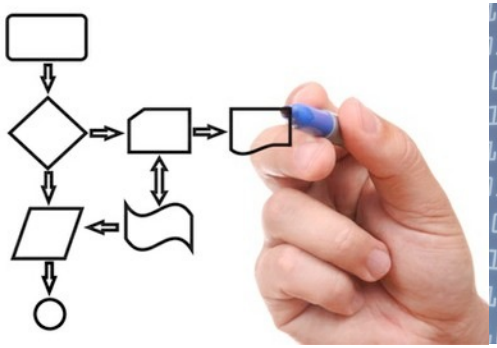
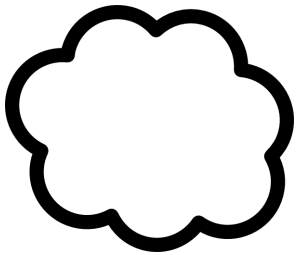
San Jose State University

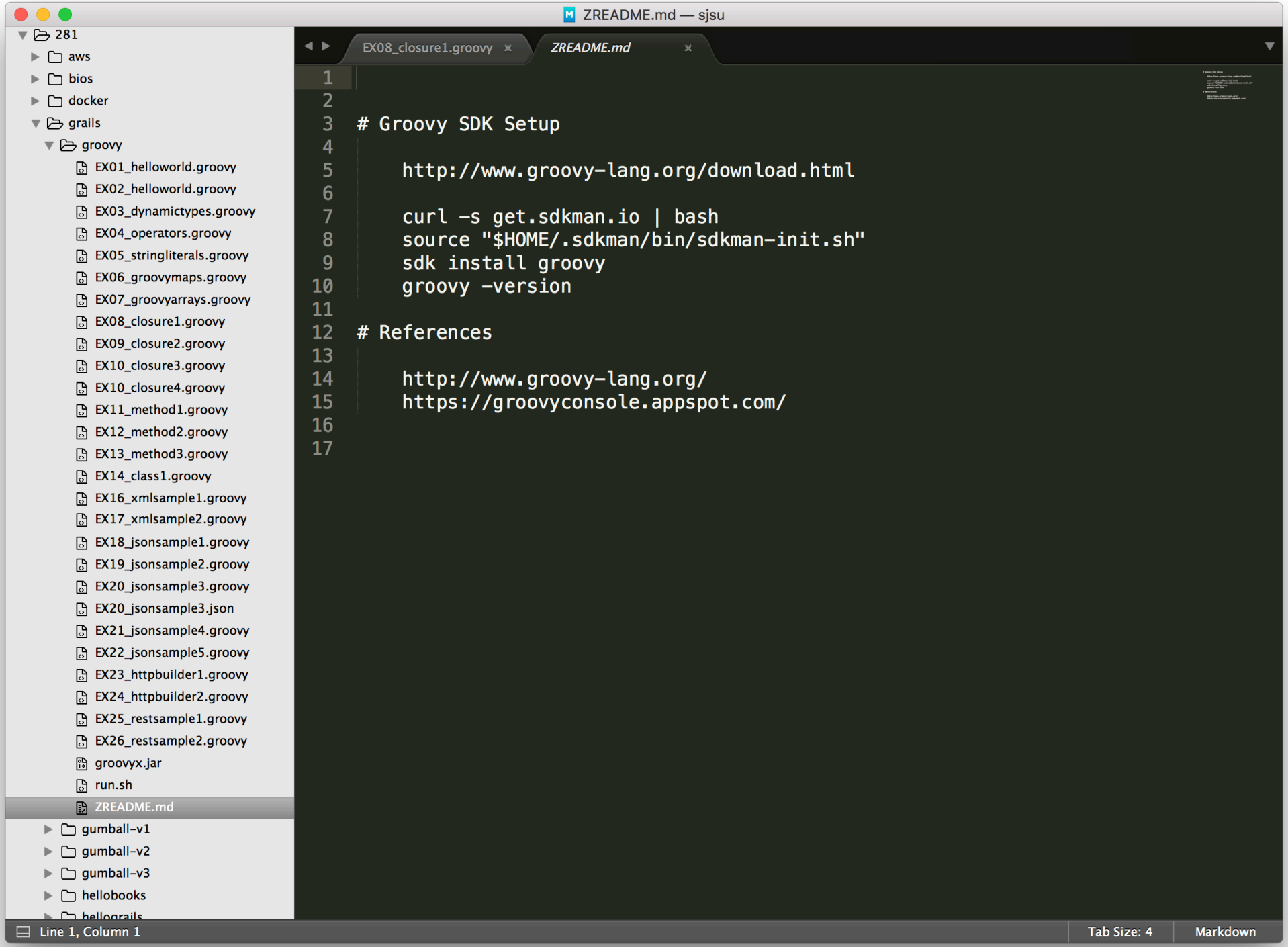
CMPE 281

Cloud Technologies

Quick Introduction to Groovy

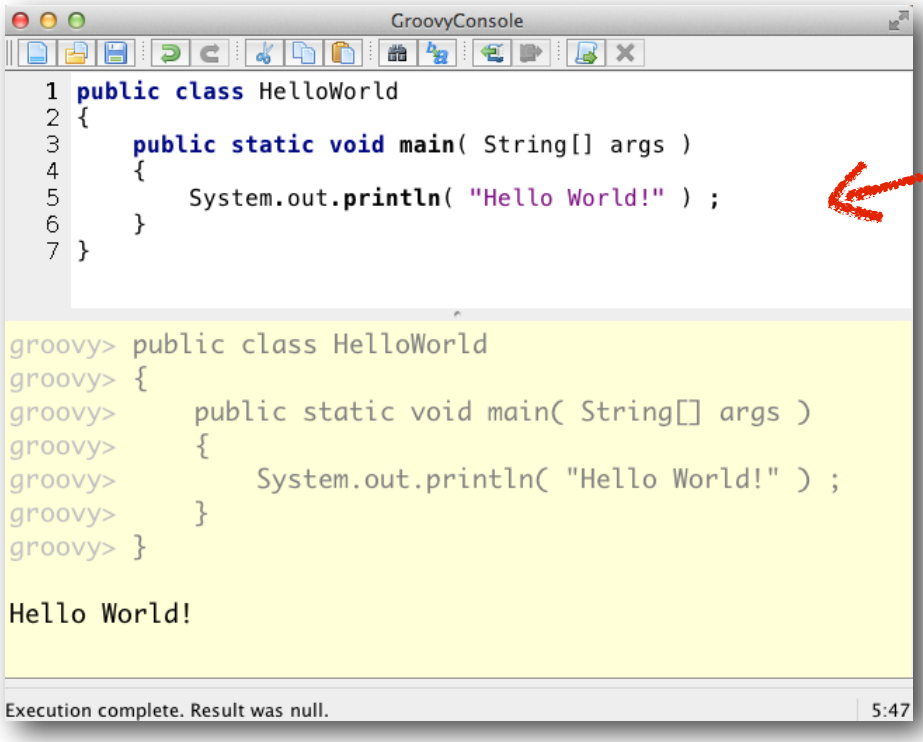




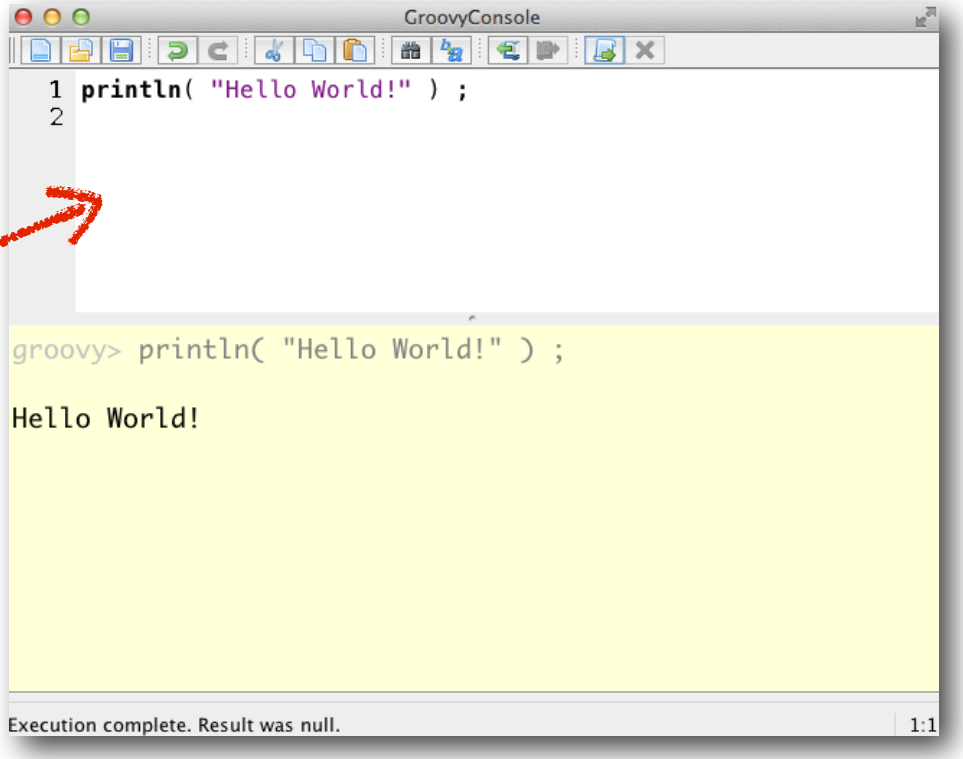

ZREADME.md — sjsu
281
aws
bios
docker
grails
groovy
EX01_helloworld.groovy
EX02_helloworld.groovy
EX03_dynamictypes.groovy
EX04_operators.groovy
EX05_stringliterals.groovy
EX06_groovymaps.groovy
EX07_groovyarrays.groovy
EX08_closure1.groovy
EX09_closure2.groovy
EX10_closure3.groovy
EX10_closure4.groovy
EX11_method1.groovy
EX12_method2.groovy
EX13_method3.groovy
EX14_class1.groovy
EX16_xmlsample1.groovy
EX17_xmlsample2.groovy
EX18_jsonsample1.groovy
EX19_jsonsample2.groovy
EX20_jsonsample3.groovy
EX20_jsonsample3.json
EX21_jsonsample4.groovy
EX22_jsonsample5.groovy
EX23_httpbuilder1.groovy
EX24_httpbuilder2.groovy
EX25_restsample1.groovy
EX26_restsample2.groovy
groovyx.jar
run.sh
ZREADME.md
gumball-v1
gumball-v2
gumball-v3
hellobooks
hellograils
EX08_closure1.groovy
ZREADME.md
Groovy SDK Setup
http://www.groovy-lang.org/download.html
curl -s get.sdkman.io | bash
source "$HOME/.sdkman/bin/sdkman-init.sh"
sdk install groovy
groovy -version
References
http://www.groovy-lang.org/
https://groovyconsole.appspot.com/
Line 1, Column 1
Tab Size: 4
Markdown

