Contents

[1 Some Python related concepts / topics 2](#_Toc528960740)

[1.1 The Python Debugger 2](#_Toc528960741)

[1.2 Everything is an object (mutable or immutable) in Python 2](#_Toc528960742)

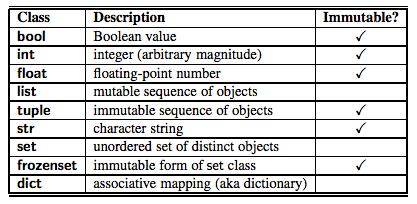
[1.3 Indentation in Python 5](#_Toc528960743)

[1.4 Iteration in Python 6](#_Toc528960744)

# Some Python related concepts / topics

## The Python Debugger

## Everything is an object (mutable or immutable) in Python



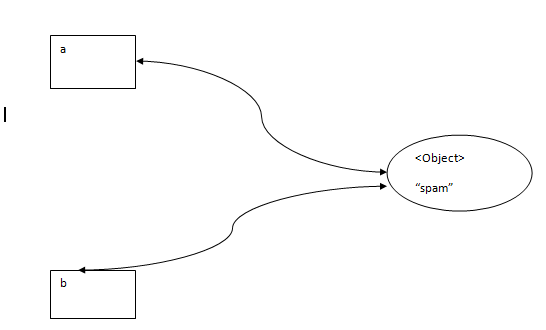
One aspect that deserves in-depth explanation is variable assignment.

When we assign a value to a variable:

<name> = <object>

We are actually **binding** a **name** to an **object.**One implication of this is that multiple names can be bound to a single object.

|  |
| --- |
| #py\_everything\_is\_an\_obj\_001.py  a = "spam"  b = "spam"  #  print(id(a))  print(id(b))  #  # id() returns the actual memory location where the variable is stored.  # Since id(a) = id(b), we know that a and b both point to a single variable,  # that resides in a single memory location.  # This is what we mean by “multiple names bound to single object  print(a is b) |

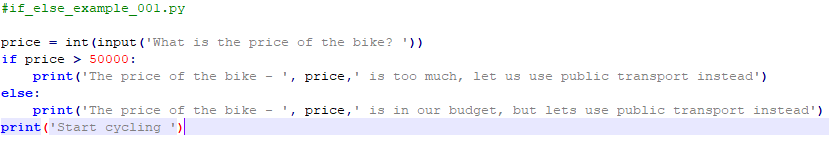


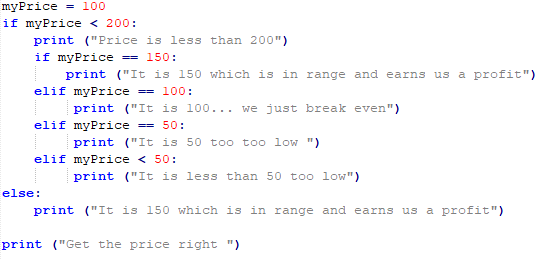
|  |
| --- |
| #py\_everything\_is\_an\_obj\_001.py  a = "spam"  b = "spam"  #  print(id(a))  print(id(b))  #  # id() returns the actual memory location where the variable is stored.  # Since id(a) = id(b), we know that a and b both point to a single variable,  # that resides in a single memory location.  # This is what we mean by “multiple names bound to single object  print(a is b) |

a

b

## Indentation in Python





How many spaces should you indent? Python requires at least one, some programmers consistently use two, four is the most popular number, but some prefer a more dramatic display and use eight. A four space indentation for a block is the recommended Python style. This text uses the recommended four spaces to set off each enclosed block. In most programming editors you can set the Tab key to insert spaces automatically so you need not count the spaces as you type. Whichever indent distance you choose, you must use this same distance consistently throughout a Python program.

Why is indentation that mixes tabs and spaces a problem and thus forbidden in Python 3? Consider creating a Python source file in one editor and then viewing it in a different editor with tab stops set differently. Lines that appear perfectly indented in the original editor would be misaligned in the new editor. Instead, code indented with four spaces within one editor would appear exactly the same in any other editor. Python 3 does allow the use of tabs for indentation—you just cannot mix them with spaces within the same source file.

## Iteration in Python