

# Cheatsheet: Python Coding Practices and Packaging Concepts



**Skills**  
Network

**Estimated time needed:** 5 minutes

Package/Method	Description	Code Example
Packaging	<p>To create a package, the folder structure is as follows:</p> <ol style="list-style-type: none"> <li>1. Project folder → Package name → <code>__init__.py</code>, <code>module_1.py</code>, <code>module_2.py</code>, and so on.</li> <li>2. In the <code>__init__.py</code> file, add code to reference the modules in the package.</li> </ol>	<b>module1.py</b> <pre>def function_1(arg):     return &lt;operation output&gt;</pre> <b>__init__.py:</b> <pre>from . import function_1</pre>
Python Style Guide	<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     ... c = function_1(a, b)</pre> <p><b>Constant Definition example</b></p> <pre>MAX_FILE_UPLOAD_SIZE</pre>
Static Code Analysis	Use Static code analysis method to evaluate your code against a predefined style and standard without executing the code.	<p><b>Shell code:</b></p> <pre>pylint code.py</pre>

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

## Author(s)

Abhishek Gagneja

## Other Contributor(s)

Andrew Pfeiffer, Sina Nazeri