Groovy

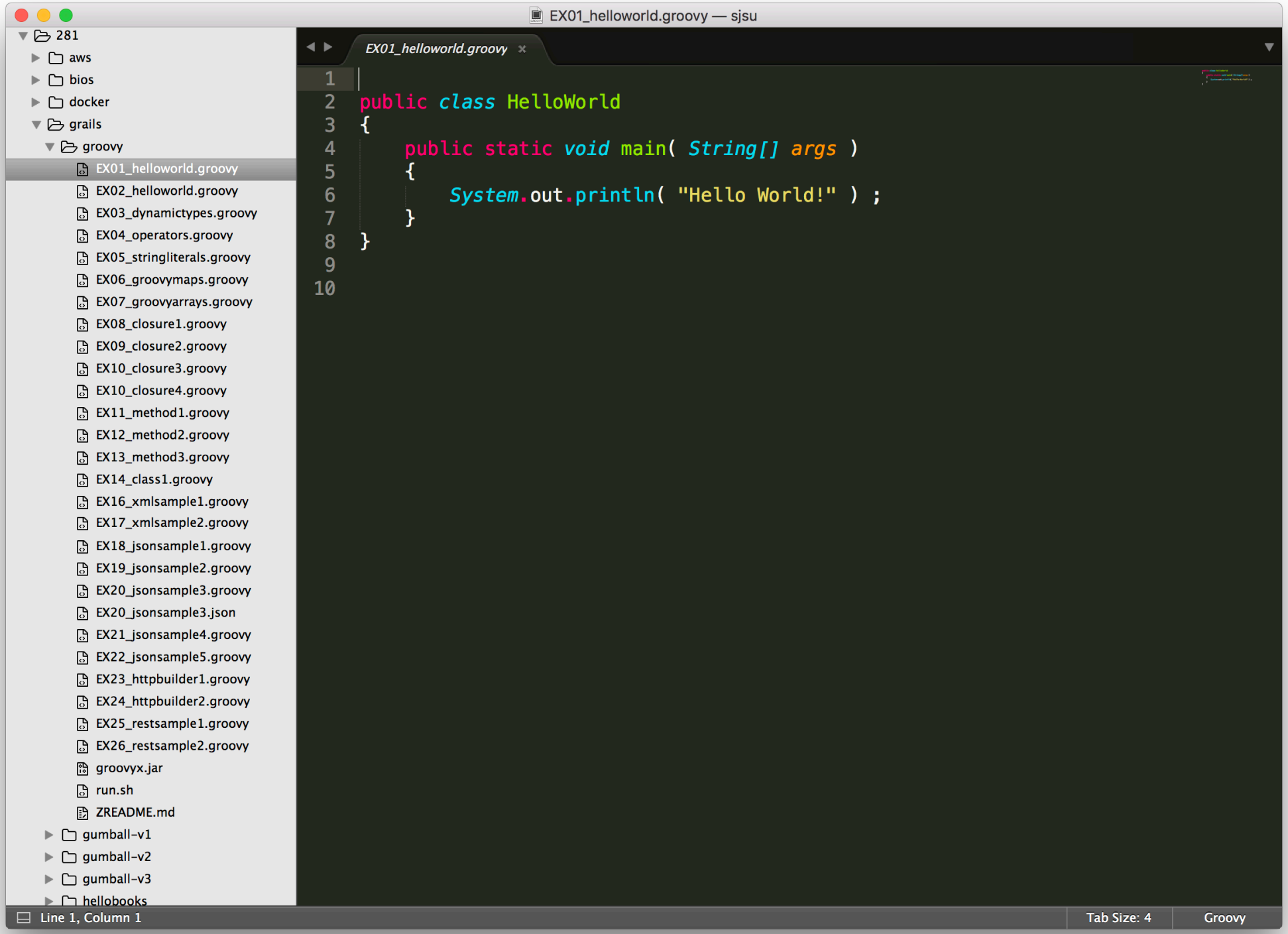
Hello World

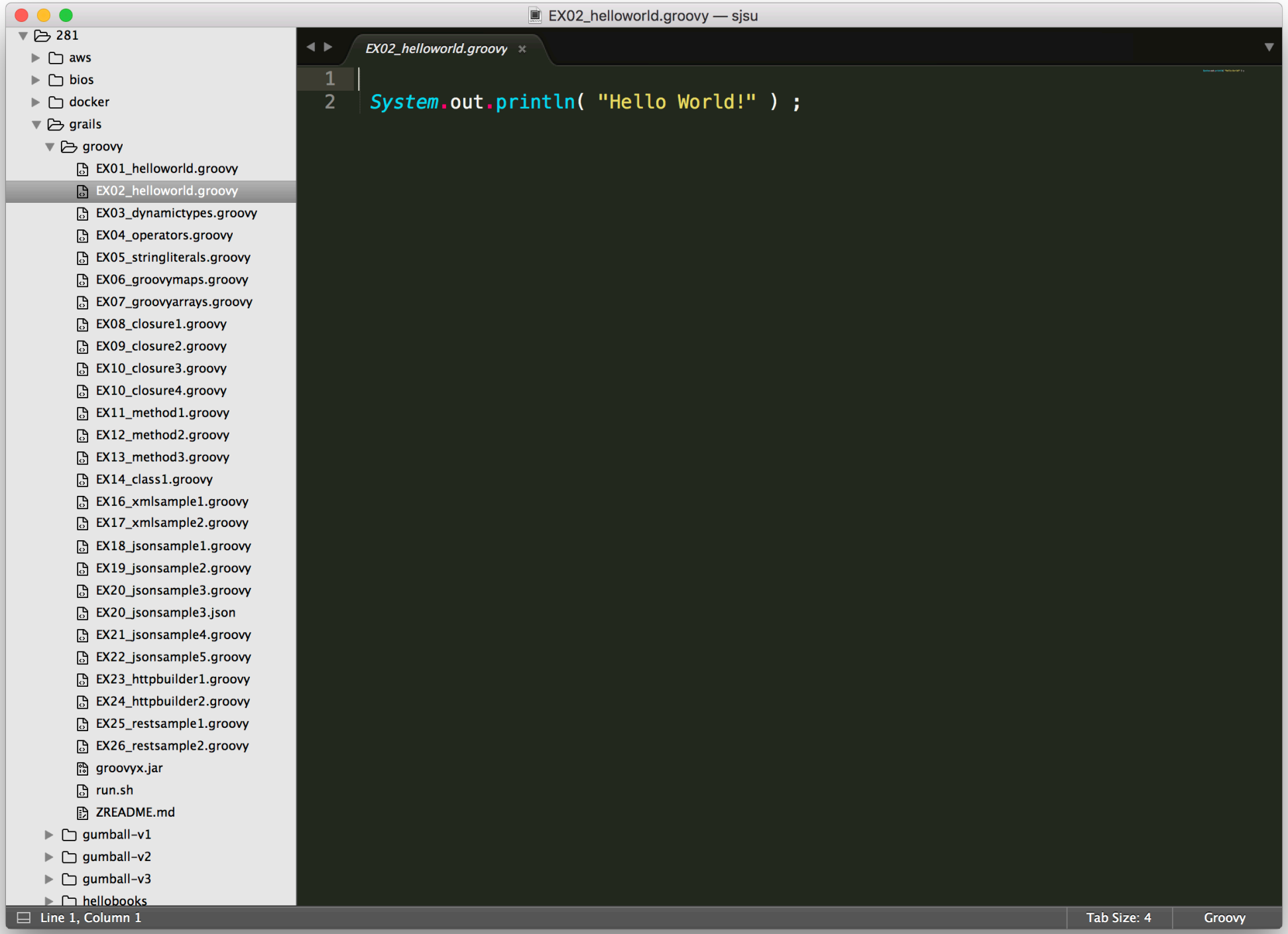


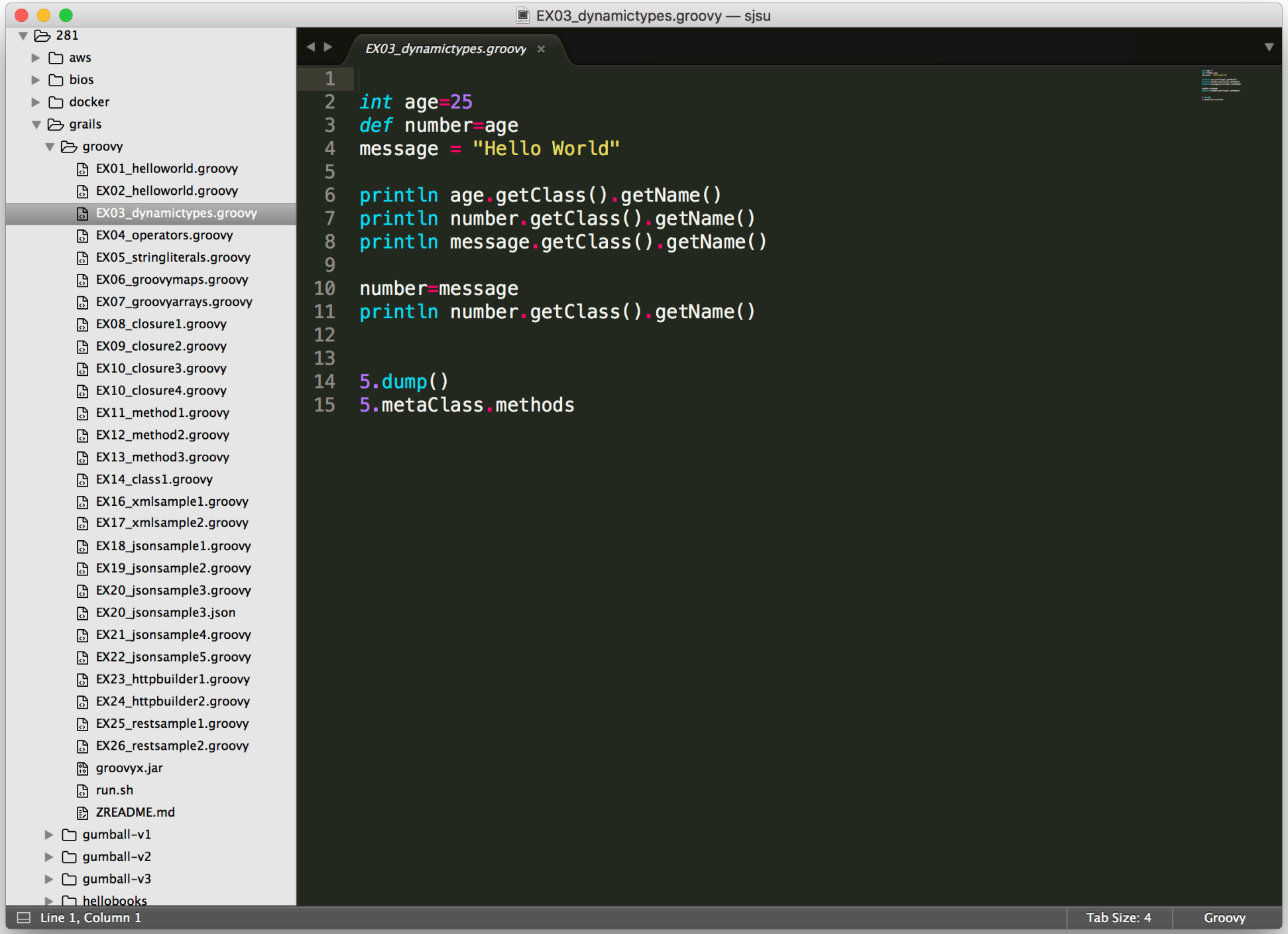
Groovy is "Java"

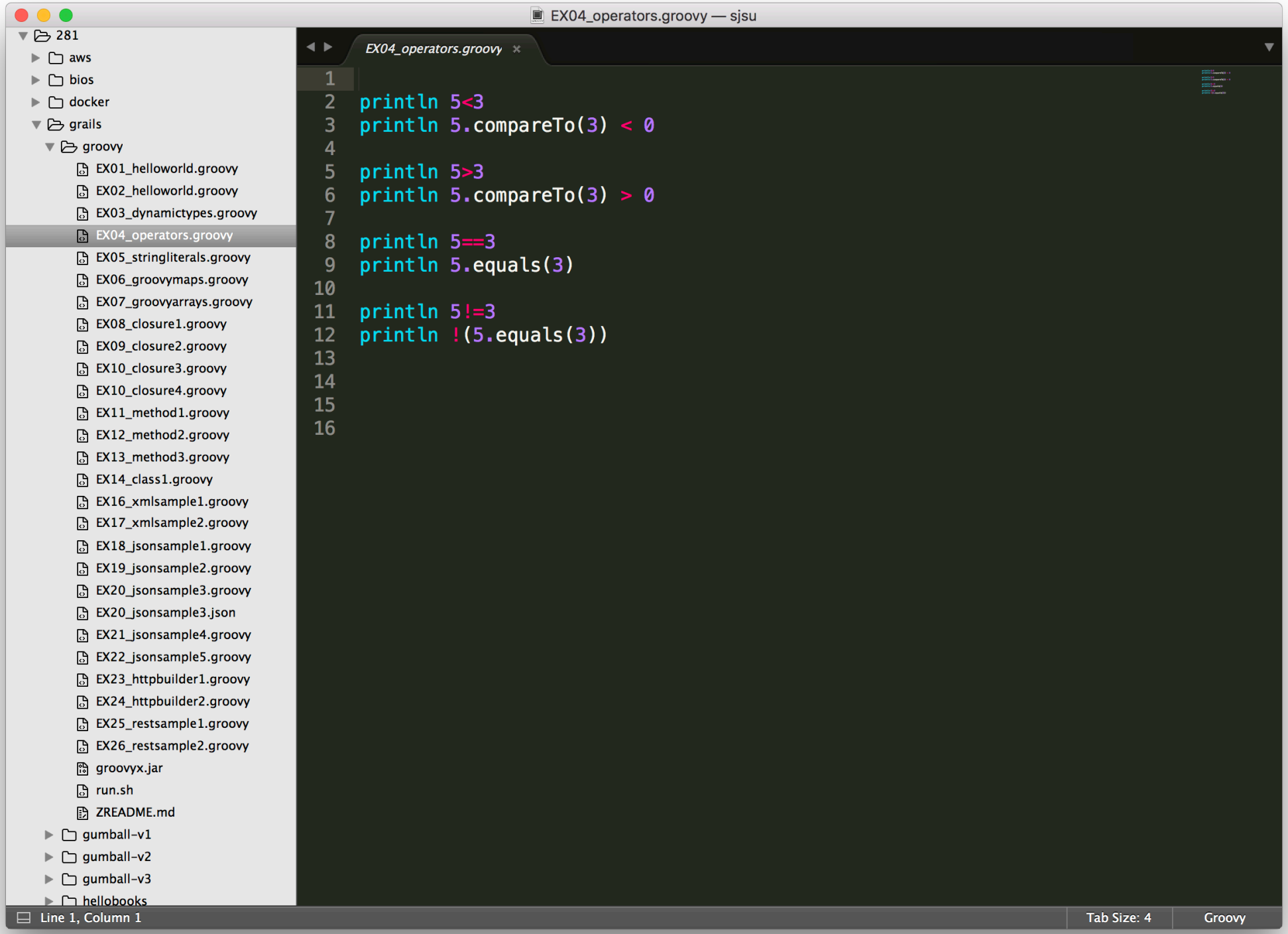


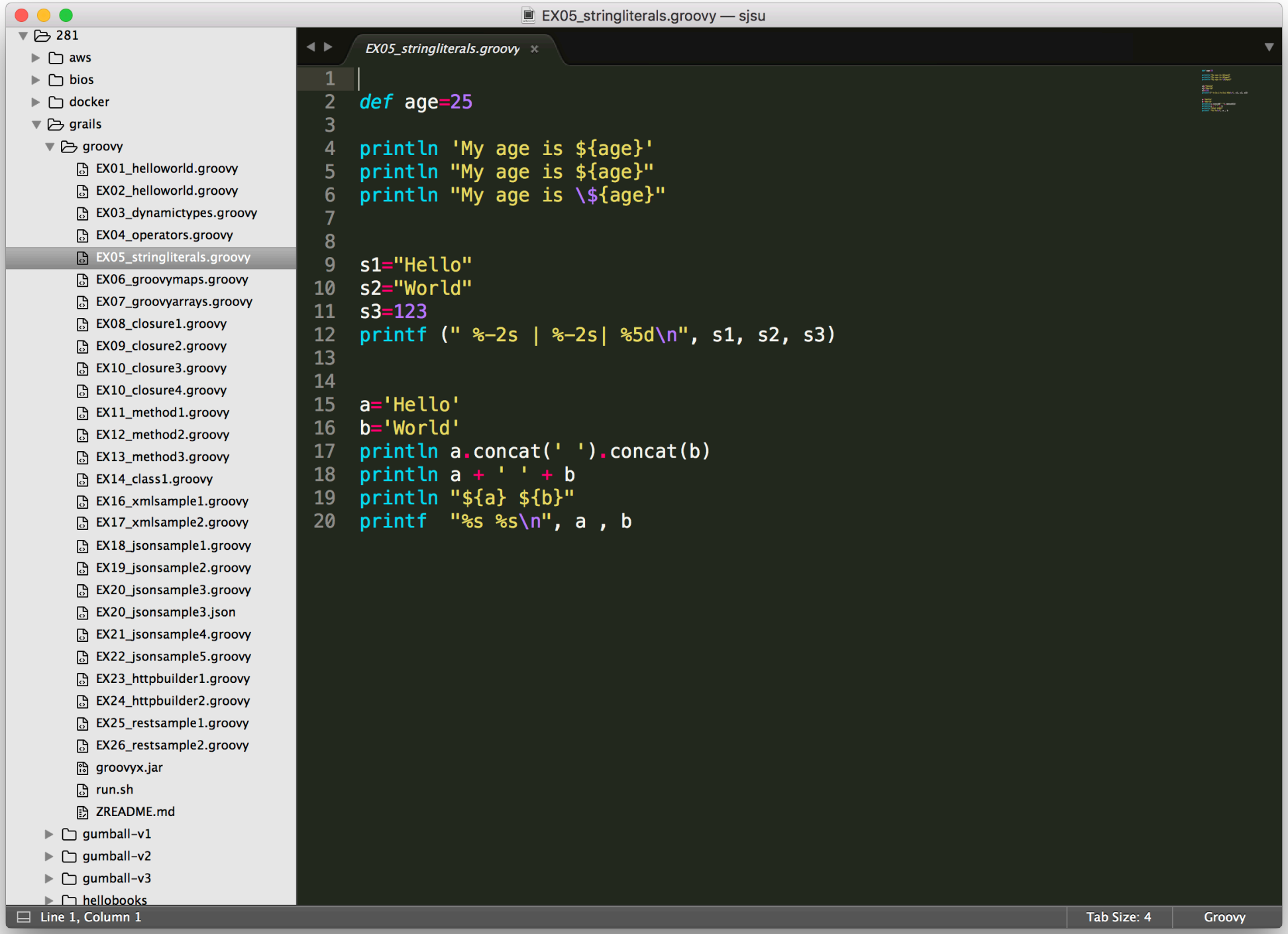
Yes, Groovy is "Java", but without the ceremony!

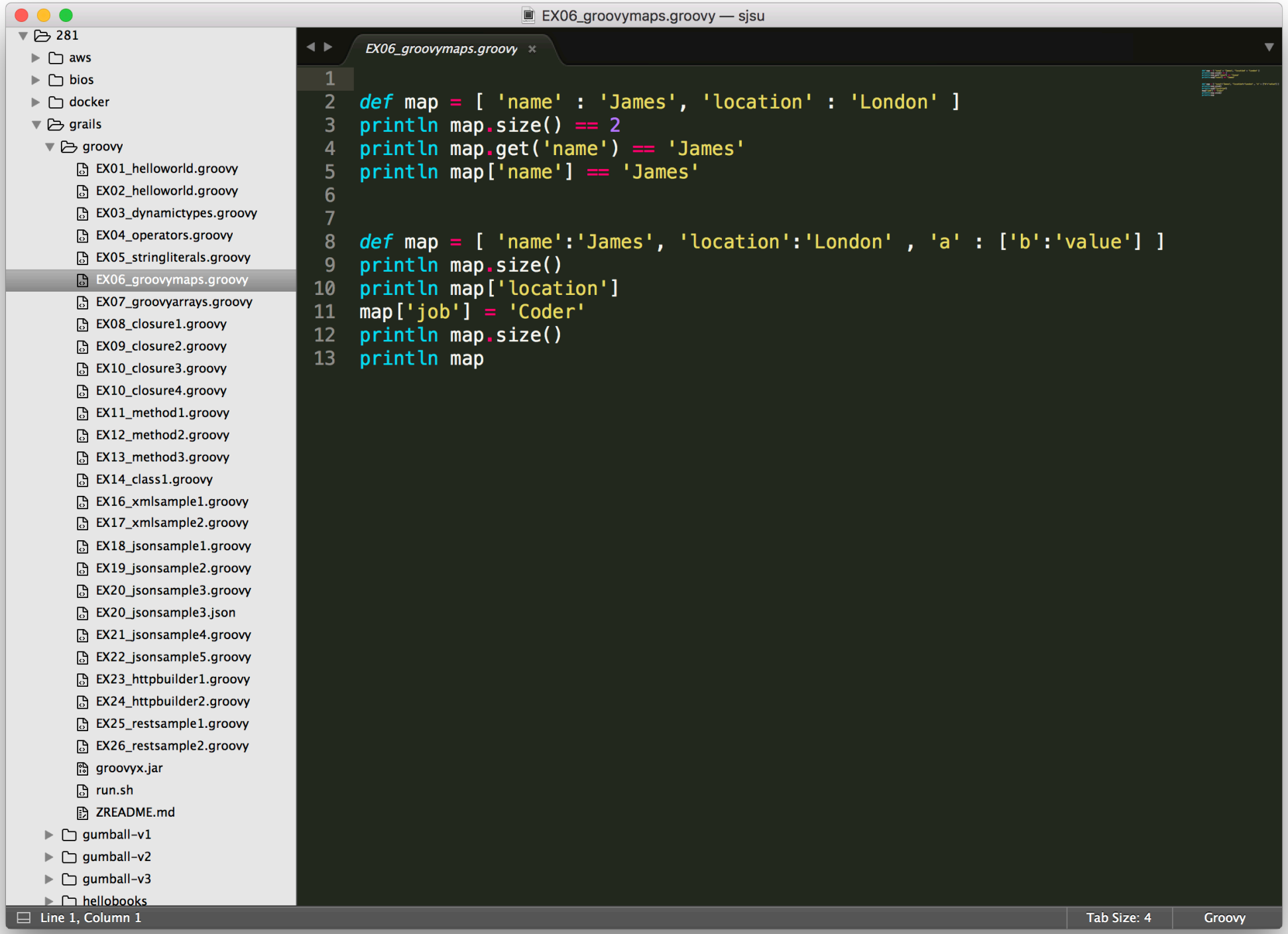

EX01_helloworld.groovy — sjsu
281
aws
bios
docker
grails
groovy
EX01_helloworld.groovy
EX02_helloworld.groovy
EX03_dynamictypes.groovy
EX04_operators.groovy
EX05_stringliterals.groovy
EX06_groovymaps.groovy
EX07_groovyarrays.groovy
EX08_closure1.groovy
EX09_closure2.groovy
EX10_closure3.groovy
EX10_closure4.groovy
EX11_method1.groovy
EX12_method2.groovy
EX13_method3.groovy
EX14_class1.groovy
EX16_xmlsample1.groovy
EX17_xmlsample2.groovy
EX18_jsonsample1.groovy
EX19_jsonsample2.groovy
EX20_jsonsample3.groovy
EX20_jsonsample3.json
EX21_jsonsample4.groovy
EX22_jsonsample5.groovy
EX23_httpbuilder1.groovy
EX24_httpbuilder2.groovy
EX25_restsample1.groovy
EX26_restsample2.groovy
groovyx.jar
run.sh
ZREADME.md
gumball-v1
gumball-v2
gumball-v3
hellobooks
EX01_helloworld.groovy
1 public class HelloWorld
2 {
3 public static void main(String[] args)
4 {
5 System.out.println("Hello World!") ;
6 }
7 }
8
9
10
Line 1, Column 1
Tab Size: 4
Groovy

