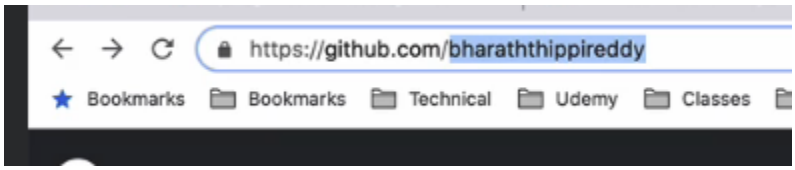


Build

Compile

Run Tests

Package Jar War

Deploy and Run

Ant		Maven
build.xml		Convention Over Configuration
Tasks		MyProject
	+	src/main/java
Customize		src/main/resources
	Gradle	src/test/java
Ivy	DSL	src/test/resouces
	Groovy or Kotlin	
	build.gradle	In-Built pom.xml

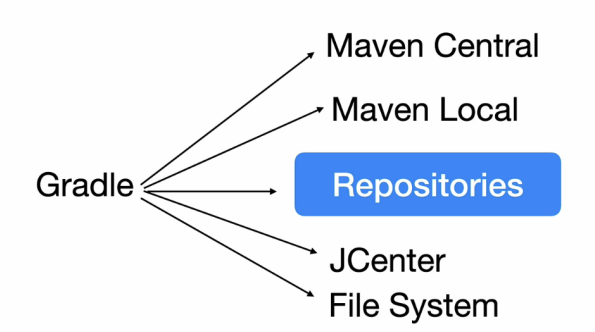
Ant is project to project differs when migrate we have to learn a lot then implement
Maven is great on the other hand but customization is that easy
Gradle covers both disadvantages.

Why

Java Swift

C++ Groovy

Kotlin



Gradle pulls dependencies from these remote repositories and cache them locally.

It provide custom scope. Whereas maven provide fixed scope

Incremental Builds

Build Cache

Daemon

Due to the above points, gradle is fast. It compiles only recently changed files.

Plugin

jar

war

jacoco

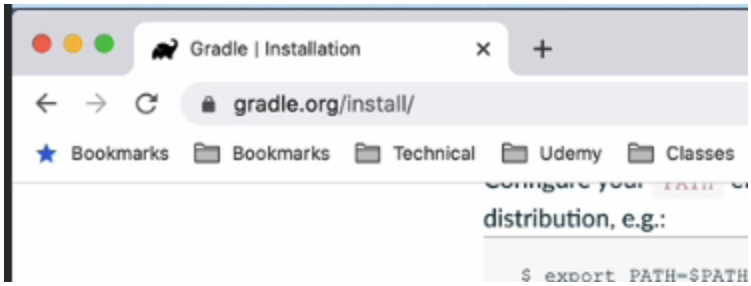
bootRun

DSL

build.gradle groovy/kotlin

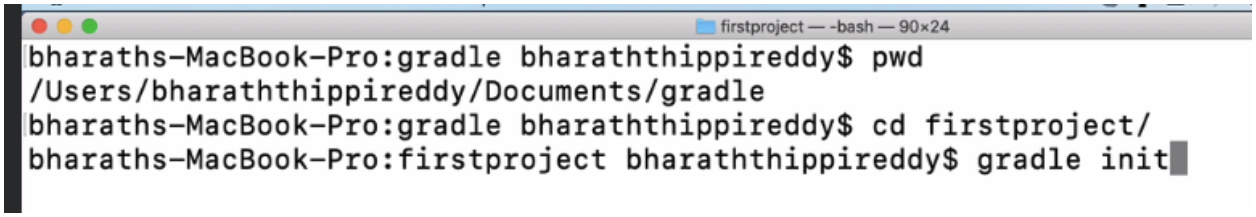
CLI **IDE**

Install Gradle: the below site give all possible options (manual/automated/windows/mac/ linux)




```
bharaths-MacBook-Pro:firstJavaProject bharaththippireddy$ gradle -v

------------------------------------------------------------
Gradle 6.5.1
------------------------------------------------------------

Build time:   2020-06-30 06:32:47 UTC
Revision:     66bc713f7169626a7f0134bf452abde51550ea0a

Kotlin:       1.3.72
Groovy:       2.5.11
Ant:          Apache Ant(TM) version 1.10.7 compiled on September 1 2019
JVM:          14.0.1 (Oracle Corporation 14.0.1+14)
OS:           Mac OS X 10.15.6 x86_64

bharaths-MacBook-Pro:firstJavaProject bharaththippireddy$
```

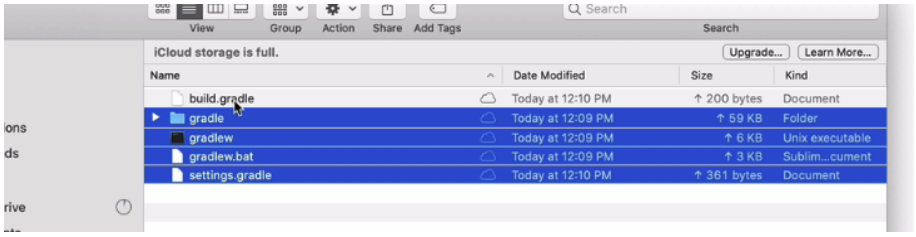
How to create a build.gradle file.

