Python-1

April 11, 2019

1 Introduction to Python programming

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

```
web development (server-side),
software development,
mathematics,
system scripting.
```

What can Python do?

```
Python can be used on a server to create web applications.
Python can be used alongside software to create workflows.
Python can connect to database systems. It can also read and modify files.
Python can be used to handle big data and perform complex mathematics.
Python can be used for rapid prototyping, or for production-ready software development.
```

Why Python?

```
Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
Python has a simple syntax similar to the English language.
Python has syntax that allows developers to write programs with fewer lines than some
    other programming languages.
Python runs on an interpreter system, meaning that code can be executed as soon as it
    is written. This means that prototyping can be very quick.
Python can be treated in a procedural way, an object-orientated way or a functional way.
```

Good to know

```
The most recent major version of Python is Python 3, which we shall be using in this
    tutorial. However, Python 2, although not being updated with anything other than
    security updates, is still quite popular.
In this tutorial Python will be written in a text editor. It is possible to write
    Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans
    or Eclipse which are particularly useful when managing larger collections of
    Python files.
```

Python Syntax compared to other programming languages

Python was designed for readability, and has some similarities to the English language with influence from mathematics. Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.

Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

2 Variables

1. Keywords
2. Identifiers
    1. How to make Identifiers
3. Assigning values to variables
    1. Single values
    2. Multiple Values
4. Literals
    1. Numbers
       1.1 Integer 1.2 Float 1.3 Complex
    2. Boolean
    3. Special
    4. Strings
    5. Collections

3 Debugging

1. Syntax Errors
2. Runtime Errors
3. Semantic Errors

4 Operators

1. Arithmetic operators

1.1 Addition
1.2 Subtraction
1.3 Multiplication
1.4 Division
1.5 Modulus
1.6 Exponentiation
1.7 Floor division

2. Comparison operators

2.1 ==
2.2 !=
2.3 >
2.4 <
2.5 >=
2.6 <=

3. Logical operators

3.1 and
3.2 or
3.3 not

4. Bitwise operators

4.1 &
4.2 |
4.3 ^
4.4 ~
4.5 <<
4.6 >>

5. Assignment operators

5.1 =
5.2 +=
5.3 -=
5.4 *=
5.5 /=
5.6 %=
5.7 **=
5.8 //=
5.9 &=
5.10 |=
5.11 ^=
5.12 >>=
5.13 <<=

6. Identity operators

```
6.1 is
6.2 is not
```

7. Membership operators

```
7.1 in
7.2 not in
```

5 Data Structures and Methods

1. Strings

Place between ′ ′ or " “. in some cases″"" """ which is called DocString
Can be called in both individual and slice
Immutable
Support formatting
Useful methods:

```
1.1 capitalize
1.2 center
1.3 count
1.4 endswith
1.5 find
1.6 index
1.7 lower
1.8 upper
```

2. Lists

Place between [ ]
Can be called in both individual and slice
Mutable
Useful methods:

```
2.1 append
2.2 extend
2.3 insert
2.4 remove
2.5 index
2.6 count
2.7 reverse
2.8 copy
2.9 clear
2.10 sort
```

3. Tuples

Place between (_)
Can be called in both individual and slice
Immutable
Useful methods:

3.1 `count`
3.2 `index`

4. Dictionary

Place between {_}
can be called in individual mode
Mutable
Useful methods:

4.1 `copy`
4.2 `clear`
4.3 `keys`
4.4 `pop`
4.5 `values`
4.6 `items`
4.7 `get`
4.8 `setdefault`

5. Sets

Place between {_}
DO not accept calls
Immutable
Useful methods:

5.1 `copy`
5.2 `clear`
5.3 `add`
5.4 `remove`
5.5 `discard`
5.6 `pop`
5.7 `update`
5.8 `isdisjoint`
5.9 `issubset`
5.10 `issuperset`
5.11 `union`
5.12 `intersection`
5.13 `difference`
5.14 `symmetric_difference`

Note: Some operators can not be used between different data types!

6 Python I/O

Simple Input and output functions in python

1. print
2. input

7 Statement, Indentation and Comments

1. Statements
2. Indentation
3. Comments
4. Doc String