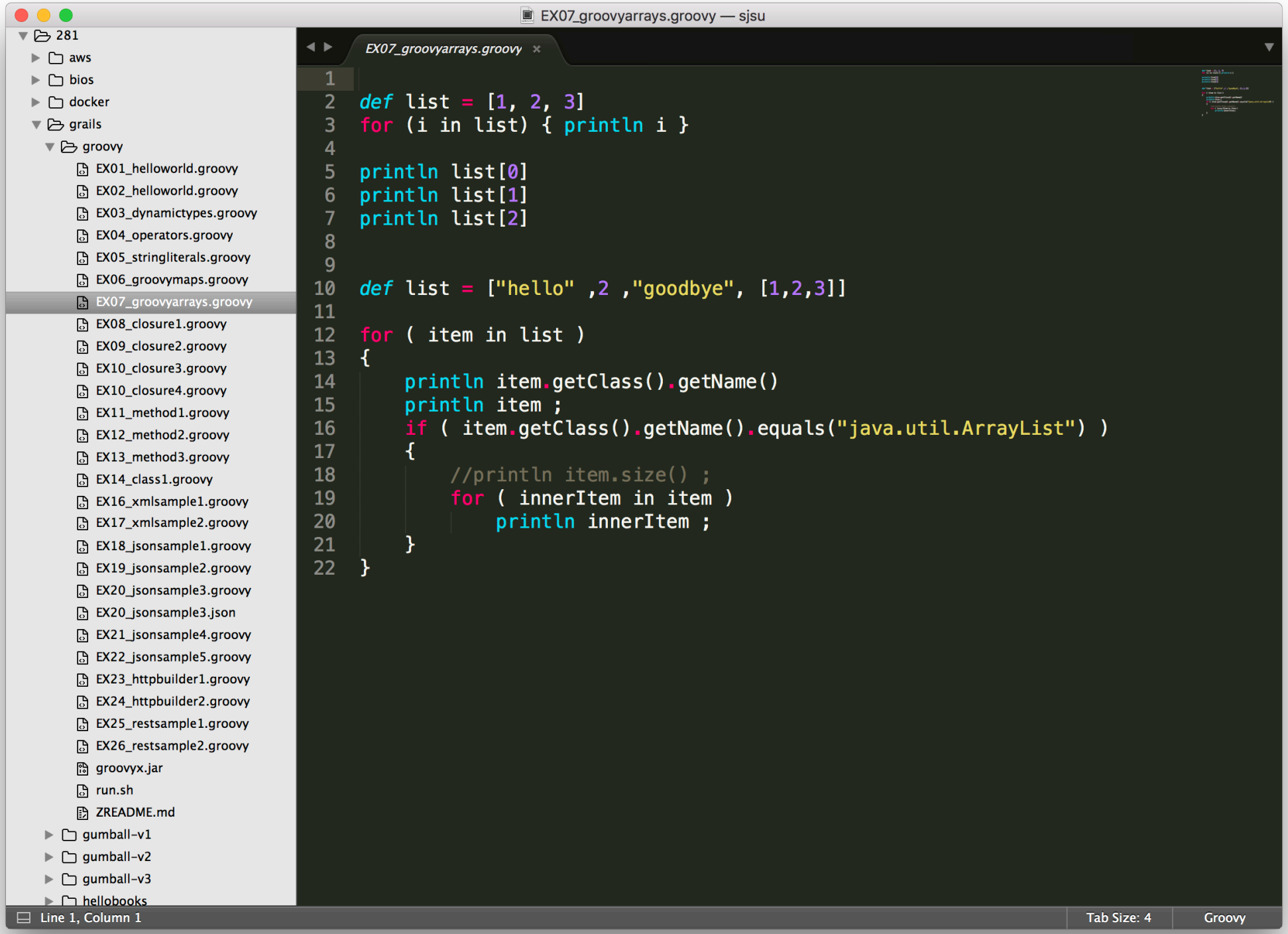

EX02_helloworld.groovy — sjsu
281
aws
bios
docker
grails
groovy
EX01_helloworld.groovy
EX02_helloworld.groovy
EX03_dynamictypes.groovy
EX04_operators.groovy
EX05_stringliterals.groovy
EX06_groovymaps.groovy
EX07_groovyarrays.groovy
EX08_closure1.groovy
EX09_closure2.groovy
EX10_closure3.groovy
EX10_closure4.groovy
EX11_method1.groovy
EX12_method2.groovy
EX13_method3.groovy
EX14_class1.groovy
EX16_xmlsample1.groovy
EX17_xmlsample2.groovy
EX18_jsonsample1.groovy
EX19_jsonsample2.groovy
EX20_jsonsample3.groovy
EX20_jsonsample3.json
EX21_jsonsample4.groovy
EX22_jsonsample5.groovy
EX23_httpbuilder1.groovy
EX24_httpbuilder2.groovy
EX25_restsample1.groovy
EX26_restsample2.groovy
groovyx.jar
run.sh
ZREADME.md
gumball-v1
gumball-v2
gumball-v3
hellobooks
EX02_helloworld.groovy
1
2
System.out.println("Hello World!") ;
Line 1, Column 1
Tab Size: 4
Groovy

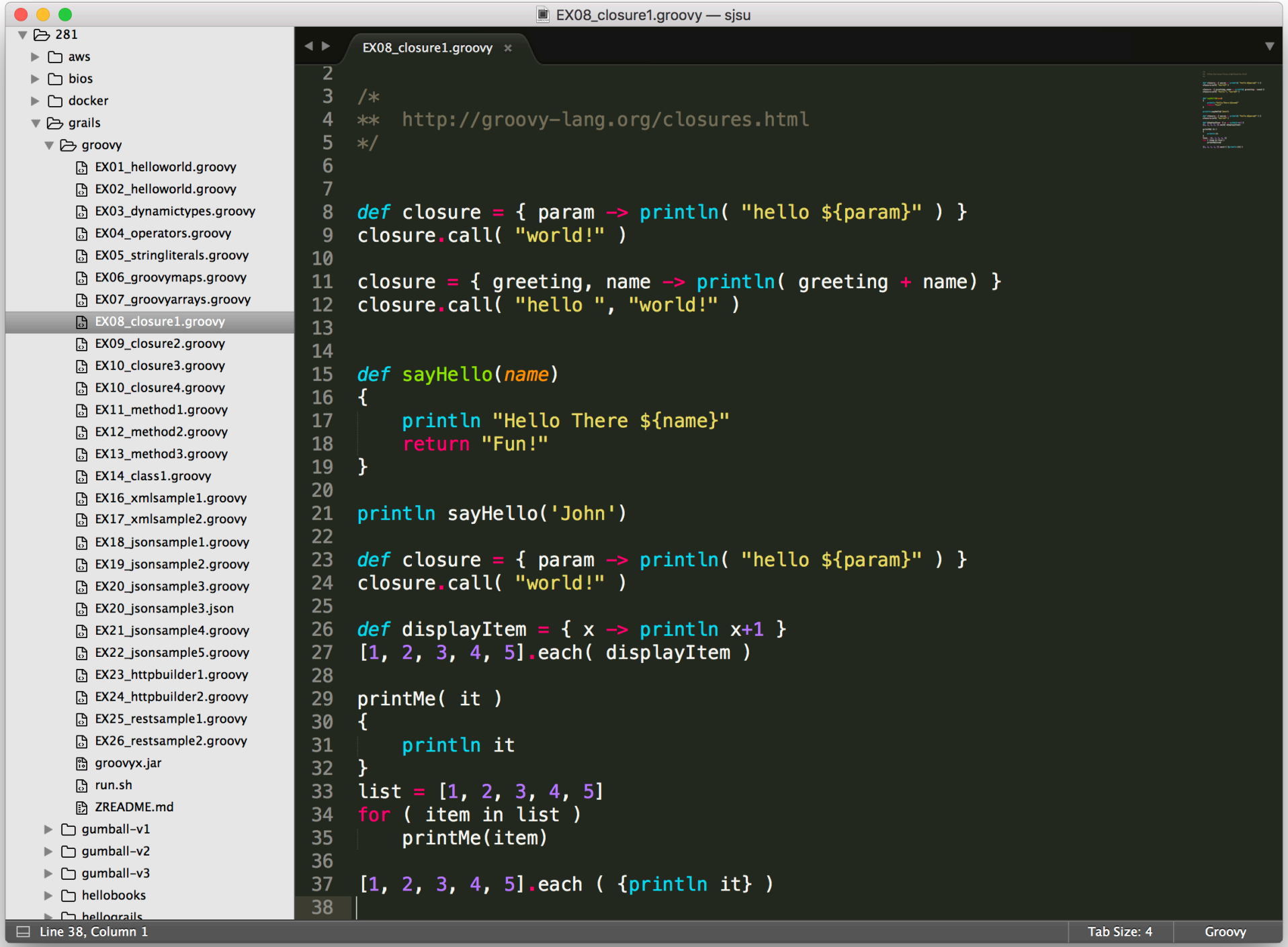

EX03_dynamictypes.groovy — sjsu
281
aws
bios
docker
grails
groovy
EX01_helloworld.groovy
EX02_helloworld.groovy
EX03_dynamictypes.groovy
EX04_operators.groovy
EX05_stringliterals.groovy
EX06_groovymaps.groovy
EX07_groovyarrays.groovy
EX08_closure1.groovy
EX09_closure2.groovy
EX10_closure3.groovy
EX10_closure4.groovy
EX11_method1.groovy
EX12_method2.groovy
EX13_method3.groovy
EX14_class1.groovy
EX16_xmlsample1.groovy
EX17_xmlsample2.groovy
EX18_jsonsample1.groovy
EX19_jsonsample2.groovy
EX20_jsonsample3.groovy
EX20_jsonsample3.json
EX21_jsonsample4.groovy
EX22_jsonsample5.groovy
EX23_httpbuilder1.groovy
EX24_httpbuilder2.groovy
EX25_restsample1.groovy
EX26_restsample2.groovy
groovyx.jar
run.sh
ZREADME.md
gumball-v1
gumball-v2
gumball-v3
hellobooks
EX03_dynamictypes.groovy
1
2 int age=25
3 def number=age
4 message = "Hello World"
5
6 println age.getClass().getName()
7 println number.getClass().getName()
8 println message.getClass().getName()
9
10 number=message
11 println number.getClass().getName()
12
13
14 5.dump()
15 5.metaClass.methods
Line 1, Column 1
Tab Size: 4
Groovy

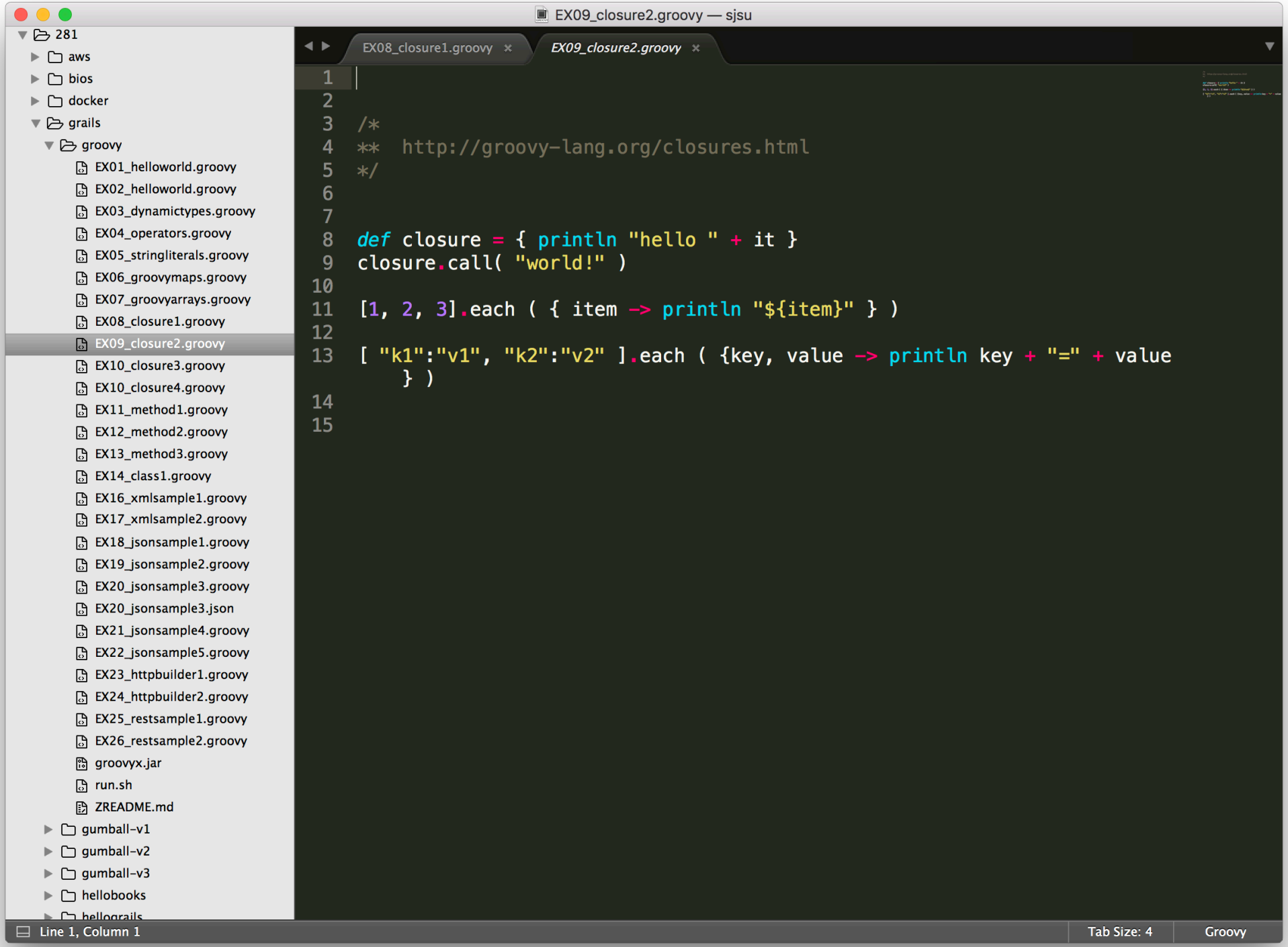

EX04_operators.groovy — sjsu
281
aws
bios
docker
grails
groovy
EX01_helloworld.groovy
EX02_helloworld.groovy
EX03_dynamictypes.groovy
EX04_operators.groovy
EX05_stringliterals.groovy
EX06_groovymaps.groovy
EX07_groovyarrays.groovy
EX08_closure1.groovy
EX09_closure2.groovy
EX10_closure3.groovy
EX10_closure4.groovy
EX11_method1.groovy
EX12_method2.groovy
EX13_method3.groovy
EX14_class1.groovy
EX16_xmlsample1.groovy
EX17_xmlsample2.groovy
EX18_jsonsample1.groovy
EX19_jsonsample2.groovy
EX20_jsonsample3.groovy
EX20_jsonsample3.json
EX21_jsonsample4.groovy
EX22_jsonsample5.groovy
EX23_httpbuilder1.groovy
EX24_httpbuilder2.groovy
EX25_restsample1.groovy
EX26_restsample2.groovy
groovyx.jar
run.sh
ZREADME.md
gumball-v1
gumball-v2
gumball-v3
hellobooks
EX04_operators.groovy
1
println 5<3
println 5.compareTo(3) < 0
println 5>3
println 5.compareTo(3) > 0
println 5==3
println 5.equals(3)
println 5!=3
println !(5.equals(3))
Line 1, Column 1
Tab Size: 4
Groovy

