

Dictionaries

```
ages = {"Dave": 24, "Mary": 42, "John": 58}
print(ages["Dave"])
print(ages["Mary"])
OR
primary = {
    "red": [255, 0, 0],
    "green": [0, 255, 0],
    "blue": [0, 0, 255],
}
```

Dictionaries are data structures used to map arbitrary keys to values

get Function

```
pairs = {1: "apple",
        "orange": [2, 3, 4],
        True: False,
        None: "True",
}
print(pairs.get("orange"))
print(pairs.get(7))
print(pairs.get(12345, "not in dictionary"))
.....
>>>
[2, 3, 4]
None
not in dictionary
>>>
```

A useful dictionary method is `get`. It does the same thing as indexing, but if the key is not found in the dictionary it returns another specified value instead ('None', by default).

Assignment

```
squares = {1: 1, 2: 4, 3: "error", 4: 16,}
squares[8] = 64
squares[3] = 9
print(squares)
-----
{1: 1, 2: 4, 3: 9, 4: 16, 8: 64}
```

Just like lists, dictionary keys can be assigned to different values.

finding keys

```
nums = {
    1: "one",
    2: "two",
    3: "three",
}
print(1 in nums)
print(4 not in nums)
print(not 4 in nums)
-----
True
True
```

To determine whether a key is in a dictionary, you can use `in` and `not in`, just as you can for a list.



By Nima (nimakarimian)

Published 26th June, 2020.

Last updated 26th June, 2020.

Page 1 of 1.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

cheatography.com/nimakarimian/

www.nimakarimian.ir