

## Python Output

HTML CSS JAVASCRIPT SQL

PYTHON

JAVA

PHP

HOW TO

W3.CSS

C

C++

C#

BOOTSTRAP

REACT

MYSQL

JQUERY

Python Variables

Python Data Types

Python Numbers

Python Casting

Python Strings

Python Booleans

Python Operators

Python Lists

Python Tuples

Python Sets

Python Dictionaries

Python Dictionaries

Access Items

Change Items

Add Items

Remove Items

Loop Dictionaries

Copy Dictionaries

Nested Dictionaries

Dictionary Methods

Dictionary Exercises

Python If...Else

Python Match

Python While Loops

Python For Loops

# Python Dictionaries

[Previous](#)[Next](#)

```
thisdict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}
```

## Dictionary

Dictionaries are used to store data values in key:value pairs.

**Get  
Certified!**

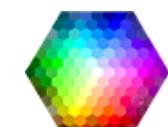
Document your skills with all  
of W3Schools Certificates

\$1,995 **\$499**

Save 75%



COLOR  
PICKER



[REMOVE ADS](#)

As of Python version 3.7, dictionaries are *ordered*. In Python 3.6 and earlier, dictionaries are *unordered*.

Dictionaries are written with curly brackets, and have keys and values:

## Example

[Get your own Python Server](#)

Create and print a dictionary:

```
thisdict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}  
print(thisdict)
```

[Try it Yourself »](#)

## Dictionary Items

Dictionary items are ordered, changeable, and do not allow duplicates.

Dictionary items are presented in key:value pairs, and can be referred to by using the key name.

## Example

Print the "brand" value of the dictionary:

```
    "model": "Mustang",
    "year": 1964
}
print(thisdict["brand"])
```

[Try it Yourself »](#)

## Ordered or Unordered?

As of Python version 3.7, dictionaries are *ordered*. In Python 3.6 and earlier, dictionaries are *unordered*.

When we say that dictionaries are ordered, it means that the items have a defined order, and that order will not change.

Unordered means that the items do not have a defined order, you cannot refer to an item by using an index.

## Changeable

Dictionaries are changeable, meaning that we can change, add or remove items after the dictionary has been created.

## Duplicates Not Allowed

## Example

Duplicate values will overwrite existing values:

```
thisdict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964,  
    "year": 2020  
}  
print(thisdict)
```

[Try it Yourself »](#)

[REMOVE ADS](#)

## Dictionary Length

To determine how many items a dictionary has, use the `len()` function:

## Example

Print the number of items in the dictionary:

```
print(len(thisdict))
```

# Dictionary Items - Data Types

The values in dictionary items can be of any data type:

## Example

String, int, boolean, and list data types:

```
thisdict = {  
    "brand": "Ford",  
    "electric": False,  
    "year": 1964,  
    "colors": ["red", "white", "blue"]  
}
```

[Try it Yourself »](#)

## type()

From Python's perspective, dictionaries are defined as objects with the data type 'dict':

```
<class 'dict'>
```

Print the data type of a dictionary:

```
thisdict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}  
print(type(thisdict))
```

[Try it Yourself »](#)

## The dict() Constructor

It is also possible to use the [`dict\(\)`](#) constructor to make a dictionary.

### Example

Using the `dict()` method to make a dictionary:

```
thisdict = dict(name = "John", age = 36, country = "Norway")  
print(thisdict)
```

[Try it Yourself »](#)

## Python Collections (Arrays)

- **List** is a collection which is ordered and changeable. Allows duplicate members.
- **Tuple** is a collection which is ordered and unchangeable. Allows duplicate members.
- **Set** is a collection which is unordered, unchangeable\*, and unindexed. No duplicate members.
- **Dictionary** is a collection which is ordered\*\* and changeable. No duplicate members.

\*Set *items* are unchangeable, but you can remove and/or add items whenever you like.

\*\*As of Python version 3.7, dictionaries are *ordered*. In Python 3.6 and earlier, dictionaries are *unordered*.

When choosing a collection type, it is useful to understand the properties of that type. Choosing the right type for a particular data set could mean retention of meaning, and, it could mean an increase in efficiency or security.

## Exercise

Which one of these is a dictionary?

- x = ('apple', 'banana', 'cherry')
- x = {'type' : 'fruit', 'name' : 'banana'}
- x = ['apple', 'banana', 'cherry']

[Submit Answer »](#)

[PLUS](#)[SPACES](#)[GET CERTIFIED](#)[FOR TEACHERS](#)[FOR BUSINESS](#)[CONTACT US](#)

### Top Tutorials

- [HTML Tutorial](#)
- [CSS Tutorial](#)
- [JavaScript Tutorial](#)
- [How To Tutorial](#)
- [SQL Tutorial](#)
- [Python Tutorial](#)
- [W3.CSS Tutorial](#)
- [Bootstrap Tutorial](#)
- [PHP Tutorial](#)
- [Java Tutorial](#)
- [C++ Tutorial](#)
- [jQuery Tutorial](#)

### Top References

- [HTML Reference](#)
- [CSS Reference](#)
- [JavaScript Reference](#)
- [SQL Reference](#)
- [Python Reference](#)
- [W3.CSS Reference](#)
- [Bootstrap Reference](#)
- [PHP Reference](#)
- [HTML Colors](#)
- [Java Reference](#)
- [AngularJS Reference](#)
- [jQuery Reference](#)

### Top Examples

- [HTML Examples](#)
- [CSS Examples](#)
- [JavaScript Examples](#)
- [How To Examples](#)
- [SQL Examples](#)
- [Python Examples](#)
- [W3.CSS Examples](#)
- [Bootstrap Examples](#)
- [PHP Examples](#)
- [Java Examples](#)
- [XML Examples](#)
- [jQuery Examples](#)

### Get Certified

- [HTML Certificate](#)
- [CSS Certificate](#)
- [JavaScript Certificate](#)
- [Front End Certificate](#)
- [SQL Certificate](#)
- [Python Certificate](#)
- [PHP Certificate](#)
- [jQuery Certificate](#)
- [Java Certificate](#)
- [C++ Certificate](#)
- [C# Certificate](#)
- [XML Certificate](#)

[FORUM](#) [ABOUT](#) [ACADEMY](#)

W3Schools is optimized for learning and training. Examples might be simplified to improve reading and learning. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using W3Schools, you agree to have read and accepted our [terms of use](#), [cookies](#) and [privacy policy](#).

[Copyright 1999-2026 by Refsnes Data. All Rights Reserved.](#) [W3Schools is Powered by W3.CSS.](#)