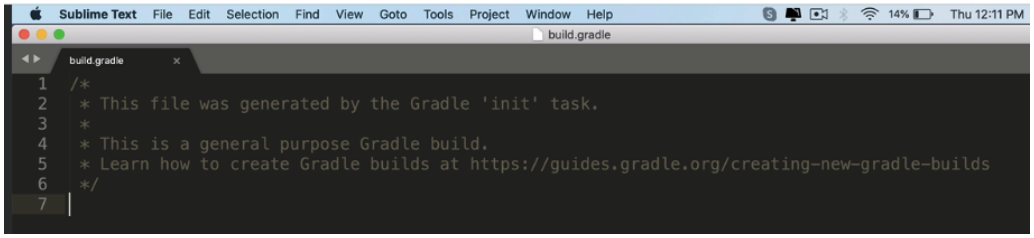


It will ask few questions

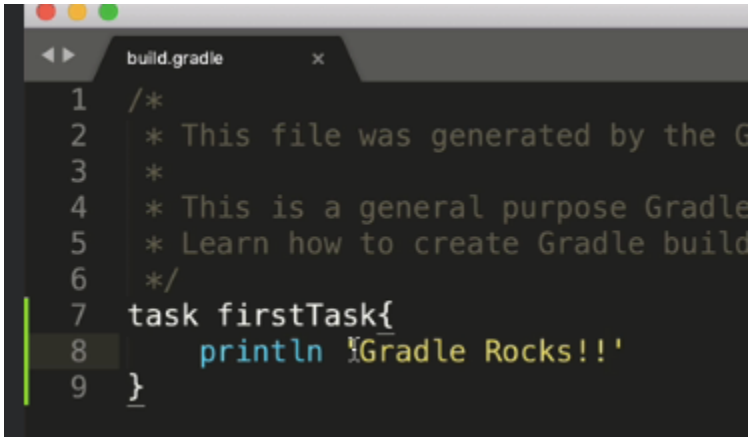
```
Select type of project to generate:
  1: basic
  2: application
  3: library
  4: Gradle plugin
Enter selection (default: basic) [1..4] 1

Select build script DSL:
  1: Groovy
  2: Kotlin
Enter selection (default: Groovy) [1..2] 1

Project name (default: firstproject):


> Task :init
Get more help with your project: https://guides.gradle.org/creating-new-gradle-builds

BUILD SUCCESSFUL in 53s
2 actionable tasks: 2 executed
bharaths-MacBook-Pro:firstproject bharaththippireddy$
```

