

# 2.6 Python Naming Conventions

---

## General

---

- Avoid using names that are too general or too wordy. Strike a good balance between the two.
- Bad: `data_structure`, `my_list`, `info_map`, `dictionary_for_the_purpose_of_storing_data_representing_word_definitions`
- Good: `user_profile`, `menu_options`, `word_definitions`
- Don't be a jackass and name things "O", "I", or "l"
- When using CamelCase names, capitalize all letters of an abbreviation (e.g. `HTTPServer`)

## Packages

---

- Package names should be all lower case
- When multiple words are needed, an underscore should separate them
- It is usually preferable to stick to 1 word names

## Modules

---

- Module names should be all lower case
- When multiple words are needed, an underscore should separate them
- It is usually preferable to stick to 1 word names

## Classes

---

- Class names should follow the UpperCaseCamelCase convention
- Python's built-in classes, however are typically lowercase words
- Exception classes should end in "Error"

# Global (module-level) Variables

---

- Global variables should be all lowercase
- Words in a global variable name should be separated by an underscore

## Instance Variables

---

- Instance variable names should be all lower case
- Words in an instance variable name should be separated by an underscore
- Non-public instance variables should begin with a single underscore
- If an instance name needs to be mangled, two underscores may begin its name

## Methods

---

- Method names should be all lower case
- Words in an method name should be separated by an underscore
- Non-public method should begin with a single underscore
- If a method name needs to be mangled, two underscores may begin its name

## Method Arguments

---

- Instance methods should have their first argument named 'self'.
- Class methods should have their first argument named 'cls'

## Functions

---

- Function names should be all lower case
- Words in a function name should be separated by an underscore

## Constants

---

- Constant names must be fully capitalized
- Words in a constant name should be separated by an underscore

## Meaning of underscores

---

Single or double underscores at the beginning and/or at the end of names have special meanings in Python:

- `_single_leading_underscore` : Weak “internal use” indicator. For example, `from M import *` does not import objects whose name starts with an underscore.
- `single_trailing_underscore_` : Used by convention to avoid conflicts with Python keyword, e.g. `class_` instead of `class`.
- `__double_leading_underscore` : When naming a class attribute, invokes name mangling (inside class `FooBar`, `__boo` becomes `_FooBar__boo` )
- `__double_leading_and_trailing_underscore__` : “Magic” objects or attributes that live in user-controlled namespaces. E.g. `__init__`, `__import__` or `__file__` . Never invent such names; only use them as documented.