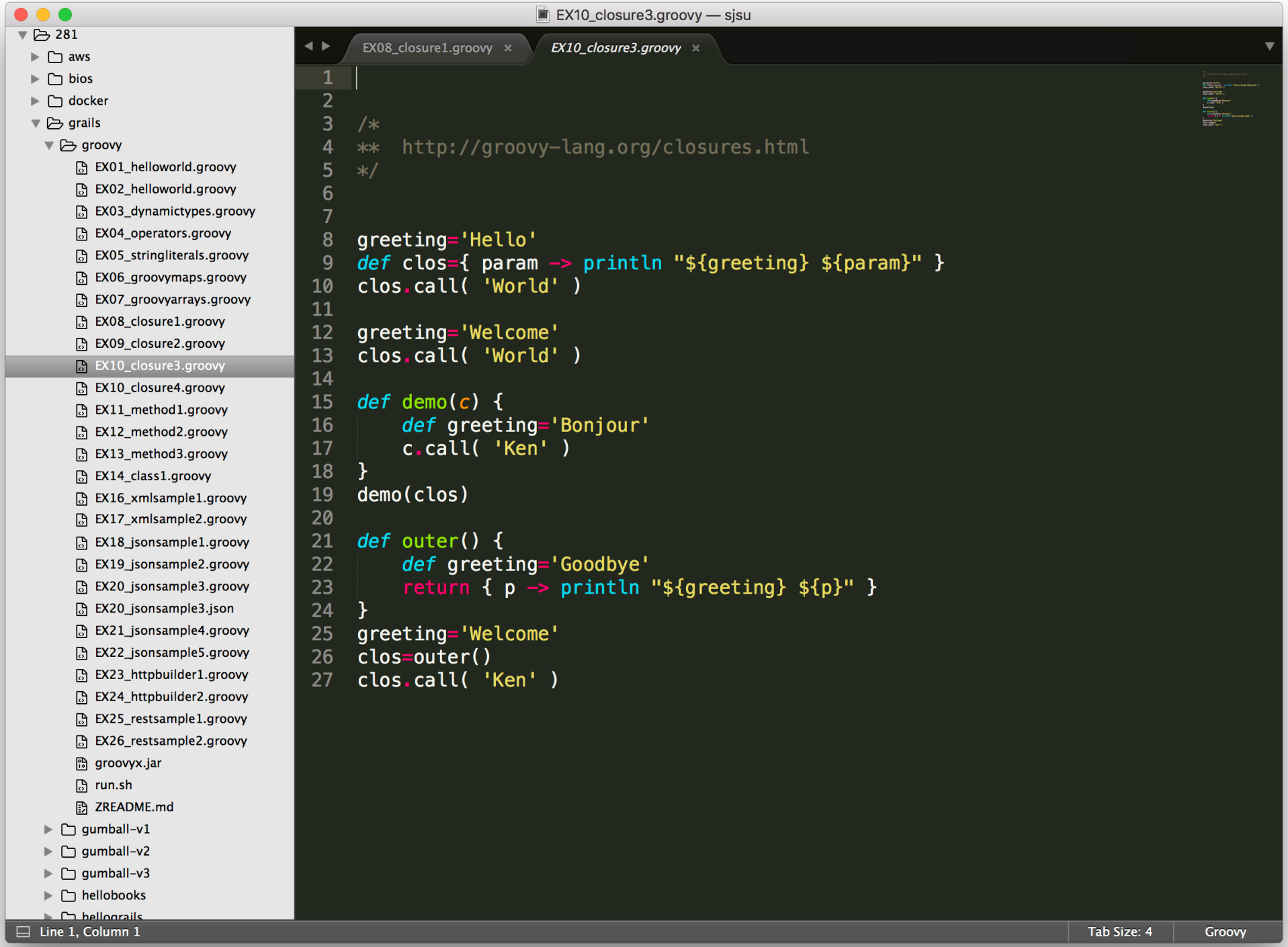

EX05_stringliterals.groovy — sjsu
281
aws
bios
docker
grails
groovy
EX01_helloworld.groovy
EX02_helloworld.groovy
EX03_dynamictypes.groovy
EX04_operators.groovy
EX05_stringliterals.groovy
EX06_groovymaps.groovy
EX07_groovyarrays.groovy
EX08_closure1.groovy
EX09_closure2.groovy
EX10_closure3.groovy
EX10_closure4.groovy
EX11_method1.groovy
EX12_method2.groovy
EX13_method3.groovy
EX14_class1.groovy
EX16_xmlsample1.groovy
EX17_xmlsample2.groovy
EX18_jsonsample1.groovy
EX19_jsonsample2.groovy
EX20_jsonsample3.groovy
EX20_jsonsample3.json
EX21_jsonsample4.groovy
EX22_jsonsample5.groovy
EX23_httpbuilder1.groovy
EX24_httpbuilder2.groovy
EX25_restsample1.groovy
EX26_restsample2.groovy
groovyx.jar
run.sh
README.md
gumball-v1
gumball-v2
gumball-v3
hellobooks
EX05_stringliterals.groovy
1 def age=25
2
3 println 'My age is ${age}'
4 println "My age is ${age}"
5 println "My age is \${age}"
6
7
8 s1="Hello"
9 s2="World"
10 s3=123
11 printf (" %-2s | %-2s| %5d\n", s1, s2, s3)
12
13
14 a='Hello'
15 b='World'
16 println a.concat(' ').concat(b)
17 println a + ' ' + b
18 println "${a} ${b}"
19 printf "%s %s\n", a , b
20
Line 1, Column 1
Tab Size: 4
Groovy

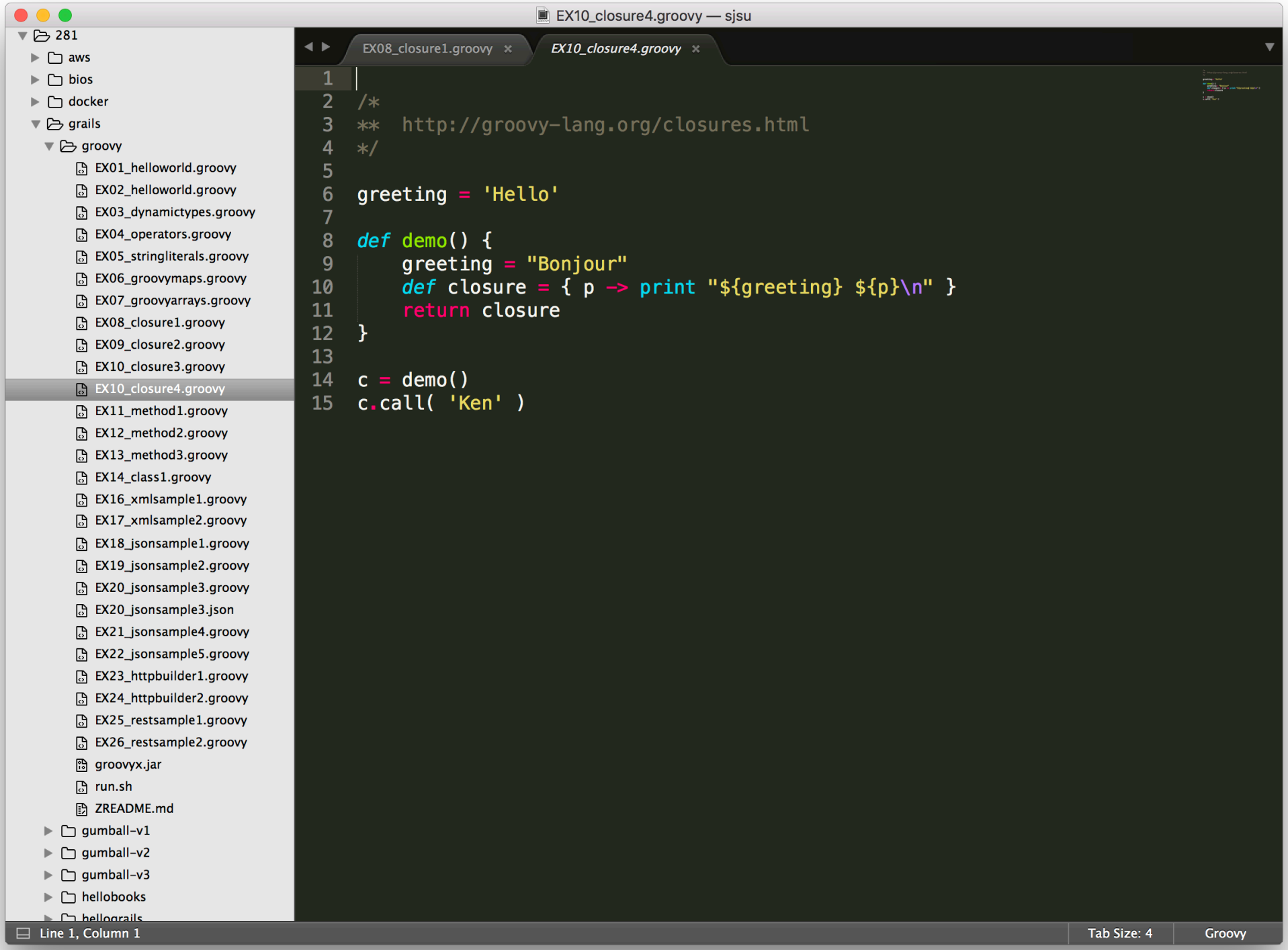

EX06_groovymaps.groovy — sjsu
281
aws
bios
docker
grails
groovy
EX01_helloworld.groovy
EX02_helloworld.groovy
EX03_dynamictypes.groovy
EX04_operators.groovy
EX05_stringliterals.groovy
EX06_groovymaps.groovy
EX07_groovyarrays.groovy
EX08_closure1.groovy
EX09_closure2.groovy
EX10_closure3.groovy
EX10_closure4.groovy
EX11_method1.groovy
EX12_method2.groovy
EX13_method3.groovy
EX14_class1.groovy
EX16_xmlsample1.groovy
EX17_xmlsample2.groovy
EX18_jsonsample1.groovy
EX19_jsonsample2.groovy
EX20_jsonsample3.groovy
EX20_jsonsample3.json
EX21_jsonsample4.groovy
EX22_jsonsample5.groovy
EX23_httpbuilder1.groovy
EX24_httpbuilder2.groovy
EX25_restsample1.groovy
EX26_restsample2.groovy
groovyx.jar
run.sh
ZREADME.md
gumball-v1
gumball-v2
gumball-v3
hellobooks
EX06_groovymaps.groovy
1 def map = ['name' : 'James', 'location' : 'London']
2 println map.size() == 2
3 println map.get('name') == 'James'
4 println map['name'] == 'James'
5
6
7 def map = ['name':'James', 'location':'London' , 'a' : ['b':'value']]
8 println map.size()
9 println map['location']
10 map['job'] = 'Coder'
11 println map.size()
12 println map
13
Line 1, Column 1
Tab Size: 4
Groovy

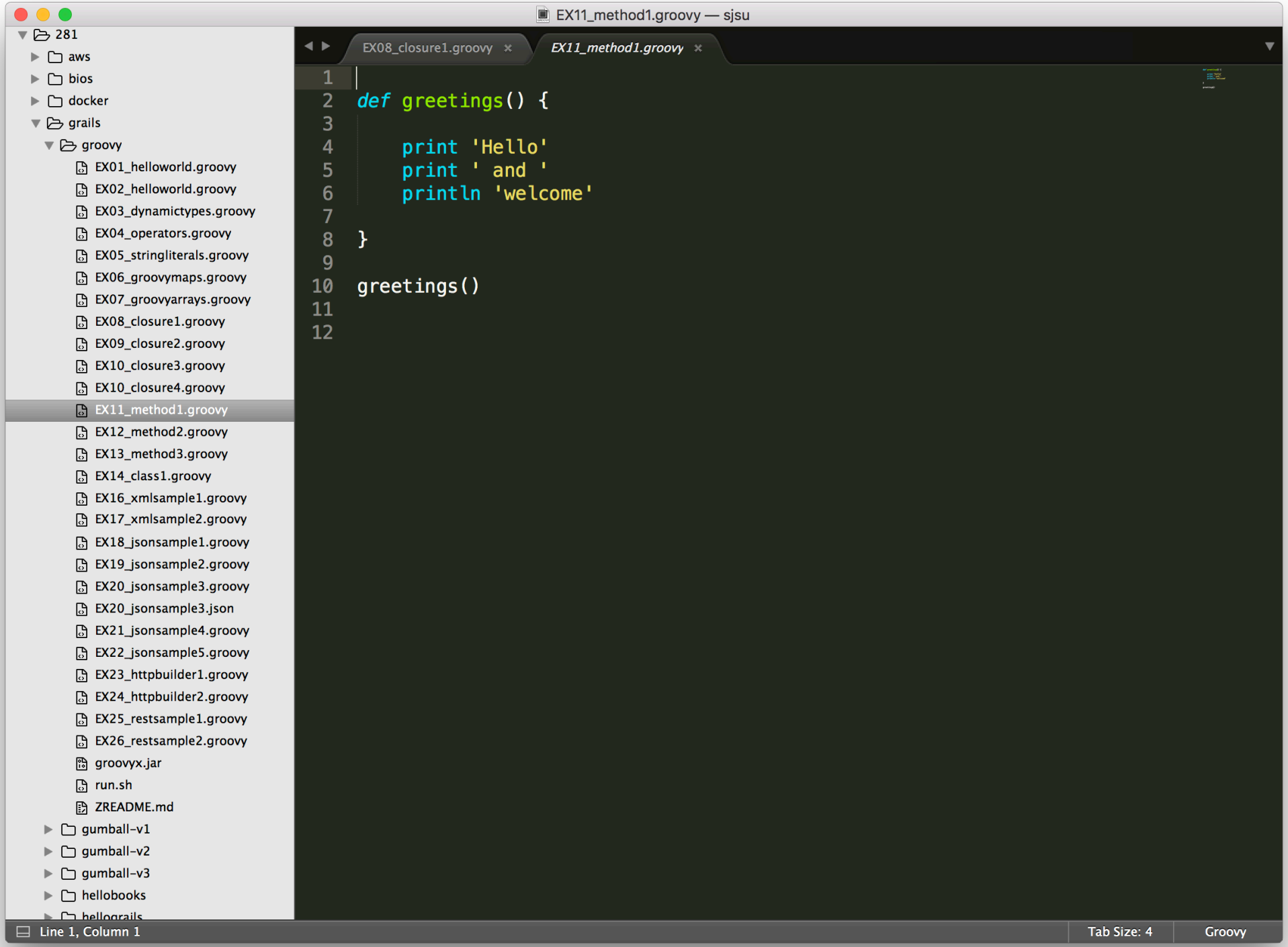

EX07_groovyarrays.groovy — sjsu
281
aws
bios
docker
grails
groovy
EX01_helloworld.groovy
EX02_helloworld.groovy
EX03_dynamictypes.groovy
EX04_operators.groovy
EX05_stringliterals.groovy
EX06_groovymaps.groovy
EX07_groovyarrays.groovy
EX08_closure1.groovy
EX09_closure2.groovy
EX10_closure3.groovy
EX10_closure4.groovy
EX11_method1.groovy
EX12_method2.groovy
EX13_method3.groovy
EX14_class1.groovy
EX16_xmlsample1.groovy
EX17_xmlsample2.groovy
EX18_jsonsample1.groovy
EX19_jsonsample2.groovy
EX20_jsonsample3.groovy
EX20_jsonsample3.json
EX21_jsonsample4.groovy
EX22_jsonsample5.groovy
EX23_httpbuilder1.groovy
EX24_httpbuilder2.groovy
EX25_restsample1.groovy
EX26_restsample2.groovy
groovyx.jar
run.sh
ZREADME.md
gumball-v1
gumball-v2
gumball-v3
hellobooks
EX07_groovyarrays.groovy
1
def list = [1, 2, 3]
for (i in list) { println i }
println list[0]
println list[1]
println list[2]
def list = ["hello" ,2 ,"goodbye", [1,2,3]]
for (item in list)
{
println item.getClass().getName()
println item ;
if (item.getClass().getName().equals("java.util.ArrayList"))
{
//println item.size() ;
for (innerItem in item)
println innerItem ;
}
}
Line 1, Column 1
Tab Size: 4
Groovy

