
Python Overview

— Besant Technologies —

Why Python?

- Simplicity
- High-level language
- Interpreted
- Interactive
- Object-oriented
- Extensible (Can be merged with c, c++)
- Libraries (numpy, scipy etc.,)

* Current python version - 3.7.4

Variables:

- Just like a container.
- Used for storing values.
- Values can be changed.
- Adv: No need to define type of variable.
- Memory efficient.
- Garbage collection.

Operators:

- Arithmetic: $+$, $-$, $*$, $/$, $//$, $\%$, $**$
- Assignment: $=$
- Relational: $==$, $!=$, $<$, $>$, $<=$, $>=$
- Logical: and , or , not
- Bitwise: $\&$, $|$, \wedge , $<<$, $>>$, \sim

Datatypes:

- None
- Integer
- Float
- Complex numbers
- Boolean
- Strings

Some more datatypes..

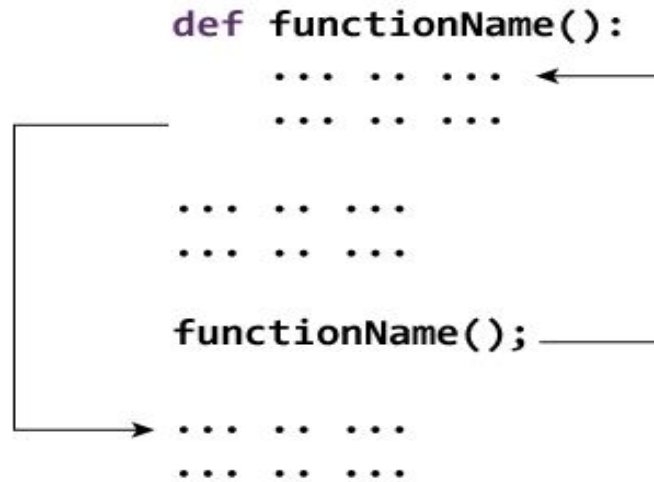
- Lists
- Tuples
- Set
- Dictionary

Conditionals and loops:

- If condition
- If-elif-...else
- For loop
- While loop
- Break
- Continue
- pass

Functions:

How Function works in Python?



Argument types:

- Formal arguments
- Default arguments
- Keyword arguments
- Variable length arguments
- Variable length keyword arguments

Lambda functions:

- Syntax: lambda arguments: expression
- Can be assigned to a variable.
- Simple programs with single line executions.

Special functions:

- Filter, map, reduce.

Modules:

- Module means a separate file.
- How to call functions from different files.
- `Import <file>`
- `From <file> import <function>`
- `From <file> import *`
- `Import <file> as <different name>`
- `dir()` - getting the functions in a file.
- `__init__.py` file
- `__name__ == __main__`

Opening & Closing a file:

- `open(<path to file>, <mode>)`
- Modes:
 - 'r' - read mode
 - 'w' - write mode
 - 'a' - append mode
 - 'r+' - reading and writing
 - 'w+' - writing and reading
 - 'Wb', 'rb' - for reading and writing binary files

* `with` statement

* `close()`

Reading & Writing:

- `f.read()`
- `f.read(<number of characters>)`
- `f.readline()`
- `f.readlines()`
- `for line in f:`
- `f.write()`
- `f.writelines()`

JSON Files:

- .json extension.
- Import json
- Independent of platform
- `json.dump(<python_variable>, file_object)`
- `<python_variable> = json.load(file_object)`