- | --- |
| Function | Description |
| [pop()](https://www.programiz.com/python-programming/methods/dictionary/pop) | Removes the item with the specified key. |
| [update()](https://www.programiz.com/python-programming/methods/dictionary/update) | Adds or changes dictionary items. |
| [clear()](https://www.programiz.com/python-programming/methods/dictionary/clear) | Remove all the items from the dictionary. |
| [keys()](https://www.programiz.com/python-programming/methods/dictionary/keys) | Returns all the dictionary's keys. |
| [values()](https://www.programiz.com/python-programming/methods/dictionary/values) | Returns all the dictionary's values. |
| [get()](https://www.programiz.com/python-programming/methods/dictionary/get) | Returns the value of the specified key. |
| [popitem()](https://www.programiz.com/python-programming/methods/dictionary/popitem) | Returns the last inserted key and value as a tuple. |
| [copy()](https://www.programiz.com/python-programming/methods/dictionary/copy) | Returns a copy of the dictionary. |

**Dictionary Membership Test**

We can check whether a key exists in a dictionary by using the in and not in operators.

file\_types = {

".txt": "Text File",

".pdf": "PDF Document",

".jpg": "JPEG Image",

}

# use of in and not in operators

print(".pdf" in file\_types) # Output: True

print(".mp3" in file\_types) # Output: False

print(".mp3" not in file\_types) # Output: True

[Run Code](https://www.programiz.com/python-programming/online-compiler)

**Note:** The in operator checks whether a key exists; it doesn't check whether a value exists or not.

Before we wrap up, let’s put your knowledge of Python dictionary to the test! Can you solve the following challenge?

Challenge:

Write a function to merge two dictionaries.

* Merge dict1 and dict2, then return the merged dictionary.

1

2

def merge\_dictionaries(dict1, dict2):

Check Code