

Cheatsheet: Python Coding Practices and Packaging Concepts



Estimated time needed: 5 minutes

Package/Method	Description	Code Example
Packaging	To create a package, the folder structure is as follows:	
	<ol style="list-style-type: none"> Project folder → Package name → init.py, module_1.py, module_2.py, and so on. In the init.py file, add code to reference the modules in the package. 	<pre> module1.py def function_1(arg): return <operation output> init.py: from . import function_1 </pre>
Python Style Guide	<ul style="list-style-type: none"> Four spaces for indentation Use blank lines to separate functions and classes Use spaces around operators and after commas 	<pre> def function_1(a, b): if a > b: c = c + 5 else: c = c - 3 return c ... </pre>

Package/Method	Description	Code Example
	<ul style="list-style-type: none"> Add larger blocks of code inside functions 	<pre>c = function_1(a, b)</pre> <p>Constant Definition example</p>
	<ul style="list-style-type: none"> Name functions and files using lowercase with underscores 	<pre>MAX_FILE_UPLOAD_SIZE</pre>
	<ul style="list-style-type: none"> Name classes using CamelCase 	
	<ul style="list-style-type: none"> Name constants in capital letters with underscores separating words 	

Static Code Analysis

Use Static code analysis method to evaluate your code against a predefined style and standard without executing the code.

Shell code:

```
pylint code.py
```

For example, use Pylint to perform static code analysis.

Unit Testing

Unit testing is a method to validate if units of code are operating as designed.

```
import unittest
```

```
from mypackage.mymodule import my_function
```

During code development, each unit is tested. The unit is tested in a continuous integration/continuous delivery test server environment.

```
class
TestMyFunction(unittest.TestCase):

    def test_my_function(self):

        result =
my_function(<args>)

        self.assertEqual(result,
<response>)
```

Package/Method	Description	Code Example
	You can use different test functions to build unit tests and review the unit test output to determine if the test passed or failed.	<code>unittest.main()</code>

Author(s)

Abhishek Gagneja

Other Contributor(s)

Andrew Pfeiffer, Sina Nazeri

Changelog

Date	Version	Changed by	Change Description
2023-07-06	0.1	Abhishek Gagneja	Initial version created
2023-07-10	0.2	Andrew Pfeiffer	Documented in AWB

© IBM Corporation 2023. All rights reserved.