It creates lot of files

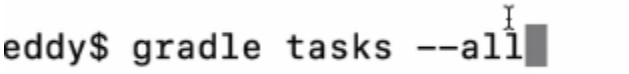


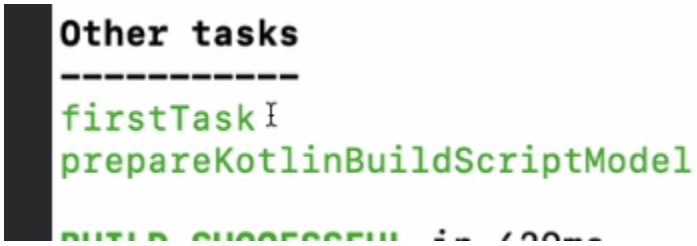


Create a task as follow

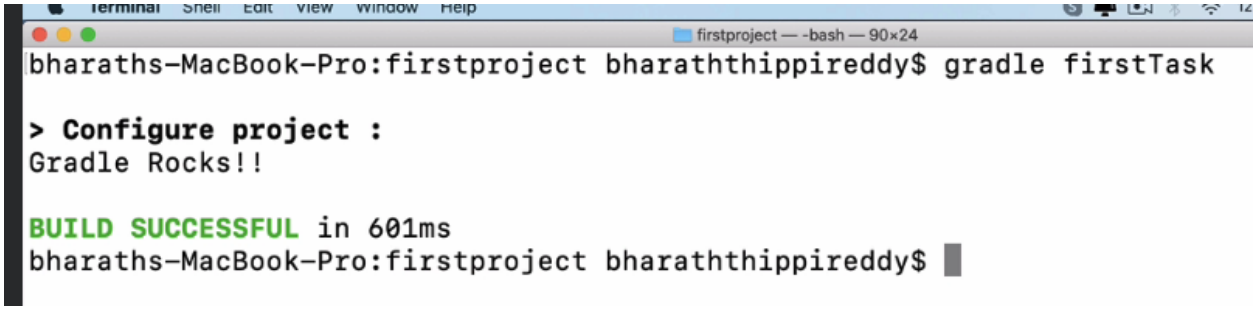


Run the task as follow

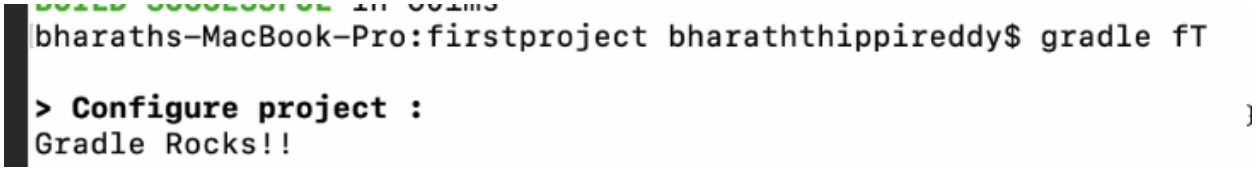




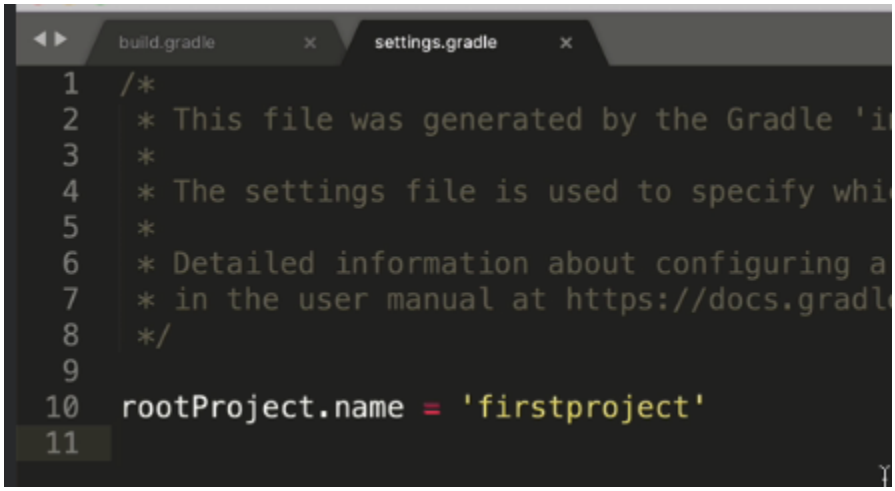
Or run as follow

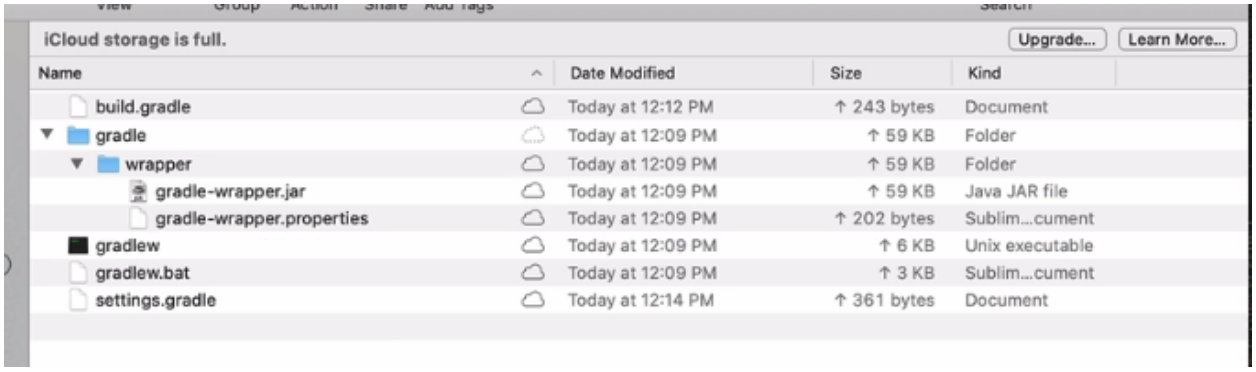


Or Task Abbreviated form

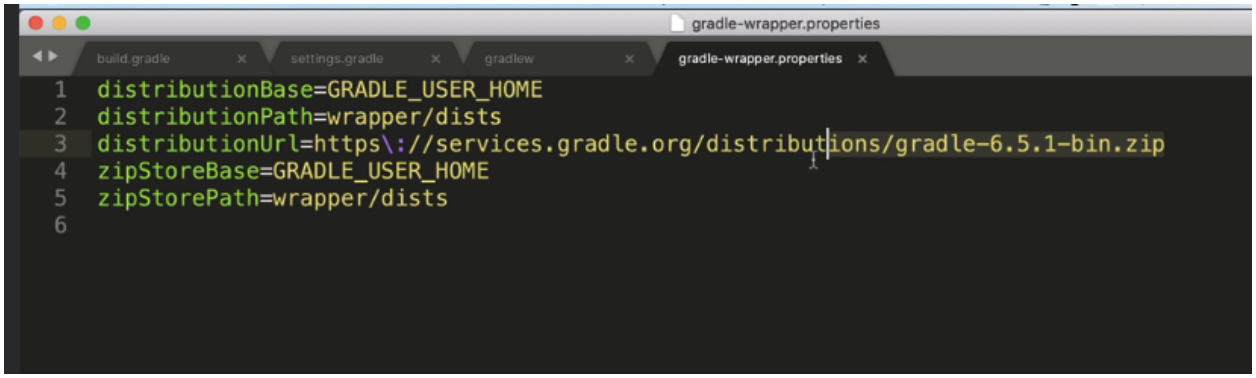


Where we find the project name. We can also change its value.





Gradlew unix executable and gradlew.bat windows executable

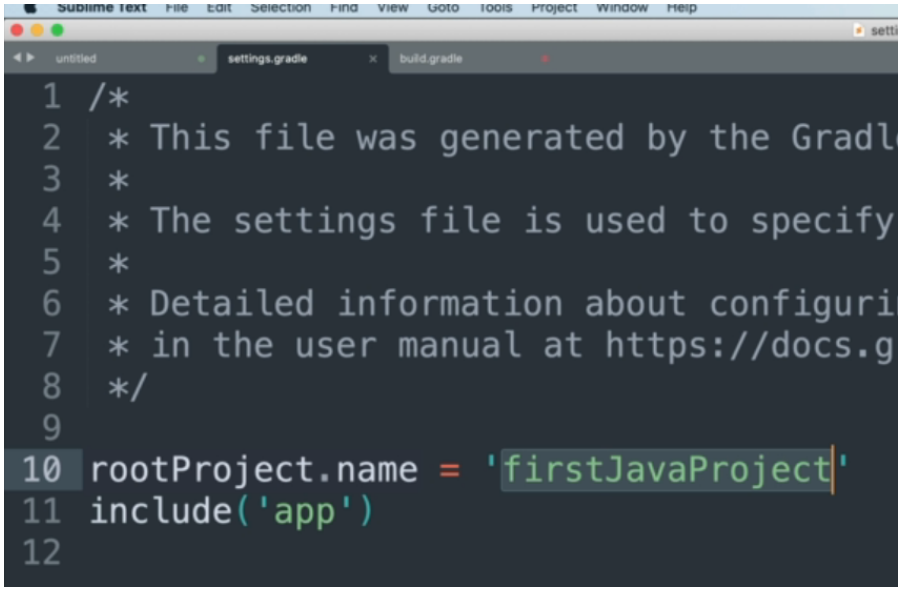


Here is the place where the gradle version of the project is maintained. In the PCF and servers whatever gradle version is available will be ignored and pull the project specific version using gradle-wrapper.jar file.