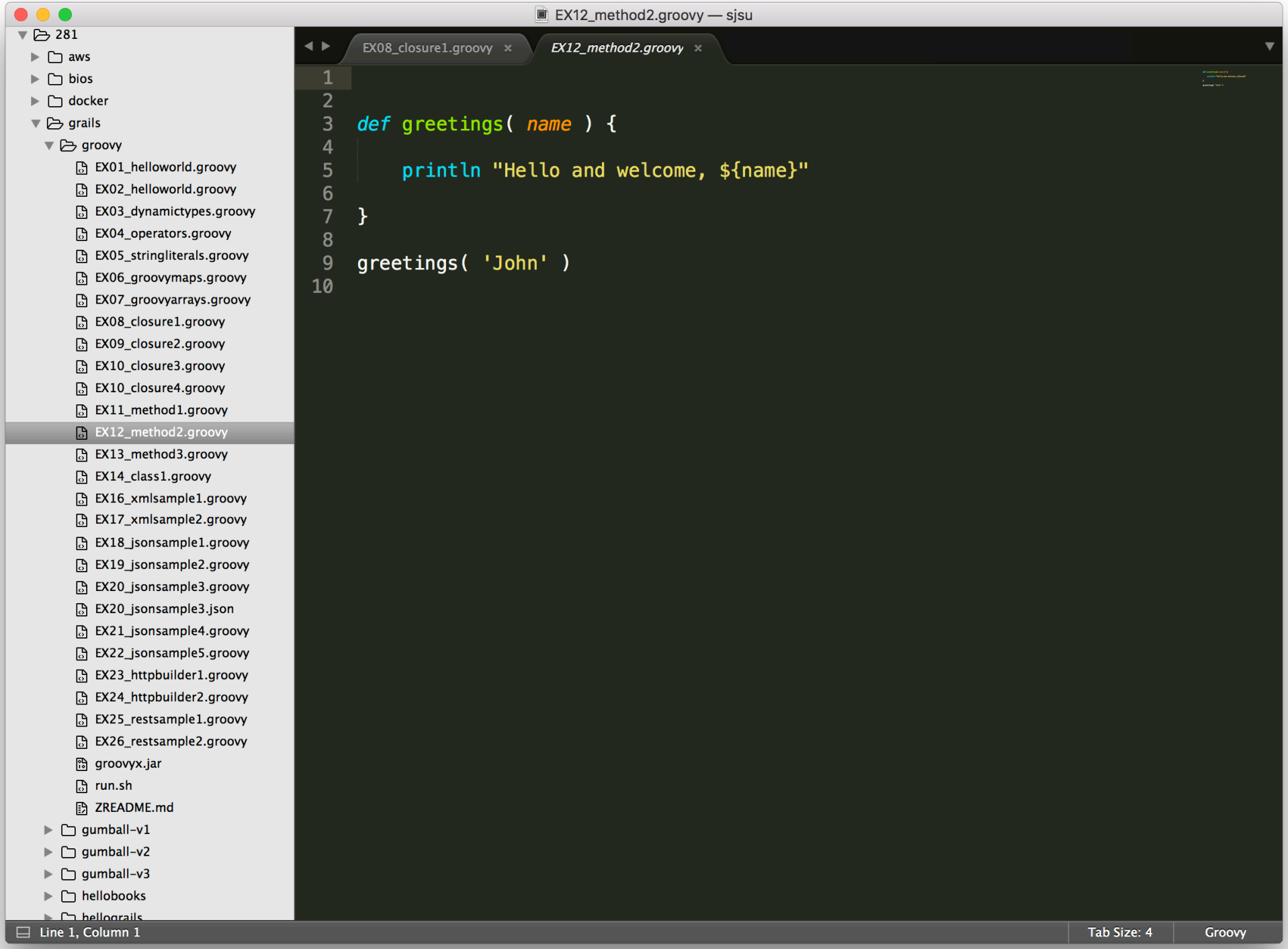

EX08_closure1.groovy — sjsu
281
aws
bios
docker
grails
groovy
EX01_helloworld.groovy
EX02_helloworld.groovy
EX03_dynamictypes.groovy
EX04_operators.groovy
EX05_stringliterals.groovy
EX06_groovymaps.groovy
EX07_groovyarrays.groovy
EX08_closure1.groovy
EX09_closure2.groovy
EX10_closure3.groovy
EX10_closure4.groovy
EX11_method1.groovy
EX12_method2.groovy
EX13_method3.groovy
EX14_class1.groovy
EX16_xmlsample1.groovy
EX17_xmlsample2.groovy
EX18_jsonsample1.groovy
EX19_jsonsample2.groovy
EX20_jsonsample3.groovy
EX20_jsonsample3.json
EX21_jsonsample4.groovy
EX22_jsonsample5.groovy
EX23_httpbuilder1.groovy
EX24_httpbuilder2.groovy
EX25_restsample1.groovy
EX26_restsample2.groovy
groovyx.jar
run.sh
ZREADME.md
gumball-v1
gumball-v2
gumball-v3
hellobooks
hellograils
EX08_closure1.groovy
2
3 /*
4 ** http://groovy-lang.org/closures.html
5 */
6
7
8 def closure = { param -> println("hello ${param}") }
9 closure.call("world!")
10
11 closure = { greeting, name -> println(greeting + name) }
12 closure.call("hello ", "world!")
13
14
15 def sayHello(name)
16 {
17 println "Hello There ${name}"
18 return "Fun!"
19 }
20
21 println sayHello('John')
22
23 def closure = { param -> println("hello ${param}") }
24 closure.call("world!")
25
26 def displayItem = { x -> println x+1 }
27 [1, 2, 3, 4, 5].each(displayItem)
28
29 printMe(it)
30 {
31 println it
32 }
33 list = [1, 2, 3, 4, 5]
34 for (item in list)
35 printMe(item)
36
37 [1, 2, 3, 4, 5].each ({println it})
38
Line 38, Column 1
Tab Size: 4
Groovy

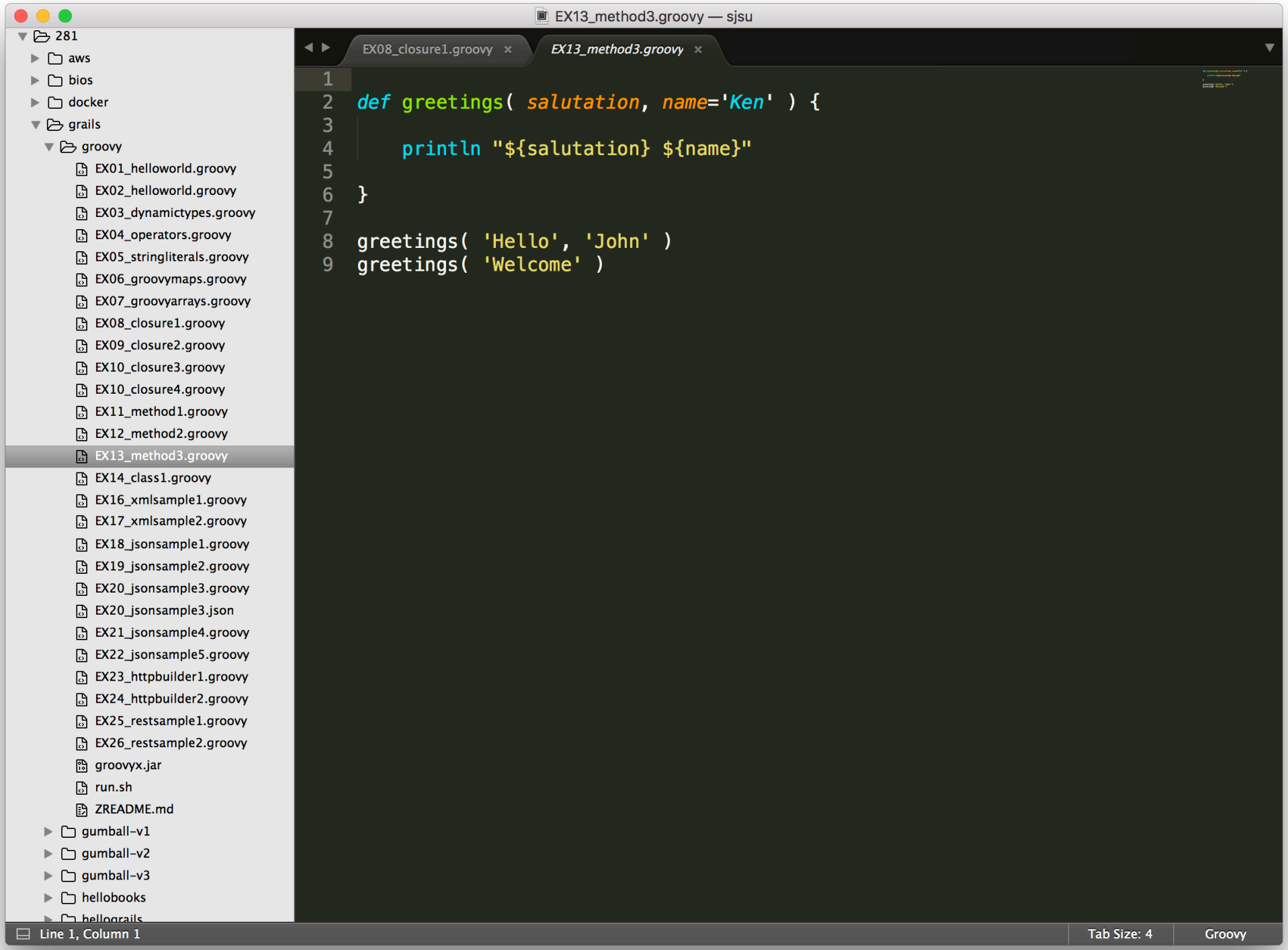

EX09_closure2.groovy — sjsu
281
aws
bios
docker
grails
groovy
EX01_helloworld.groovy
EX02_helloworld.groovy
EX03_dynamictypes.groovy
EX04_operators.groovy
EX05_stringliterals.groovy
EX06_groovymaps.groovy
EX07_groovyarrays.groovy
EX08_closure1.groovy
EX09_closure2.groovy
EX10_closure3.groovy
EX10_closure4.groovy
EX11_method1.groovy
EX12_method2.groovy
EX13_method3.groovy
EX14_class1.groovy
EX16_xmlsample1.groovy
EX17_xmlsample2.groovy
EX18_jsonsample1.groovy
EX19_jsonsample2.groovy
EX20_jsonsample3.groovy
EX20_jsonsample3.json
EX21_jsonsample4.groovy
EX22_jsonsample5.groovy
EX23_httpbuilder1.groovy
EX24_httpbuilder2.groovy
EX25_restsample1.groovy
EX26_restsample2.groovy
groovyx.jar
run.sh
README.md
gumball-v1
gumball-v2
gumball-v3
hellobooks
hellograils
EX08_closure1.groovy
EX09_closure2.groovy
1
2
3 /*
4 ** http://groovy-lang.org/closures.html
5 */
6
7
8 def closure = { println "hello " + it }
9 closure.call("world!")
10
11 [1, 2, 3].each ({ item -> println "${item}" })
12
13 ["k1":"v1", "k2":"v2"].each ({key, value -> println key + "=" + value
})
14
15
Line 1, Column 1
Tab Size: 4
Groovy

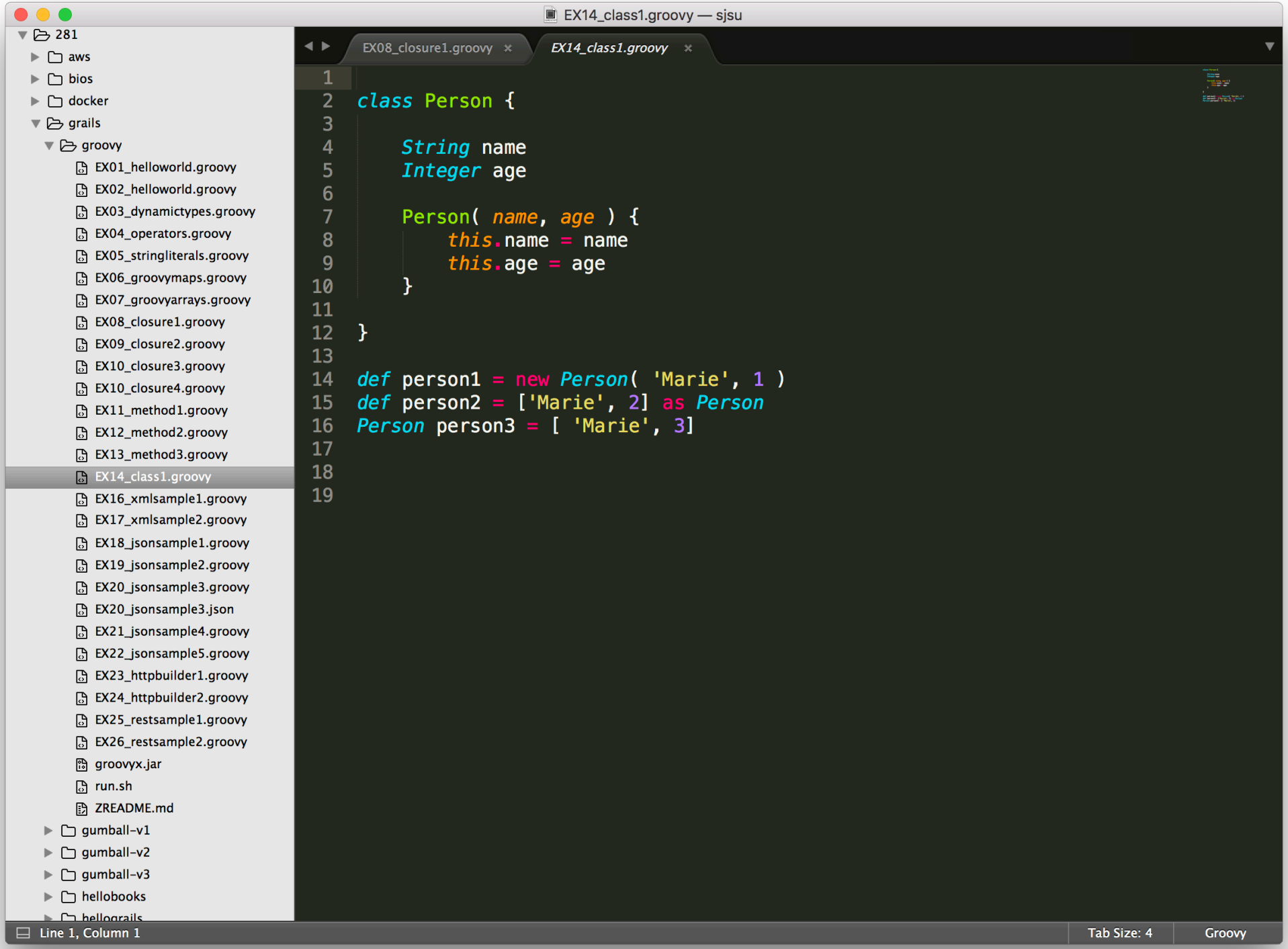

EX10_closure3.groovy — sjsu
281
aws
bios
docker
grails
groovy
EX01_helloworld.groovy
EX02_helloworld.groovy
EX03_dynamictypes.groovy
EX04_operators.groovy
EX05_stringliterals.groovy
EX06_groovymaps.groovy
EX07_groovyarrays.groovy
EX08_closure1.groovy
EX09_closure2.groovy
EX10_closure3.groovy
EX10_closure4.groovy
EX11_method1.groovy
EX12_method2.groovy
EX13_method3.groovy
EX14_class1.groovy
EX16_xmlsample1.groovy
EX17_xmlsample2.groovy
EX18_jsonsample1.groovy
EX19_jsonsample2.groovy
EX20_jsonsample3.groovy
EX20_jsonsample3.json
EX21_jsonsample4.groovy
EX22_jsonsample5.groovy
EX23_httpbuilder1.groovy
EX24_httpbuilder2.groovy
EX25_restsample1.groovy
EX26_restsample2.groovy
groovyx.jar
run.sh
ZREADME.md
gumball-v1
gumball-v2
gumball-v3
hellobooks
hellograils
EX08_closure1.groovy
EX10_closure3.groovy
1 /*
2 ** http://groovy-lang.org/closures.html
3 */
4
5 greeting='Hello'
6 def clos={ param -> println "${greeting} ${param}" }
7 clos.call('World')
8
9 greeting='Welcome'
10 clos.call('World')
11
12 def demo(c) {
13 def greeting='Bonjour'
14 c.call('Ken')
15 }
16 demo(clos)
17
18 def outer() {
19 def greeting='Goodbye'
20 return { p -> println "${greeting} ${p}" }
21 }
22 greeting='Welcome'
23 clos=outer()
24 clos.call('Ken')
Line 1, Column 1
Tab Size: 4
Groovy

