

Fundamentals of Python

Review the fundamentals of Python with this short e-book. Check out the table of contents to navigate to each topic.

Contents

- 1: Variables and Strings
- 2: Integers, Floats, and Booleans
- 3: Lists
- 4: Tuples and Sets
- 5: Dictionaries

Variables

A variable is a name associated with a value. It can be named anything, but:

- It cannot start with a number.
- It cannot contain spaces.
- The only symbol it can contain is the underscore.

Correct

```
response
top500
one_apple
first_numbers
_usernames
_last_
current_
```

Incorrect

```
top 500
one apple
1st_numbers
^last_
$current
response#
```

Strings

A string is a sequence of characters surrounded by quotation marks: single quotes `' '` or double quotes `" "`

- Anything surrounded by the quotation marks indicate a string.
- Every opening quote must have a closing quote to form a string.
- If opening a string with a single quote, the closing quote must be a single quote as well.

Correct

```
"This is a sample string using double quotes"  
'This is a sample string using single quotes'  
'This is a normal string'  
'50 + 51'
```

Incorrect

```
'This string will end here.' From here the continuation is not a string'  
'This is an Incorrect string"  
"Same as this one'
```

Integers

An integer is a whole number of any length that can be positive or negative, written without a fractional element.

Examples

-1 0 1 86400

Floats

A float is a number that can be positive or negative written with a fractional element.

Examples

-100.54 1.0 59.1

Booleans

One of two values: `True` and `False`.

Extra Resources

1. [What are variables?](#)
2. [Strings and numbers](#)
3. [String formatting](#)
4. [Strings, Variables, and Getting Input from Users](#)

Lists

- An ordered, sequential data type.
- Used to store multiple elements in one variable.
- Defined by using a pair of square brackets.

```
shopping_list = ['cereals', 'milk', 'cherries']  
midterm_grades = [4, 9, 6, 6]
```

- Lists can contain different data types.

```
phone_numbers = ['james', 8067366796]
```

- Each element in a list has a position (index) through which it can be accessed.

```
shopping_list[0] # cereals  
shopping_list[1] # milk  
shopping_list[2] # cherries
```

Extra Resources

1. [What is a list?](#)
2. [Split, join, and slices](#)
3. [Extending Python lists](#)

Tuples

- Similar to lists, but are **immutable**.
- Defined by separating multiple values with commas.
- For better readability, it can be surrounded by parentheses (brackets).

```
shopping_list = 'apples', 'milk', 'cherries'  
midterm_grades = 4, 9, 6, 6  
phone_numbers = 'james', 8067366796  
better_readability = ('with', 'parenthesis')
```

Sets

- An unordered data type.
- Elements cannot be accessed by their indices.
- Defined by using a pair of curly braces.

```
shopping_list = {'apples', 'milk', 'cherries'}
```

Extra Resources

1. [Basic Python collections](#)
2. [Sets - 30 Days of Python](#)
3. [Python set operators](#)

Dictionaries

- Defined with 3 key components: curly braces, keys, and values.
- Keys must be strings or other hashable values.
- Values associated to each key can be anything.
- Dictionaries can be defined in one line:

```
employees = {'ID': 16915, 'name': 'James', 'department': ['Sales', 'Accounting']}
```

- Or in multiple lines to aid readability:

```
employees = {  
    'ID': 16915,  
    'name': 'James',  
    'department': ['Sales', 'Accounting']  
}
```

- Access values in a dictionary by using square brackets and the key:

```
employees['name'] # 'James'
```

Extra Resources

1. [What is a dictionary?](#)
2. [Updating Python dictionaries](#)