

Python Basics

Basic Stuff

Printing

- Print and go to next line: `print(message)`
- Print and stay in same line: `print(message, end='')`
- Print more than one value with custom separator:
`print(value1, value2, ..., sep='separator')`

Values and Variables

- Create a variable: `variableName = value`
 - Dynamic data typing takes place
- Get datatype of a value: `type(value)`
 - Returns type name in the format <class 'type'>
- Find memory address of a variable: `id(variable)`
- Delete a variable from memory: `del variable`
- Get number of times an variable is referenced:

```
import sys
sys.getrefcount(value/variable)
```

Importing

- Import a module into the code:
`import module`
- Import a module with an alias name:
`import module as alias`
- Import only certain stuff from a module:
`from module import stuff1, stuff2`
- Import everything from a module:
`from module import *`

Comments

- Single line comment: `# This is a comment`
- Multi-line comment:

```
'''                               '''
This is a comment               or   This is a comment
'''                               '''
```

Identifiers

- Identifiers are names given to variables, classes and other structures
- Identifier naming rules:
 - Can be of any length
 - Case-sensitive (alpha and ALPHA are different)
 - Can contain letters, numbers and underscore(_)
 - Should not be a keyword

Keywords

Reserved words that have a special meaning in Python. Cannot be used as identifiers.

Eg: if, and, while, for, else, try, except etc

Data Types

Primitive Data Types

- **int** - Usual numbers, both positive and negative. No decimals or commas.
e.g.: 1, 56, -69, 1048576
- **float** - all positive and negative real numbers. Can be expressed in basic form (1.256) or exponential form (1256E-3)
e.g.:
Basic form: 1.56, -420.15, 12345.67898
Exponential form: aEb indicates $a \times 10^b$: 21.5E2
67.8E-97, 6.023E23

- **bool** - Only one of the two values – **True** or **False**. Represent truth or falsehood of something.

- **str** - Any kinds of characters, including letters, numbers, symbols and letters from other languages, enclosed in a pair of single (' ') or double (" ") quotes.

e.g.: "Datte Kimi_yowaimo123!@# \$"

- **complex** - Any complex number in the form of $a \pm bj$, where $j = \sqrt{-1}$

e.g.: 12-8j, -5-6j, 37+8j

Expression: A series of tokens which equate to a value.

e.g.: 1+6/2 (gives 4)

Statement: A series of tokens which performs some action, and doesn't return a value.

e.g.: `print('Hello!')`

Type casting

The forceful conversion of one data type to another.

To convert a value of one data type to another one, use:

- `int(value)` for int datatype
 - `float(value)` for float datatype
- and so on.

The **value** in the brackets should be compatible with the datatype being converted to. Else, it shows an error.