Similarly create Java Project

```
bharaththippireddy@bharaths-MacBook-Pro-2 gradle % cd firstJavaProject
bharaththippireddy@bharaths-MacBook-Pro-2 firstJavaProject % gradle init

Select type of project to generate:
  1: basic
  2: application
  3: library
  4: Gradle plugin
Enter selection (default: basic) [1..4] 2

Select implementation language:
  1: C++
  2: Groovy
  3: Java
  4: Kotlin
  5: Scala
  6: Swift
Enter selection (default: Java) [1..6] 3

Split functionality across multiple subprojects?:
  1: no - only one application project
  2: yes - application and library projects
Enter selection (default: no - only one application project) [1..2] 1
```

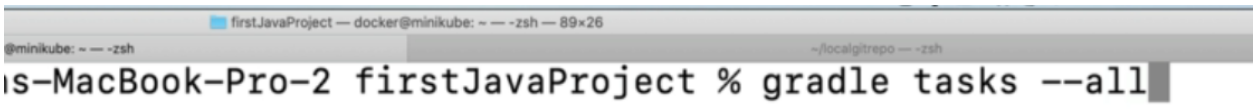
```
Split functionality across multiple subprojects?:
  1: no - only one application project
  2: yes - application and library projects
Enter selection (default: no - only one application project) [1..2] 1

Select build script DSL:
  1: Groovy
  2: Kotlin
Enter selection (default: Groovy) [1..2] 1

Generate build using new APIs and behavior (some features may change in the next minor re

Select test framework:
  1: JUnit 4
  2: TestNG
  3: Spock
  4: JUnit Jupiter
Enter selection (default: JUnit Jupiter) [1..4] 1

Project name (default: firstJavaProject):
Source package (default: firstJavaProject): com.bharath.gradle
```



We can change if we want another name like the following.

Then in that case we need rename folder “app” to “firstJavaProject”.

The below command lists all the tasks.



```
dependencyInsight - Displays the insight into a specific dep
tJavaProject'.
firstJavaProject:dependencyInsight - Displays the insight in
roject ':firstJavaProject'.
help - Displays a help message.
firstJavaProject:help - Displays a help message.
javaToolchains - Displays the detected java toolchains.
firstJavaProject:javaToolchains - Displays the detected java
outgoingVariants - Displays the outgoing variants of root pr
firstJavaProject:outgoingVariants - Displays the outgoing va
Project'.
projects - Displays the sub-projects of root project 'firstJ
firstJavaProject:projects - Displays the sub-projects of pro
properties - Displays the properties of root project 'firstJ
firstJavaProject:properties - Displays the properties of pro
tasks - Displays the tasks runnable from root project 'first
played tasks may belong to subprojects).
firstJavaProject:tasks - Displays the tasks runnable from pr

Verification tasks
------------------
firstJavaProject:check - Runs all checks.
firstJavaProject:test - Runs the test suite.

Other tasks
-----------
```

All java tasks clean build and test are all as part of the plugin below.

```
8
9  plugins {
10     // Apply the application plugin to add support for building a CLI applicatio
11     id 'application'
12 }
13
```

And also we have an application node where is the main method reside by running gradle run that will be executed.

```
27 application {
28     // Define the main class for the application.
29     mainClass = 'com.bharath.gradle.App'
30 }
```

```
bharaththippireddy@bharaths-MacBook-Pro-2 firstJavaProject % gradle compileJava

BUILD SUCCESSFUL in 558ms
1 actionable task: 1 executed
bharaththippireddy@bharaths-MacBook-Pro-2 firstJavaProject % gradle clean

BUILD SUCCESSFUL in 453ms
1 actionable task: 1 executed
bharaththippireddy@bharaths-MacBook-Pro-2 firstJavaProject % gradle test

BUILD SUCCESSFUL in 1s
3 actionable tasks: 3 executed
bharaththippireddy@bharaths-MacBook-Pro-2 firstJavaProject %
```

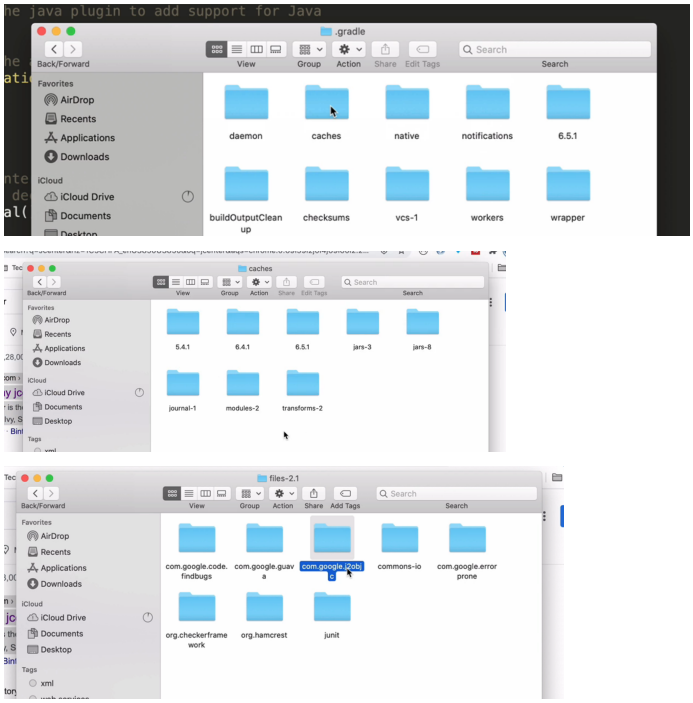
```
bharaththippireddy@bharaths-MacBook-Pro-2 firstJavaProject % gradle jar

BUILD SUCCESSFUL in 477ms
2 actionable tasks: 1 executed, 1 up-to-date
bharaththippireddy@bharaths-MacBook-Pro-2 firstJavaProject % gradle run

> Task :firstJavaProject:run
Hello World!

BUILD SUCCESSFUL in 515ms
2 actionable tasks: 1 executed, 1 up-to-date
bharaththippireddy@bharaths-MacBook-Pro-2 firstJavaProject %
```

```
13
14 repositories {
15     // Use Maven Central for resolving dependencies.
16     mavenCentral()
17 }
18
19 dependencies {
20     // Use JUnit test framework.
21     testImplementation 'junit:junit:4.13.2'
22
23     // This dependency is used by the application.
24     implementation 'com.google.guava:guava:30.1.1-jre'
25 }
26
```