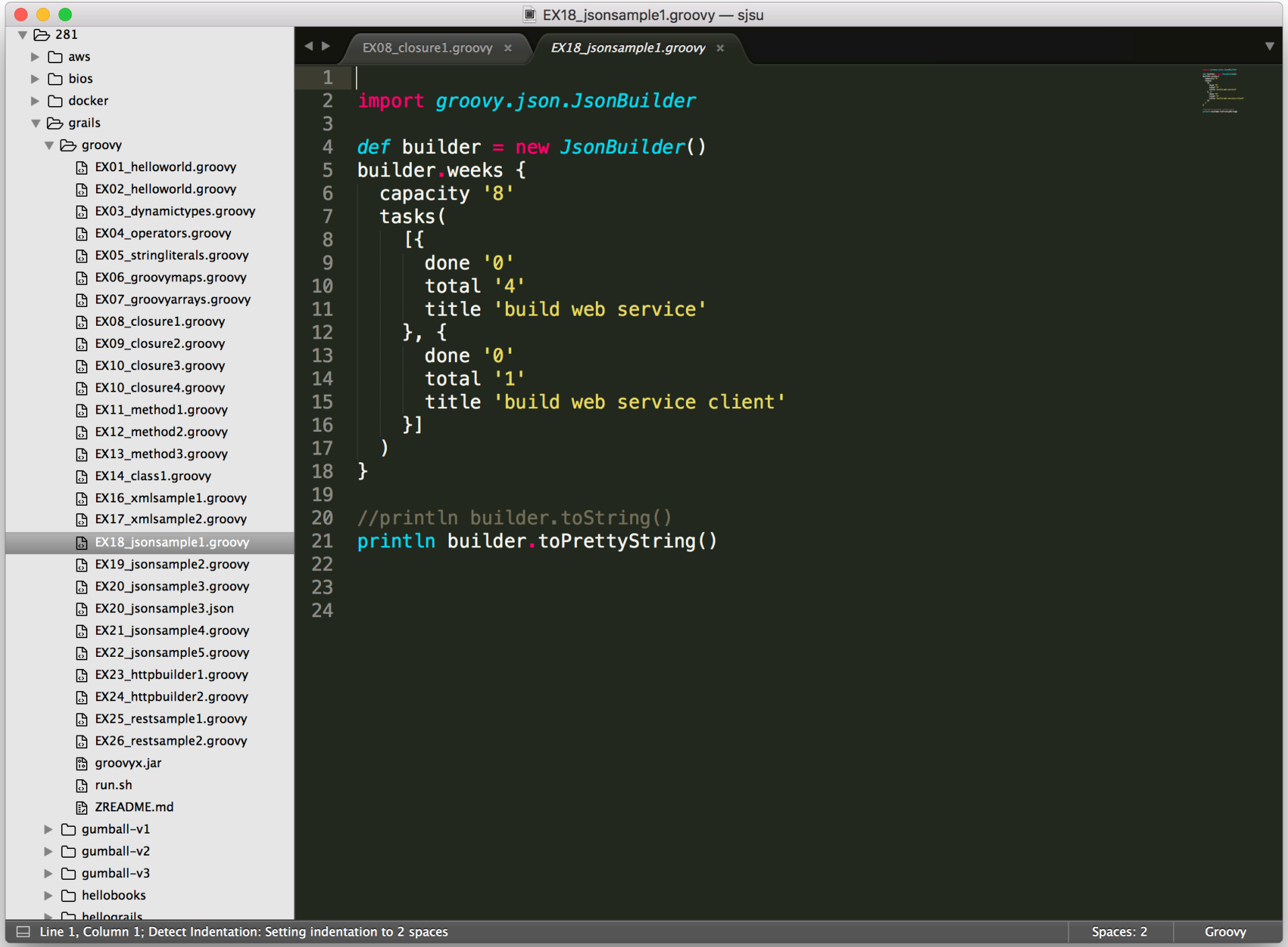

EX10_closure4.groovy — sjsu
281
aws
bios
docker
grails
groovy
EX01_helloworld.groovy
EX02_helloworld.groovy
EX03_dynamictypes.groovy
EX04_operators.groovy
EX05_stringliterals.groovy
EX06_groovymaps.groovy
EX07_groovyarrays.groovy
EX08_closure1.groovy
EX09_closure2.groovy
EX10_closure3.groovy
EX10_closure4.groovy
EX11_method1.groovy
EX12_method2.groovy
EX13_method3.groovy
EX14_class1.groovy
EX16_xmlsample1.groovy
EX17_xmlsample2.groovy
EX18_jsonsample1.groovy
EX19_jsonsample2.groovy
EX20_jsonsample3.groovy
EX20_jsonsample3.json
EX21_jsonsample4.groovy
EX22_jsonsample5.groovy
EX23_httpbuilder1.groovy
EX24_httpbuilder2.groovy
EX25_restsample1.groovy
EX26_restsample2.groovy
groovyx.jar
run.sh
ZREADME.md
gumball-v1
gumball-v2
gumball-v3
hellobooks
hellograils
EX08_closure1.groovy
EX10_closure4.groovy
1 /*
2 ** http://groovy-lang.org/closures.html
3 */
4
5 greeting = 'Hello'
6
7 def demo() {
8 greeting = "Bonjour"
9 def closure = { p -> print "${greeting} ${p}\n" }
10 return closure
11 }
12
13 c = demo()
14 c.call('Ken')
Line 1, Column 1
Tab Size: 4
Groovy

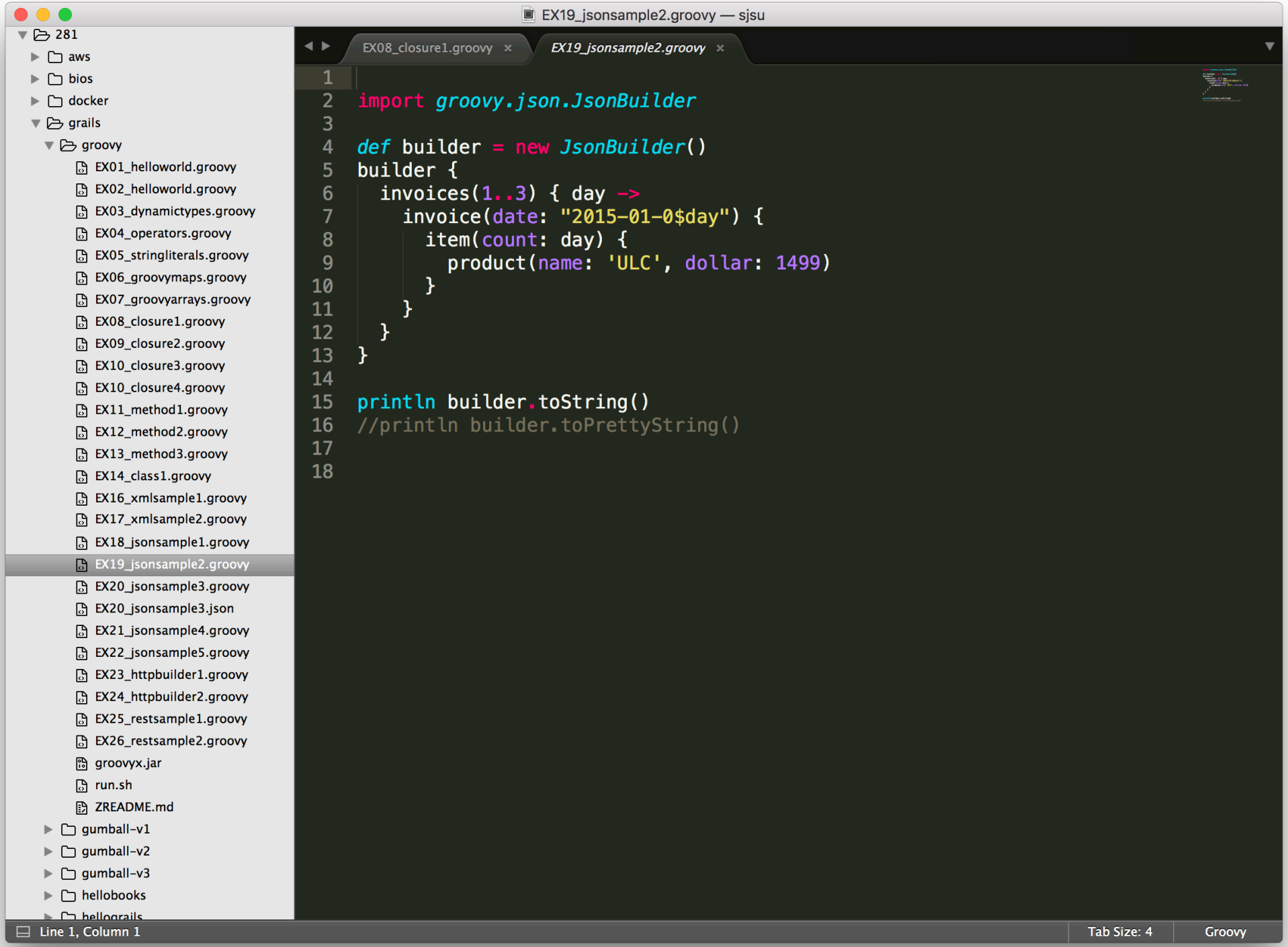

EX11_method1.groovy — sjsu
281
aws
bios
docker
grails
groovy
EX01_helloworld.groovy
EX02_helloworld.groovy
EX03_dynamictypes.groovy
EX04_operators.groovy
EX05_stringliterals.groovy
EX06_groovymaps.groovy
EX07_groovyarrays.groovy
EX08_closure1.groovy
EX09_closure2.groovy
EX10_closure3.groovy
EX10_closure4.groovy
EX11_method1.groovy
EX12_method2.groovy
EX13_method3.groovy
EX14_class1.groovy
EX16_xmlsample1.groovy
EX17_xmlsample2.groovy
EX18_jsonsample1.groovy
EX19_jsonsample2.groovy
EX20_jsonsample3.groovy
EX20_jsonsample3.json
EX21_jsonsample4.groovy
EX22_jsonsample5.groovy
EX23_httpbuilder1.groovy
EX24_httpbuilder2.groovy
EX25_restsample1.groovy
EX26_restsample2.groovy
groovyx.jar
run.sh
ZREADME.md
gumball-v1
gumball-v2
gumball-v3
hellobooks
hellograils
EX08_closure1.groovy
EX11_method1.groovy
1
def greetings() {
print 'Hello'
print ' and '
println 'welcome'
}
greetings()
Line 1, Column 1
Tab Size: 4
Groovy

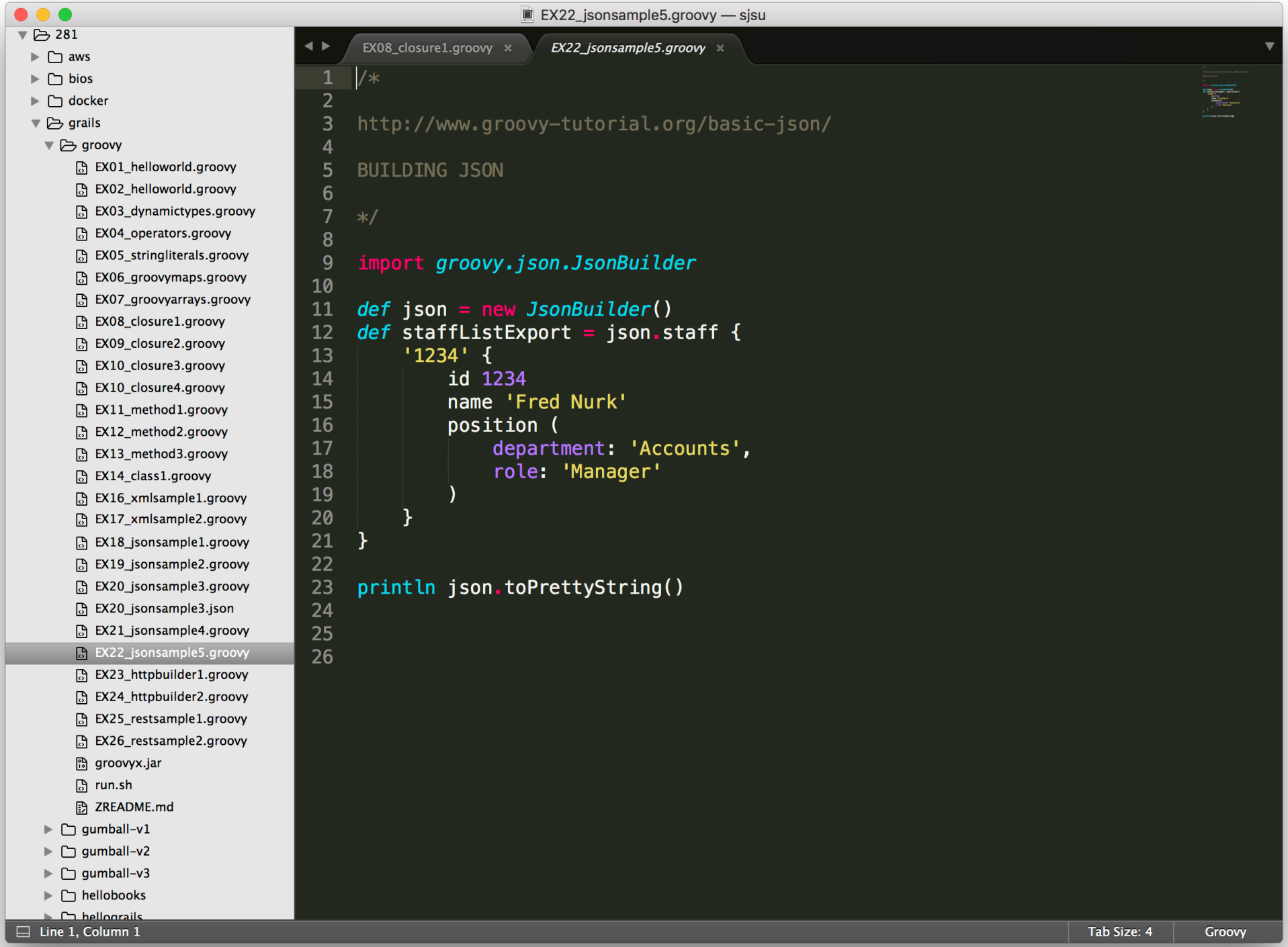

EX12_method2.groovy — sjsu
281
aws
bios
docker
grails
groovy
EX01_helloworld.groovy
EX02_helloworld.groovy
EX03_dynamictypes.groovy
EX04_operators.groovy
EX05_stringliterals.groovy
EX06_groovymaps.groovy
EX07_groovyarrays.groovy
EX08_closure1.groovy
EX09_closure2.groovy
EX10_closure3.groovy
EX10_closure4.groovy
EX11_method1.groovy
EX12_method2.groovy
EX13_method3.groovy
EX14_class1.groovy
EX16_xmlsample1.groovy
EX17_xmlsample2.groovy
EX18_jsonsample1.groovy
EX19_jsonsample2.groovy
EX20_jsonsample3.groovy
EX20_jsonsample3.json
EX21_jsonsample4.groovy
EX22_jsonsample5.groovy
EX23_httpbuilder1.groovy
EX24_httpbuilder2.groovy
EX25_restsample1.groovy
EX26_restsample2.groovy
groovyx.jar
run.sh
README.md
gumball-v1
gumball-v2
gumball-v3
hellobooks
hellograils
EX08_closure1.groovy
EX12_method2.groovy
def greetings(name) {
println "Hello and welcome, ${name}"
}
greetings('John')
Line 1, Column 1
Tab Size: 4
Groovy


EX13_method3.groovy — sjsu
281
aws
bios
docker
grails
groovy
EX01_helloworld.groovy
EX02_helloworld.groovy
EX03_dynamictypes.groovy
EX04_operators.groovy
EX05_stringliterals.groovy
EX06_groovymaps.groovy
EX07_groovyarrays.groovy
EX08_closure1.groovy
EX09_closure2.groovy
EX10_closure3.groovy
EX10_closure4.groovy
EX11_method1.groovy
EX12_method2.groovy
EX13_method3.groovy
EX14_class1.groovy
EX16_xmlsample1.groovy
EX17_xmlsample2.groovy
EX18_jsonsample1.groovy
EX19_jsonsample2.groovy
EX20_jsonsample3.groovy
EX20_jsonsample3.json
EX21_jsonsample4.groovy
EX22_jsonsample5.groovy
EX23_httpbuilder1.groovy
EX24_httpbuilder2.groovy
EX25_restsample1.groovy
EX26_restsample2.groovy
groovyx.jar
run.sh
ZREADME.md
gumball-v1
gumball-v2
gumball-v3
hellobooks
hellograils
EX08_closure1.groovy
EX13_method3.groovy
def greetings(salutation, name='Ken') {
println "${salutation} ${name}"
}
greetings('Hello', 'John')
greetings('Welcome')
Line 1, Column 1
Tab Size: 4
Groovy


EX14_class1.groovy — sjsu
281
aws
bios
docker
grails
groovy
EX01_helloworld.groovy
EX02_helloworld.groovy
EX03_dynamictypes.groovy
EX04_operators.groovy
EX05_stringliterals.groovy
EX06_groovymaps.groovy
EX07_groovyarrays.groovy
EX08_closure1.groovy
EX09_closure2.groovy
EX10_closure3.groovy
EX10_closure4.groovy
EX11_method1.groovy
EX12_method2.groovy
EX13_method3.groovy
EX14_class1.groovy
EX16_xmlsample1.groovy
EX17_xmlsample2.groovy
EX18_jsonsample1.groovy
EX19_jsonsample2.groovy
EX20_jsonsample3.groovy
EX20_jsonsample3.json
EX21_jsonsample4.groovy
EX22_jsonsample5.groovy
EX23_httpbuilder1.groovy
EX24_httpbuilder2.groovy
EX25_restsample1.groovy
EX26_restsample2.groovy
groovyx.jar
run.sh
README.md
gumball-v1
gumball-v2
gumball-v3
hellobooks
hellograils
EX08_closure1.groovy
EX14_class1.groovy
1 class Person {
2
3 String name
4 Integer age
5
6 Person(name, age) {
7 this.name = name
8 this.age = age
9 }
10
11 }
12
13 def person1 = new Person('Marie', 1)
14 def person2 = ['Marie', 2] as Person
15 Person person3 = ['Marie', 3]
16
17
18
19
Line 1, Column 1
Tab Size: 4
Groovy


