



Python Dictionaries

DOWNLOAD PDF FROM TELEGRAM 
 CODING BUGS  NOTES GALLERY

CHEATSHEET

A dictionary is an unordered collection of key-value pairs, where each key is unique and associated with a corresponding value. Dictionaries are one of the built-in data types in Python and are represented using curly braces {}.

Fromkeys

Create a new dictionary with specified keys and a default value.

Get

Return the value of the specified key or a default value if the key is not found

Items

Return a view of key-value pairs as tuples.

Keys

Return a view of the dictionary's keys.

Pop

Remove and return the value associated with the specified key.

Popitem

Remove and return the last key-value pair as a tuple.

Values

Return a view of the dictionary's values.

Update

Update the dictionary with key-value pairs from another dictionary or iterable.

Copy

Create a shallow copy of the dictionary.

Clear

Remove all items from the dictionary.



Fromkeys

Create a new dictionary with specified keys and a default value.

```
# Create a new dictionary using fromkeys() with a default value
default_value = 10
keys = ["apple", "banana", "orange", "grape"]
fruits_count = dict.fromkeys(keys, default_value)

print(fruits_count)
# Output: {'apple': 10, 'banana': 10, 'orange': 10, 'grape': 10}
```

Get

Return the value of the specified key or a default value if the key is not found

```
# Create a dictionary of student grades
student_grades = {
    "A": 85,
    "B": 78,
    "C": 92,
    "D": 88
}

# Get the grade of a student using get() with a default value
grade_a = student_grades.get("A", "Not Found")
grade_j = student_grades.get("J", "Not Found")

print(grade_a) # Output: 85 (Grade of A is found)
print(grade_j) # Output: Not Found (J is not in the dictionary)
```

DIGYQUANT ANALYTICS

Training Program



**MACHINE LEARNING
ARTIFICIAL INTELLIGENCE**

12 AUGUST

ENROLL NOW





Items

Return a view of key-value pairs as tuples.

```
# Create a dictionary of student names and their ages
student_ages = {
    "A": 20,
    "B": 22,
    "C": 19,
    "D": 21
}

# Using items() to retrieve the key-value pairs as tuples
print(student_ages.items())
# Output: dict_items([('A', 20), ('B', 22), ('C', 19), ('D', 21)])

# Converting the items view into a list of tuples
student_items_list = list(student_ages.items())

print(student_items_list)
# Output: [('A', 20), ('B', 22), ('C', 19), ('D', 21)]
```

DigyQuant Benefits

**1:1 Mentorship from
industry experts**

**In-depth Q&A and
Clarification**

**Mentors with 12+ years of
experience**

**Hands-on Learning with
Projects and Exercises**

Swipe >>>



Keys

Return a view of the dictionary's keys.

```
# Create a dictionary of student names and their ages
student_ages = {"A": 20, "B": 22, "C": 19, "D": 21}
```

```
# Using keys() to retrieve the view of keys
print(student_ages.keys())
# Output: dict_keys(['A', 'B', 'C', 'D'])
```

```
# Converting the keys view into a list
student_keys_list = list(student_ages.keys())
```

```
print(student_keys_list)
# Output: ['A', 'B', 'C', 'D']
```

Pop

Remove and return the value associated with the specified key.

```
# Create a dictionary of student names and their ages
student_ages = {"A": 20, "B": 22, "C": 19, "D": 21}
```

```
# Remove and return the value associated with the key "Bob"
removed_age = student_ages.pop("B")
```

```
print(removed_age) # Output: 22 (Value associated with "B" key)
print(student_ages) # Output: {'A': 20, 'C': 19, 'D': 21}
```





Popitem

Remove and return the last key-value pair as a tuple.

```
# Create a dictionary of student names and their ages
student_ages = {"A": 20, "B": 22, "C": 19, "D": 21}

# Remove and return the last key-value pair from the dictionary
last_item = student_ages.popitem()

print(last_item) # Output: ('D', 21) (Last key-value pair)
print(student_ages) # Output: {'A': 20, 'B': 22, 'C': 19}
```

Values

Return a view of the dictionary's values.

```
# Create a dictionary of student names and their ages
student_ages = {"A": 20, "B": 22, "C": 19, "D": 21}

# Using values() to retrieve the view of values
print(student_ages.values())
# Output: dict_values([20, 22, 19, 21])

# Converting the values view into a list
student_ages_list = list(student_ages.values())

print(student_ages_list)
# Output: [20, 22, 19, 21]
```





Update

Update the dictionary with key-value pairs from another dictionary or iterable.

```
# Create two dictionaries
dict1 = {"A": 20, "B": 22}
dict2 = {"C": 19, "D": 21}

# Update dict1 with the key-value pairs from dict2
dict1.update(dict2)

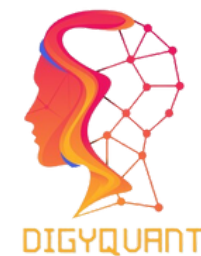
print(dict1)
# Output: {'A': 20, 'B': 22, 'C': 19, 'D': 21}
```

```
# Create a list of key-value pairs as tuples
key_value_pairs = [("E", 23), ("F", 25)]

# Update dict1 with the key-value pairs from the list
dict1.update(key_value_pairs)

print(dict1)
# Output: {'A': 20, 'B': 22, 'C': 19, 'D': 21, 'E': 23, 'F': 25}
```





Copy

Create a shallow copy of the dictionary.

```
# Create a dictionary
original_dict = {"A": 20, "B": 22, "C": 19}

# Create a copy of the dictionary using the copy() method
copied_dict = original_dict.copy()

# Modify the copied dictionary
copied_dict["D"] = 21

print(original_dict)
# Output: {'A': 20, 'B': 22, 'C': 19}

print(copied_dict)
# Output: {'A': 20, 'B': 22, 'C': 19, 'D': 21}
```

Clear

Remove all items from the dictionary.

```
# Create a dictionary of student names and their ages
student_ages = {"A": 20, "B": 22, "C": 19, "D": 21}

# Clear the dictionary (remove all key-value pairs)
student_ages.clear()

print(student_ages)
# Output: {}
```



DIGYQUANT ANALYTICS

Training Program

DOWNLOAD PDF FROM TELEGRAM 
 CODING BUGS  NOTES GALLERY



**MACHINE LEARNING
ARTIFICIAL INTELLIGENCE**

12 AUGUST

ENROLL NOW

*Thank
You*