mavenCentral() is the place we download our dependencies.
implementation is for compile and run
testImplementation is for only to run the test.

```
16
17
18 repositories {
19
20     // Use jcenter for resolving dependencies.
21     // You can declare any Maven/Ivy/file repository here.
22     mavenCentral{
23         url=""
24     }
25 }
26
```

Caching location where dependencies and its transitive dependencies are stored.



MavenLocal is our .m2 folder -
It will check dependencies first in mavenCentral if it could not find it, then it will get from jcenter then followed by mavenLocal.

```
repositories {

    // Use jcenter for resolving dependencies.
    // You can declare any Maven/Ivy/file repository here.
    mavenCentral()

    jcenter()

    mavenLocal()
}
```

Scopes

compile

provided

runtime

test

Vs

Configurations

implementation

api (compile)

compileOnly

runtimeOnly

testCompileOnly

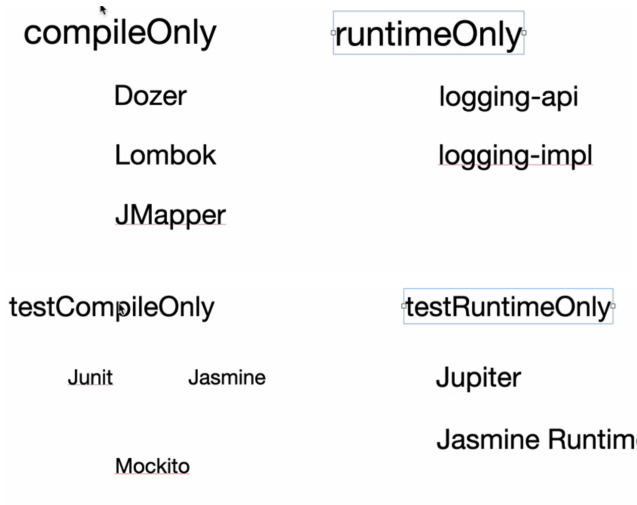
testImplementation

testRuntimeOnly

maven scope

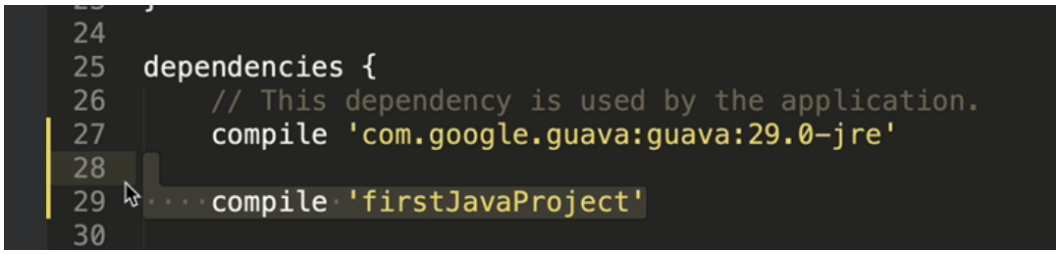
gradle configuration

Mostly used granular level configuration provided below for reference.



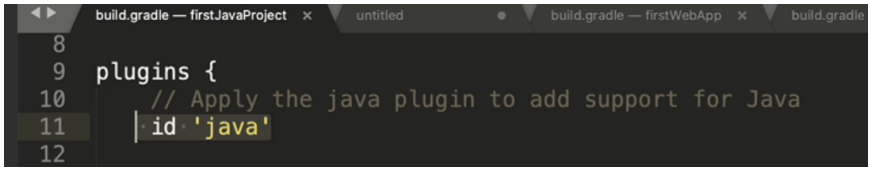
api/compile vs implementation

If we use api/compile as follow the 27 line dependencies were available to project 29 as well.
It is memory leak

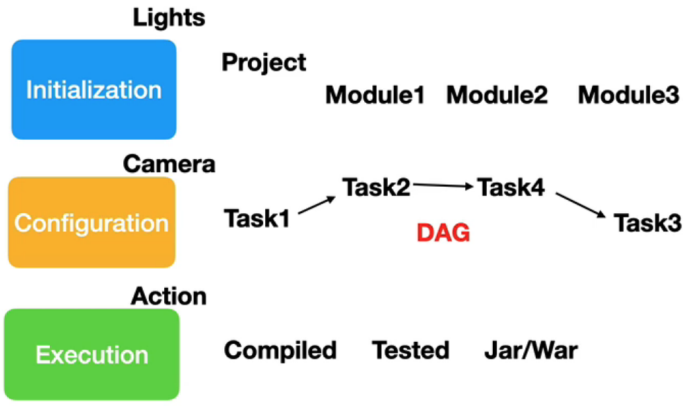


api/compile working as transitive dependencies. Whereas implement work as compile and run.