EX18_jsonsample1.groovy — sjsu
281
aws
bios
docker
grails
groovy
EX01_helloworld.groovy
EX02_helloworld.groovy
EX03_dynamictypes.groovy
EX04_operators.groovy
EX05_stringliterals.groovy
EX06_groovymaps.groovy
EX07_groovyarrays.groovy
EX08_closure1.groovy
EX09_closure2.groovy
EX10_closure3.groovy
EX10_closure4.groovy
EX11_method1.groovy
EX12_method2.groovy
EX13_method3.groovy
EX14_class1.groovy
EX16_xmlsample1.groovy
EX17_xmlsample2.groovy
EX18_jsonsample1.groovy
EX19_jsonsample2.groovy
EX20_jsonsample3.groovy
EX20_jsonsample3.json
EX21_jsonsample4.groovy
EX22_jsonsample5.groovy
EX23_httpbuilder1.groovy
EX24_httpbuilder2.groovy
EX25_restsample1.groovy
EX26_restsample2.groovy
groovyx.jar
run.sh
ZREADME.md
gumball-v1
gumball-v2
gumball-v3
hellobooks
hellograils
EX08_closure1.groovy
EX18_jsonsample1.groovy
1 import groovy.json.JsonBuilder
2
3 def builder = new JsonBuilder()
4 builder.weeks {
5 capacity '8'
6 tasks(
7 [{
8 done '0'
9 total '4'
10 title 'build web service'
11 }, {
12 done '0'
13 total '1'
14 title 'build web service client'
15 }]
16)
17 }
18
19 //println builder.toString()
20 println builder.toPrettyString()
21
22
23
Line 1, Column 1; Detect Indentation: Setting indentation to 2 spaces
Spaces: 2
Groovy

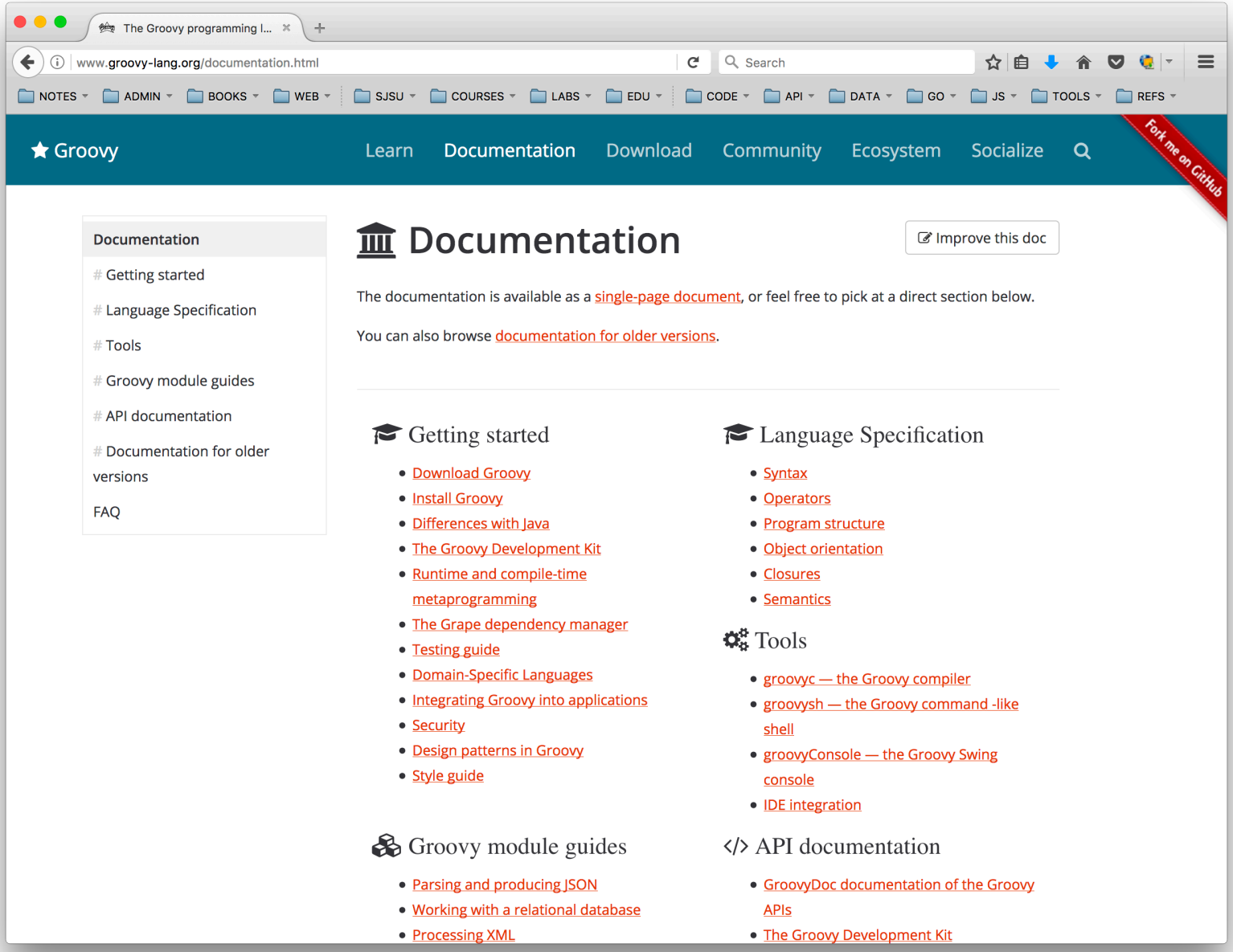

EX19_jsonsample2.groovy — sjsu
281
aws
bios
docker
grails
groovy
EX01_helloworld.groovy
EX02_helloworld.groovy
EX03_dynamictypes.groovy
EX04_operators.groovy
EX05_stringliterals.groovy
EX06_groovymaps.groovy
EX07_groovyarrays.groovy
EX08_closure1.groovy
EX09_closure2.groovy
EX10_closure3.groovy
EX10_closure4.groovy
EX11_method1.groovy
EX12_method2.groovy
EX13_method3.groovy
EX14_class1.groovy
EX16_xmlsample1.groovy
EX17_xmlsample2.groovy
EX18_jsonsample1.groovy
EX19_jsonsample2.groovy
EX20_jsonsample3.groovy
EX20_jsonsample3.json
EX21_jsonsample4.groovy
EX22_jsonsample5.groovy
EX23_httpbuilder1.groovy
EX24_httpbuilder2.groovy
EX25_restsample1.groovy
EX26_restsample2.groovy
groovyx.jar
run.sh
ZREADME.md
gumball-v1
gumball-v2
gumball-v3
hellobooks
hellograils
EX08_closure1.groovy
EX19_jsonsample2.groovy
1
2 import groovy.json.JsonBuilder
3
4 def builder = new JsonBuilder()
5 builder {
6 invoices(1..3) { day ->
7 invoice(date: "2015-01-0$day") {
8 item(count: day) {
9 product(name: 'ULC', dollar: 1499)
10 }
11 }
12 }
13 }
14
15 println builder.toString()
16 //println builder.toPrettyString()
17
18
Line 1, Column 1
Tab Size: 4
Groovy


EX22_jsonsample5.groovy — sjsu
281
aws
bios
docker
grails
groovy
EX01_helloworld.groovy
EX02_helloworld.groovy
EX03_dynamictypes.groovy
EX04_operators.groovy
EX05_stringliterals.groovy
EX06_groovymaps.groovy
EX07_groovyarrays.groovy
EX08_closure1.groovy
EX09_closure2.groovy
EX10_closure3.groovy
EX10_closure4.groovy
EX11_method1.groovy
EX12_method2.groovy
EX13_method3.groovy
EX14_class1.groovy
EX16_xmlsample1.groovy
EX17_xmlsample2.groovy
EX18_jsonsample1.groovy
EX19_jsonsample2.groovy
EX20_jsonsample3.groovy
EX20_jsonsample3.json
EX21_jsonsample4.groovy
EX22_jsonsample5.groovy
EX23_httpbuilder1.groovy
EX24_httpbuilder2.groovy
EX25_restsample1.groovy
EX26_restsample2.groovy
groovyx.jar
run.sh
ZREADME.md
gumball-v1
gumball-v2
gumball-v3
hellobooks
hellograils
EX08_closure1.groovy
EX22_jsonsample5.groovy
/*
http://www.groovy-tutorial.org/basic-json/
BUILDING JSON
*/
import groovy.json.JsonBuilder
def json = new JsonBuilder()
def staffListExport = json.staff {
'1234' {
id 1234
name 'Fred Nurk'
position (
department: 'Accounts',
role: 'Manager'
)
}
}
println json.toPrettyString()
Line 1, Column 1
Tab Size: 4
Groovy

Documentation

GDK

http://www.groovy-lang.org/documentation.html



Some Strings Examples

```
'Hello'.compareToIgnoreCase('hello')     // 0
'Hello'.concat('world')                  // Hello world
'Hello'.endsWith('lo')                   // true
'Hello'.equalsIgnoreCase('hello')        // true
'Hello'.indexOf('lo')                    // 3
'Hello world'.indexOf('o', 6)            // 7
'Hello'.matches('Hello')                 // true
'Hello'.matches('He')                    // false
'Hello'.replaceAll('l', 'L')             // HeLLo
'Hello world'.split('l')                 // 'He', 'o wor', 'd'
'Hello'.substring(1)                     // ello
'Hello'.substring(1, 4)                  // ell
'Hello'.toUpperCase()                    // HELLO
def message = 'Hello'
message.center(11)                       //□□□Hello□□□
message.center(3)                        // Hello
message.center(11, '#')                  // ###Hello###
message.eachMatch('.') { ch ->           // print H e l l o
  println ch }                             on separate lines
message.getAt(0)                         // H
message.getAt(0..<3)                     // Hel
message.getAt([0, 2, 4])                 // Hlo
message.leftShift('world')               // Hello world
message << 'world'                       // Hello world
message.minus('ell')                     // Ho
message-'ell'                            // Ho
message.padLeft(4)                       // Hello
message.padLeft(11)                      // □□□□□□Hello
message.padLeft(11, '#')                 // ######Hello
message.padRight(4)                      // Hello
message.padRight(11)                     // Hello□□□□□□
message.padRight(11, '#')                // Hello######
message.plus('world')                    // Hello world
message+'world'                          // Hello world
message.replaceAll('[a-z]') { ch ->      // HELLO
  ch.toUpperCase() }
message.reverse()                        // olleH
message.toList()                         // ['H', 'e', 'l', 'l', 'o
def message='Hello world'
message.tokenize()                       // ['Hello', 'world']
message.tokenize('l')                    // ['He', 'o wor', 'd']
```

```
def greeting='Hello world'
'Hello'+'world'              // Hello world       concatenate
'Hello'*3                    // HelloHelloHello   repeat
greeting-'o world'           // Hell              remove first occurrence
greeting.size()              // 11                synonymous with length
greeting.length()            // 11                synonymous with size
greeting.count('o')          // 2
greeting.contains('ell')     // true
```

In a regular expression, two special *positional characters* are used to denote the beginning and end of a line: caret (^) and dollar sign ($):

```
def rhyme='Humpty Dumpty sat on a wall'
rhyme=~'^Humpty'                          // true
rhyme=~'wall$'                            // true
```

Regular expressions can also include *quantifiers*. The plus sign (+) represents one or more times, applied to the preceding element of the expression. The asterisk (*) is used to represent zero or more occurrences. The question mark (?) denotes zero or once. The metacharacter { and } is used to match a specific number of instances of the preceding character. The following all yield `true`:

```
'aaaaab'=~'a*b'
'b'=~'a*b'
'aaacd'=~'a*c?d'
'aaad'=~'a*c?d'
'aaaaab'=~'a{5}b'
!('aab'=~'a{5}b')
```

Collections

Example	*Description*
`[11, 12, 13, 14]`	A list of integer values
`['Ken', 'John', 'Andrew']`	A list of `String`s
`[1, 2, [3, 4], 5]`	A nested list
`['Ken', 21, 1.69]`	A heterogeneous list of object references
`[ ]`	An empty list

```
def numbers = [11, 12, 13, 14]     // list with four items
numbers [0]                        // 11
numbers [3]                        // 14

numbers [-1]   // 14
numbers [-2]   // 13

[11, 12, 13, 14][2]   // 13

numbers [0..2]    // [11, 12, 13]
numbers [1..<3]   // [12, 13]

numbers [1] = 22          // [11, 22, 13, 14]
numbers [1] = [33, 44]    // [11, [33, 44], 13, 14]

numbers << 15     // [11, [33, 44], 13, 14, 15]

numbers = [11, 12, 13, 14]     // list with four items
numbers + [15, 16]             // [11, 12, 13, 14, 15, 16]

numbers = [11, 12, 13, 14]     // list with four items
numbers - [13]                 // [11, 12, 14]
```

TABLE 4.2 List methods

Name	*Signature/description*
`add`	`boolean add(Object value)` Append the new `value` to the end of this `List`.
`add`	`void add(int index, Object value)` Inserts a new `value` into this `List` at the given `index` position.
`addAll`	`boolean addAll(Collection values)` Append the new `values` on to the end of this `List`.
`contains`	`boolean contains(Object value)` Returns `true` if this `List` contains the specified `value`.
`flatten` *	`List flatten()` Flattens this `List` and returns a new `List`.
`get`	`Object get(int index)` Returns the element at the specified position in this `List`.

```
[11, 12, 13, 14].add(15)                  // [11, 12, 13, 14, 15]
[11, 12, 13, 14].add(2, 15)               // [11, 12, 15, 13, 14]
[11, 12, 13, 14].add([15, 16])            // [11, 12, 13, 14, 15, 16]
[11, 12, 13, 14].get(1)                   // 12
[11, 12, 13, 14].isEmpty()                // false
[14, 13, 12, 11].size()                   // 4
[11, 12, [13, 14]].flatten()              // [11, 12, 13, 14]
[11, 12, 13, 14].getAt(1)                 // 12
[11, 12, 13, 14].getAt(1..2)              // [12, 13]
[11, 12, 13, 14].getAt([2, 3])            // [13, 14]
[11, 12, 13, 14].intersect([13, 14, 15])  // [13, 14]
[11, 12, 13, 14].pop()                    // 14
[11, 12, 13, 14].reverse()                // [14, 13, 12, 11]
[14, 13, 12, 11].sort()                   // [11, 12, 13, 14]
```

Maps

Example	*Description*
`['Ken' : 'Barclay', 'John' : 'Savage']`	Forename/surname collection
`[4 : [2], 6 : [3, 2], 12 : [6, 4, 3, 2]]`	Integer keys and their list of divisors
`[ : ]`	Empty map


```
def names = ['Ken' : 'Barclay', 'John' : 'Savage']
def divisors = [4 : [2], 6 : [3, 2], 12 : [6, 4, 3, 2]]
names['Ken']                                          // 'Barclay'
names.Ken                                             // 'Barclay'
names['Jessie']                                       // null
divisors[6]                                           // [3, 2]
```

```
divisors[6] = [6, 3, 2, 1]              // [4 : [2], 6 : [6, 3, 2, 1],
                                        // 12 : [6, 4, 3, 2]]
```

```
def careful = [ 1 : 'Ken', '1' : 'Barclay']
careful[1]                                  // Ken
careful['1']                                // Barclay
```

TABLE 4.4 Map methods

Name	*Signature/description*
`containsKey`	`boolean containsKey(Object key)` Does this `Map` contain this `key`?
`get`	`Object get(Object key)` Look up the `key` in this `Map` and return the corresponding value. If there is no entry in this `Map` for the key, then return `null`.
`get` *	`Object get(Object key, Object defaultValue)` Look up the `key` in this `Map` and return the corresponding value. If there is no entry in this `Map` for the key, then return the `defaultValue`.
`getAt` *	`Object getAt(Object key)` Support method for the subscript operator.
`keySet`	`Set keySet()` Obtain a `Set` of the keys in this `Map`.
`put`	`Object put(Object key, Object value)` Associates the specified `value` with the specified `key` in this `Map`. If this `Map` previously contained a mapping for this `key`, the old value is replaced by the specified `value`.
`putAt` *	`Object putAt(Object key, Object value)` Support method to allow `Maps` to operate with subscript assignment.
`size`	`int size()` Returns the number of key-value mappings in this `Map`.
`values`	`Collection values()` Returns a collection view of the values contained in this `Map`.

```
def mp = ['Ken' : 2745, 'John' : 2746, 'Sally' : 2742]
mp.put('Bob', 2713)          // [Bob:2713, Ken:2745, Sally:2742, John:2746]
mp.containsKey('Ken')        // true
mp.get('David', 9999)        // 9999
mp.get('Sally')              // 2742
mp.get('Billy')              // null
mp.keySet()                  // [David, Bob, Ken, Sally, John]
mp.size()                    // 4
mp['Ken']                    // 2745
```

Methods

```
def methodName() {
   // Method code goes here
}
```

```
def methodName(para1, para2, para3) {
  // Method code goes here
}
```

```
def someMethod(para1, para2=0, para3=0) {
  // Method code goes here
}
```

```
def greetings() {
    print 'Hello'
    print ' and '
    println 'welcome'
}

greetings()
```

```
def greetings(name) {
  println "Hello and welcome, ${name}"
}

greetings('John')
```

```
def greetings(salutation, name='Ken') {
  println "${salutation} ${name}"
}
greetings('Hello', 'John')        // Hello John
greetings('Welcome')              // Welcome Ken
```

```
Hello and welcome
```

```
Hello and welcome, John
```

```
Hello John
Welcome Ken
```

Closures - Invocation & Parameters

9.1 CLOSURES

The syntax for defining a closure is:

```
{comma-separated-formal-parameter-list -> statement-list}
```

If no formal parameters are required, then the parameter `List` and the `->` separator are omitted. Here is a simple example of a closure with no parameters.

```
def clos={println 'Hello world'}
clos.call()
```

```
Hello world
```

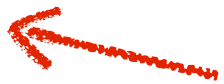
```
def clos={param -> println "Hello ${param}"}

clos.call('world')        // actual argument is 'world'
clos.call('again')        // actual argument is 'again'
clos('shortcut')          // abbreviated form
```

```
Hello world
Hello again
Hello shortcut
```

Observe the third invocation in which the `call` has been omitted.

