

# Object-Oriented Syntax for Python

Creating a class	<code>class ClassIdentifier(object):</code>
Attributes <ul style="list-style-type: none"> <li>✓ Public</li> <li>✓ Private</li> <li>✓ Protected</li> </ul>	<code>self.attributeIdentifier=12</code> <code>self.__attributeIdentifier=12</code> <code>self._attributeIdentifier=12</code>
Defining a Method <ul style="list-style-type: none"> <li>✓ Public</li> <li>✓ Protected</li> <li>✓ Private</li> </ul>	<code>def methodIdentifier(self):</code> <code>def _methodIdentifier(self):</code> <code>def __methodIdentifier(self):</code>
Defining a constructor method	<code>def __init__(self):</code>
Instantiation	<code>object=ClassIdentifier(object)</code>
Calling a method (outside class)	<code>object.methodIdentifier()</code>
Calling a method (inside class by convention)	<code>self.methodIdentifier()</code>
Inheritance	<code>Class SubclassIdentifier(BaseclassIdentifier)</code>
Composition association	<code>def methodIdentifier(self):</code> <code>self.objectIdentifier=ClassIdentifier()</code>
Call composition method	<code>objectIdentifier=SubclassIdentifier()</code> <code>object.attributeIdentifier.classMethodIdentifier()</code>
Aggregation association	<code>def methodIdentifier(self,objectIdentifier):</code> <code>self.objectIdentifier=objectIdentifier</code>



Call Aggregation method	<pre>objectIdentifier=ClassIdentifier() obj2=SubclassIdentifier(objectIdentifier) obj2.attributeIdentifier.methodFromClassIdentifier()</pre>
Static Method	<pre>class ClassIdentifier(object):     @staticmethod     def staticMethodIdentifier():         pass ClassIdentifier.staticMethodIdentifier()</pre>
Class attributes ✓ Public ✓ Private ✓ Protected	<pre>attributeIdentifier=12 __attributeIdentifier=12 _attributeIdentifier=12</pre>