All the above configuration comes from java plugin



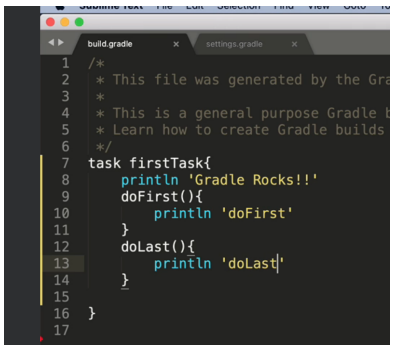
Gradle has 3 phases as shown below.




```
/Users/bharaththippireddy/Documents/gradle/firstproject
bharaths-MacBook-Pro:firstproject bharaththippireddy$ gradle fT

> Configure project :
Gradle Rocks!!
```

When we run our earlier project we see one phase configuration. In that phase it will print all the logs statements

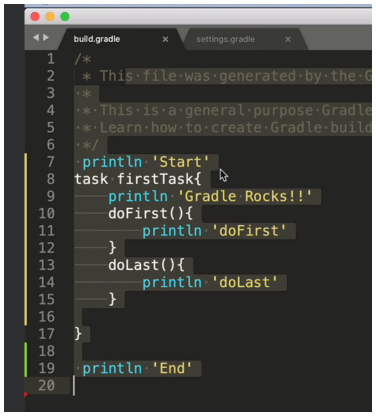
Every task has two methods as shown below.



And its output.

```
BUILD SUCCESSFUL in 560ms
bharaths-MacBook-Pro:firstproject bharaththippireddy$ gradle fT

> Configure project :
Gradle Rocks!!

> Task :firstTask
doFirst
doLast
```

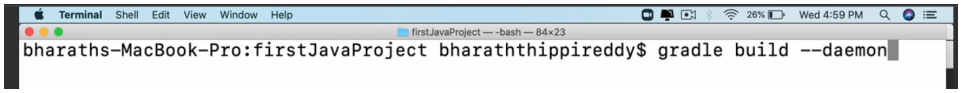
We wont normally use doFirst



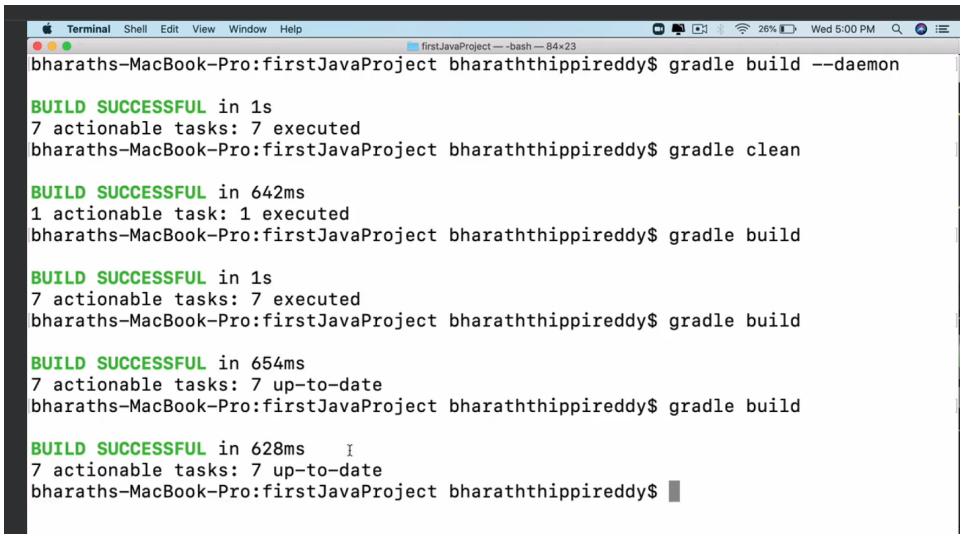
add print statement out of task blocks and notice its output.

```
1 actionable task: 1 executed
bharaths-MacBook-Pro:firstproject bharaththippireddy$ gradle fT

> Configure project :
Start
Gradle Rocks!!
End

> Task :firstTask
doFirst
doLast
```

This will cache all the information



Subsequent bills happen very fast due to cache of information about the project.



We also have not to apply daemon as shown below.

```
bharaths-MacBook-Pro:firstJavaProject bharaththippireddy$ gradle build --no-daemon
To honour the JVM settings for this build a new JVM will be forked. Please consider
using the daemon: https://docs.gradle.org/6.5.1/userguide/gradle_daemon.html.
Daemon will be stopped at the end of the build stopping after processing

BUILD SUCCESSFUL in 4s
7 actionable tasks: 7 up-to-date
bharaths-MacBook-Pro:firstJavaProject bharaththippireddy$ gradle build --no-daemon
```

Maven vs Gradle

We have plugin section in both pom.xml and build.gradle

```xml
<build>
  <plugins>
    <plugin>
      <groupId>org.jvnet.jaxb2.maven2</groupId>
      <artifactId>maven-jaxb2-plugin</artifactId>
      <version>0.14.0</version>
      <executions>
        <execution>
          <goals>
            <goal>generate</goal>
          </goals>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
```

```
plugins {
    // Apply the java plugin to add support for Java
    id 'java'

    // Apply the application plugin to add support for building a CLI application.
    id 'application'
}
```

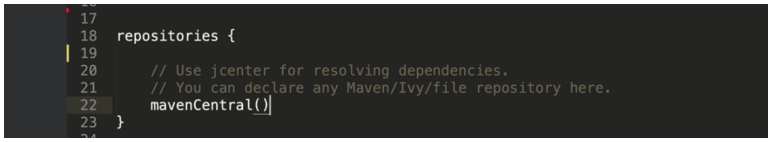
Similarly dependencies and scope

```xml
  <url>http://maven.apache.org</url>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.8.2</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
```

```
dependencies {
    // This dependency is used by the application.
    implementation 'com.google.guava:guava:29.0-jre'

    // Use JUnit test framework
    testImplementation 'junit:junit:4.13'
}
```

Repository

```xml
  <name>Maven Demo</name>
  <url>http://maven.apache.org</url>
```



Good job!