The next illustration repeats the previous example and produces the same result, but shows that an implicit single parameter referred to as `it` can be used.

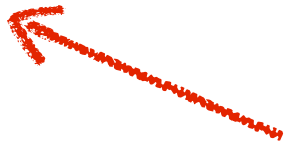
```
def clos={println "Hello ${it}"}

clos.call('world')
clos.call('again')

clos('shortcut')
```



Default Single Parameter "it"

Closures - Enclosing Scope

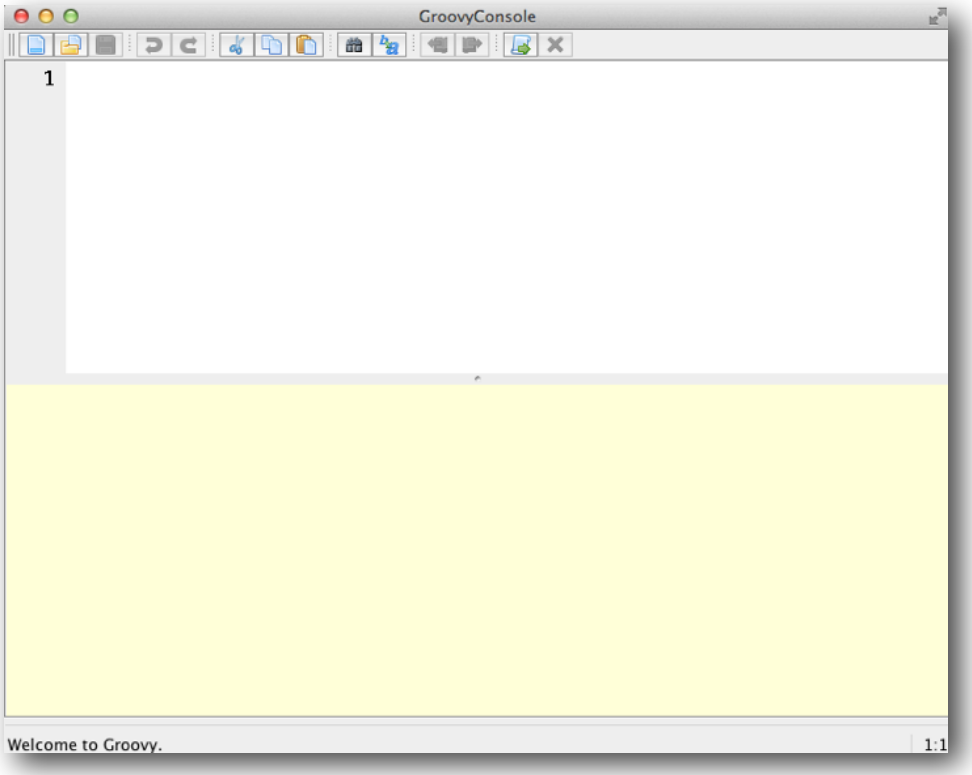

```
def greeting = 'Hello'
def clos = {param -> println "${greeting} ${param}"}
clos.call('world')

// Now show that changes to this variable change the closure.
greeting = 'Welcome'
clos.call('world')

def demo(clo) {
    def greeting = 'Bonjour'                    // does not affect closure
    clo.call('Ken')
}

demo(clos)
```

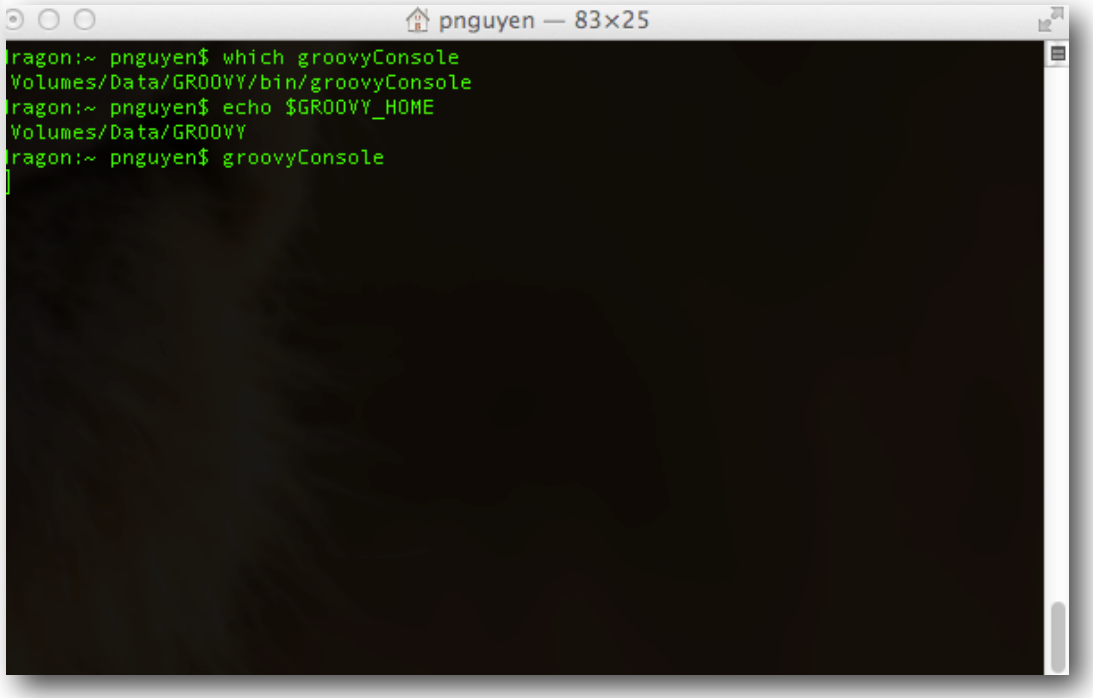


The resulting output is:

```
Hello world
Welcome world
Welcome Ken
```

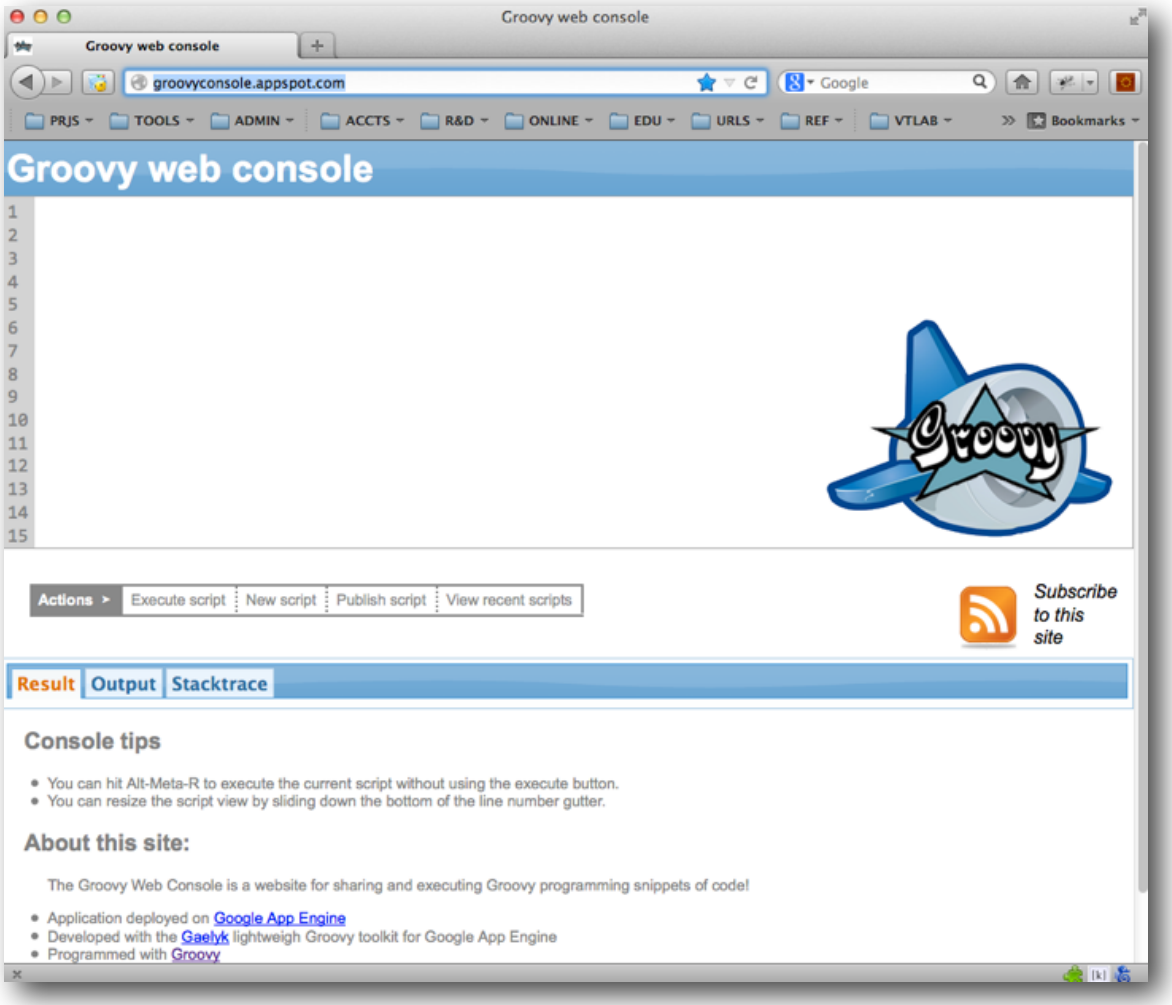
Groovy Console



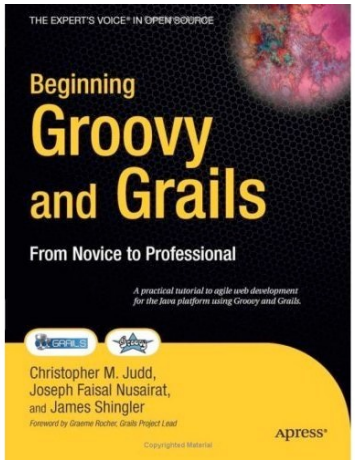


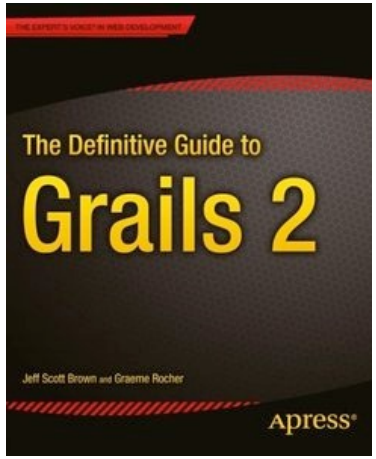
Groovy Web Console

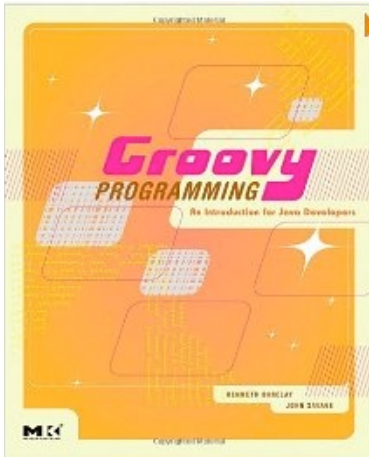
http://groovyconsole.appspot.com/

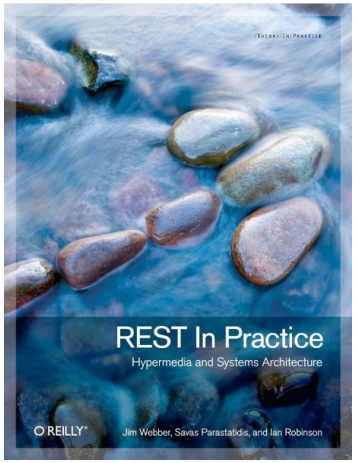


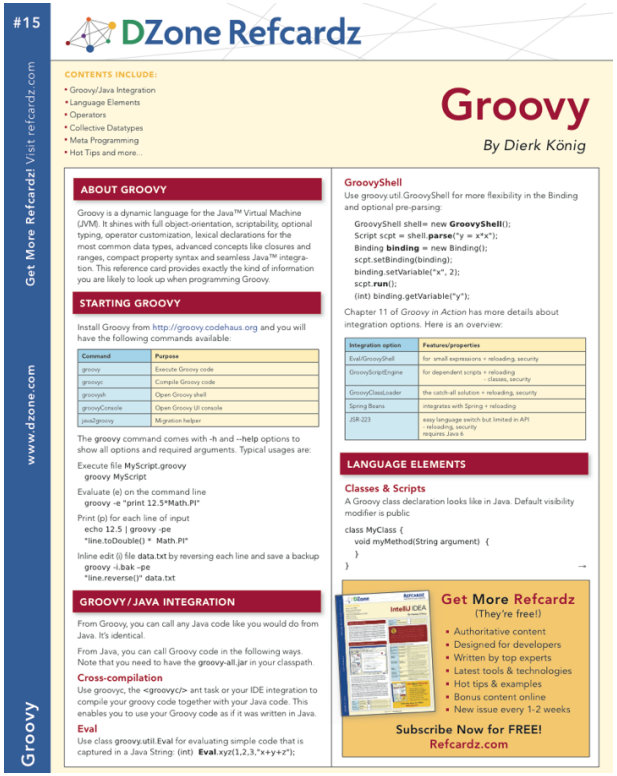
References

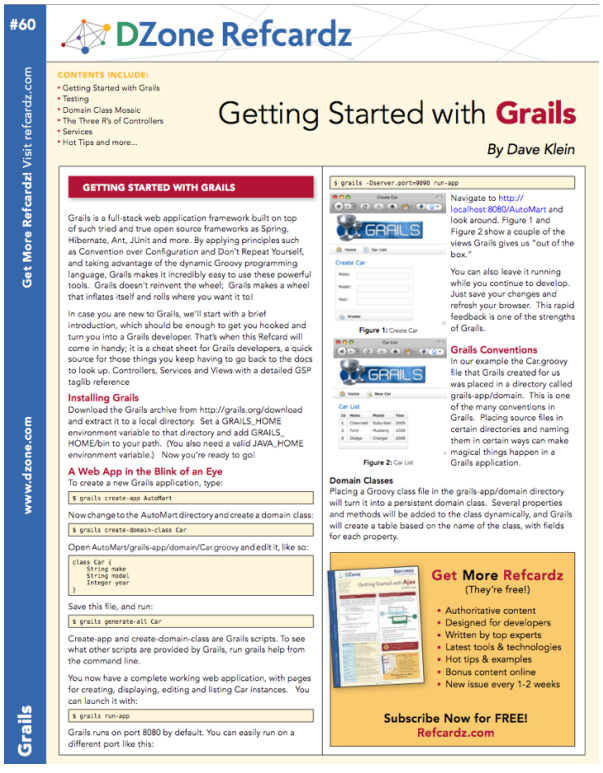

THE EXPERT'S VOICE® IN OPEN SOURCE
Beginning
Groovy
and Grails
From Novice to Professional
A practical tutorial to agile web development for the Java platform using Groovy and Grails.
Christopher M. Judd,
Joseph Faisal Nusairat,
and James Shingler
Foreword by Graeme Rocher, Grails Project Lead
Apress®


The Definitive Guide to
Grails 2
Jeff Scott Brown and Graeme Rocher
Apress®


Groovy
PROGRAMMING
An Introduction for Java Developers


REST In Practice
Hypermedia and Systems Architecture
O'REILLY®
Jim Webber, Savas Parastatidis, and Ian Robinson


#15
DZone Refcardz
Groovy
By Dierk König
CONTENTS INCLUDE:
• Groovy/Java Integration
• Language Elements
• Operators
• Collective Datatypes
• Meta Programming
• Hot Tips and more...
ABOUT GROOVY
STARTING GROOVY
GROOVY / JAVA INTEGRATION
GroovyShell
LANGUAGE ELEMENTS
Classes & Scripts
Get More Refcardz
(They're free!)
Authoritative content
Designed for developers
Written by top experts
Latest tools & technologies
Hot tips & examples
Bonus content online
New issue every 1-2 weeks
Subscribe Now for FREE!
Refcardz.com
Get More Refcardz! Visit refcardz.com
www.dzone.com
Groovy


#60
DZone Refcardz
Getting Started with Grails
By Dave Klein
CONTENTS INCLUDE:
• Getting Started with Grails
• Testing
• Domain Class Mosaic
• The Three R's of Controllers
• Services
• Hot Tips and more...
GETTING STARTED WITH GRAILS
Installing Grails
A Web App in the Blink of an Eye
Domain Classes
Grails Conventions
Get More Refcardz
(They're free!)
Authoritative content
Designed for developers
Written by top experts
Latest tools & technologies
Hot tips & examples
Bonus content online
New issue every 1-2 weeks
Subscribe Now for FREE!
Refcardz.com
Get More Refcardz! Visit refcardz.com
www.dzone.com
Grails