

# Cheatsheet: Python Coding Practices and Packaging Concepts



**Skills**  
Network

**Estimated time needed:** 5 minutes

Package/ Method	Description	Code Example
Packaging	To create a package, the folder structure is as follows:	<b>module1.py</b>
	1. <b>Project folder</b> → <b>Package name</b> → <b>init.py</b> , <b>module_1.py</b> , <b>module_2.py</b> , and so on.	<pre>def function_1(arg):     return &lt;operation output&gt;</pre>
Python Style Guide	2. In the <b>init.py</b> file, add code to reference the modules in the package.	<b>init.py:</b> <pre>from . import function_1</pre>
	<ul style="list-style-type: none"><li>• Four spaces for indentation</li><li>• Use blank lines to separate functions and classes</li><li>• Use spaces around operators and after commas</li><li>• Add larger blocks of code inside functions</li><li>• Name functions and files using lowercase with underscores</li><li>• Name classes using CamelCase</li><li>• Name constants in capital letters with underscores separating words</li></ul>	<pre>def function_1(a, b):     if a &gt; b:         c = c + 5     else:         c = c - 3     return c ... c = function_1(a, b)</pre>
Static Code Analysis		<b>Constant Definition example</b> <pre>MAX_FILE_UPLOAD_SIZE</pre>
	Use Static code analysis method to evaluate your code against a predefined style and standard without executing the code.	<b>Shell code:</b> <pre>pylint code.py</pre>
	For example, use Pylint to perform static code analysis.	

Package/ Method	Description	Code Example
Unit Testing		<pre>import unittest</pre>
	Unit testing is a method to validate if units of code are operating as designed.	<pre>from mypackage.mymodule import my_function</pre>
	During code development, each unit is tested. The unit is tested in a continuous integration/continuous delivery test server environment.	<pre>class TestMyFunction(unittest.TestCase):      def test_my_function(self):          result = my_function(&lt;args&gt;)          self.assertEqual(result,         &lt;response&gt;)  unittest.main()</pre>
	You can use different test functions to build unit tests and review the unit test output to determine if the test passed or failed.	

## 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.