Question 1:

Which of the following plugin gives tasks to run the java application

- java
- run
- **application**
- main

Good job!

Question 2:

The java compile task is added by which of the following plugin

- compile
- jvm
- application
- **java**

Good job!

Question 3:

What should be the dependency scope for junit

- runtime
- implementation
- testcompile
- **testImplementation**

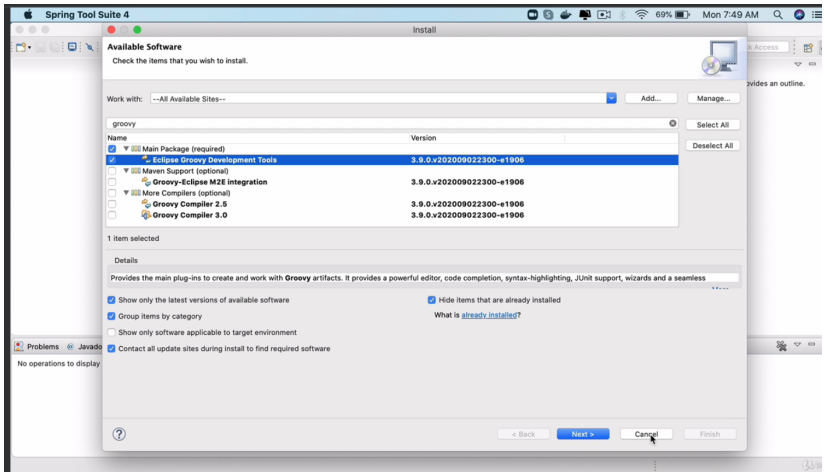
Good job!

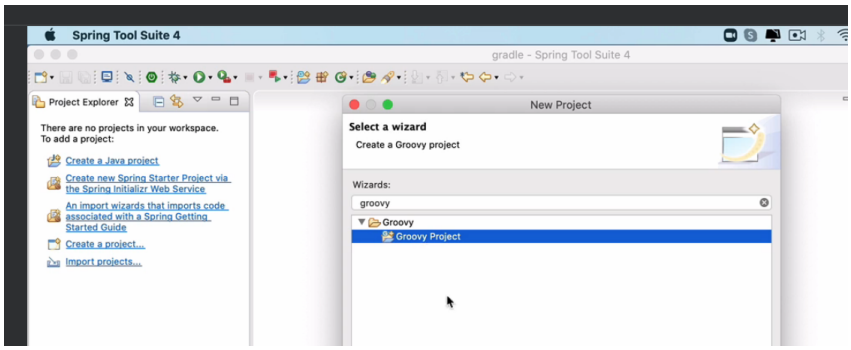
Question 4:

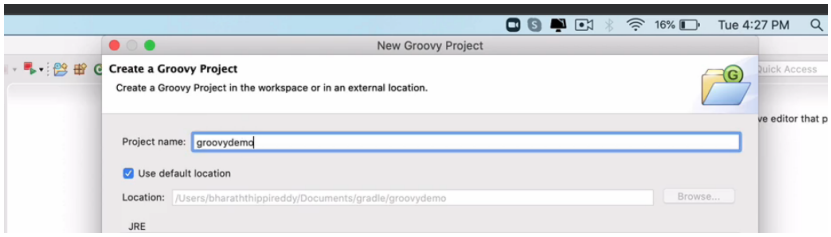
Gradle will run the compile task every-time even if the code has not changded

