

# Appendixes

## Appendix A: Java/Groovy<sup>1</sup>

Feature	Java	Groovy
Public class	<code>public class</code>	<code>class</code>
Loops	<code>for(Type it : c){...}</code>	<code>c.each {...}</code>
Lists	<code>List list = asList(1,2,3);</code>	<code>def list = [1,2,3]</code>
Maps	<code>Map m = ...; m.put(x,y);</code>	<code>def m = [x: y]</code>
Function def.	<code>void method(Type t) {}</code>	<code>def method(t) {}</code>
Mutable value	<code>Type t</code>	<code>def t</code>
Immutable value	<code>final Type t</code>	<code>final t</code>
Null safety	<code>(x == null ? null : x.y)</code>	<code>x?.y</code>
Null replacement	<code>(x == null ? "y" : x)</code>	<code>x ?: "y"</code>
Sort	<code>Collections.sort(list)</code>	<code>list.sort()</code>
Wildcard import	<code>import java.util.*;</code>	<code>import java.util.*</code>
Var-args	<code>(String... args)</code>	<code>(String... args)</code>
Type parameters	<code>Class&lt;T&gt;</code>	<code>Class&lt;T&gt;</code>
Concurrency	<code>Fork/Join</code>	<code>GPars</code>

<sup>1</sup>Version 1.3 of this cheat sheet.

## No Java Analog

Feature	Groovy
Default closure arg.	it
Default value	def method(t = "yes")
Add method to object	t.metaClass.method = {}
Auto-delegate	@Delegate
Extension methods	Categories
Rename import	import java.util.Vector as Vect

## Tricks

Feature	Groovy
Range	def range = [a..z]
Slice	def slice = list[0..3]
<< Operator	list << addMeToList
Cast operation	def dog = [name: "Fido", speak:{println "woof"}] as Dog
GString	def gString = "Dog's name is \${dog.name}"