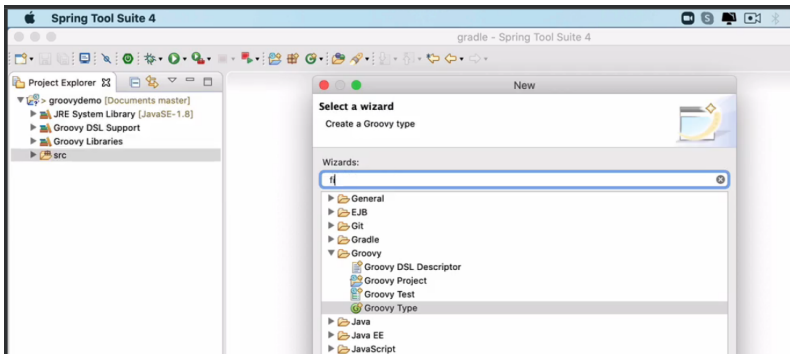
- True
- **False**

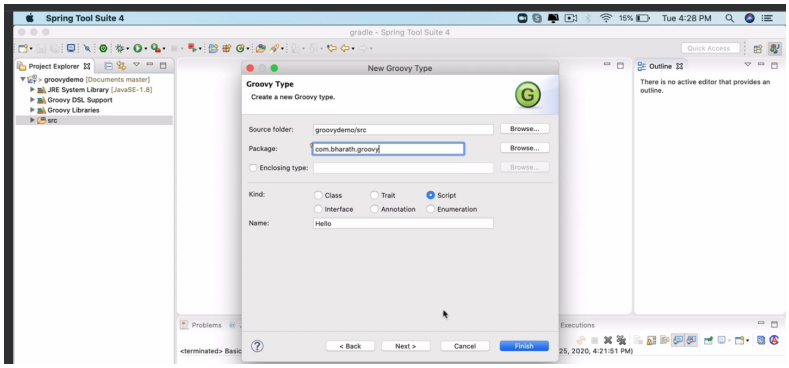
Groovy installation on IDE

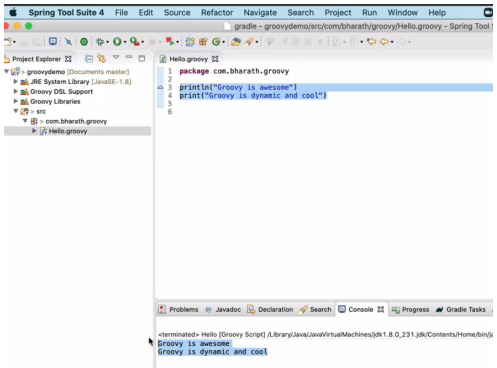

Spring Tool Suite 4
Install
Available Software
Check the items that you wish to install.
Work with: --All Available Sites--
Add...
Manage...
groovy
Select All
Deselect All
Name
Version
Main Package (required)
Eclipse Groovy Development Tools
3.9.0.v202009022300-e1906
Maven Support (optional)
Groovy-Eclipse M2E integration
3.9.0.v202009022300-e1906
More Compilers (optional)
Groovy Compiler 2.5
3.9.0.v202009022300-e1906
Groovy Compiler 3.0
3.9.0.v202009022300-e1906
1 item selected
Details
Provides the main plug-ins to create and work with Groovy artifacts. It provides a powerful editor, code completion, syntax-highlighting, JUnit support, wizards and a seamless
Show only the latest versions of available software
Hide items that are already installed
Group items by category
What is already installed?
Show only software applicable to target environment
Contact all update sites during install to find required software
< Back
Next >
Cancel
Finish
Problems
Javado
No operations to display

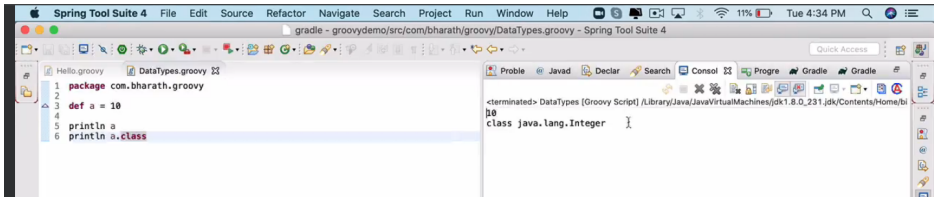

Spring Tool Suite 4
gradle - Spring Tool Suite 4
Project Explorer
There are no projects in your workspace.
To add a project:
Create a Java project
Create new Spring Starter Project via the Spring Initializr Web Service
An import wizards that imports code associated with a Spring Getting Started Guide
Create a project...
Import projects...
New Project
Select a wizard
Create a Groovy project
Wizards:
groovy
Groovy
Groovy Project

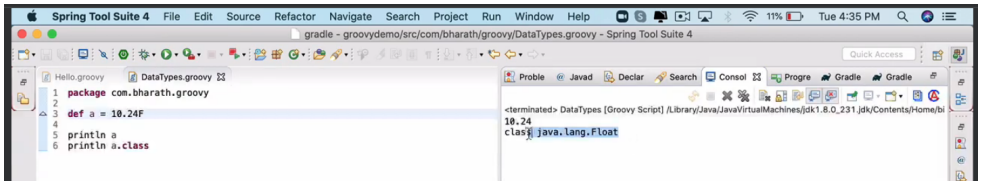

16%
Tue 4:27 PM
New Groovy Project
Create a Groovy Project
Create a Groovy Project in the workspace or in an external location.
Project name: groovydemo
Use default location
Location: /Users/bharaththippireddy/Documents/gradle/groovydemo
Browse...
JRE


Spring Tool Suite 4
gradle - Spring Tool Suite 4
Project Explorer
groovydemo [Documents master]
JRE System Library [JavaSE-1.8]
Groovy DSL Support
Groovy Libraries
src
New
Select a wizard
Create a Groovy type
Wizards:
General
EJB
Git
Gradle
Groovy
Groovy DSL Descriptor
Groovy Project
Groovy Test
Groovy Type
Java
Java EE
JavaScript

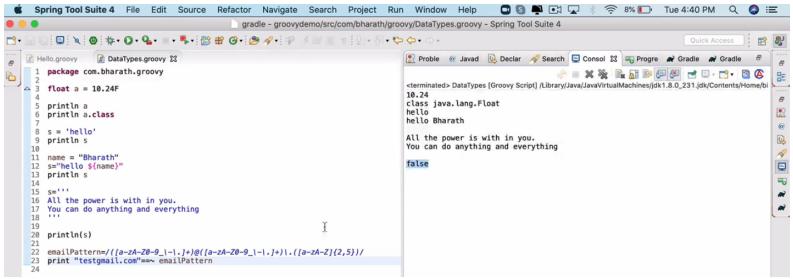




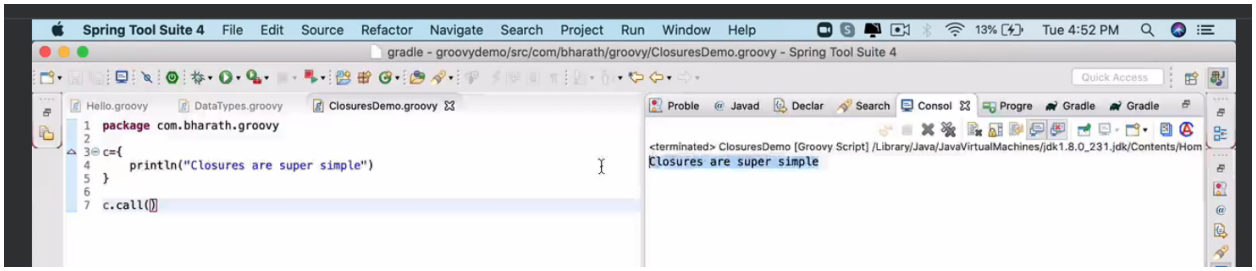


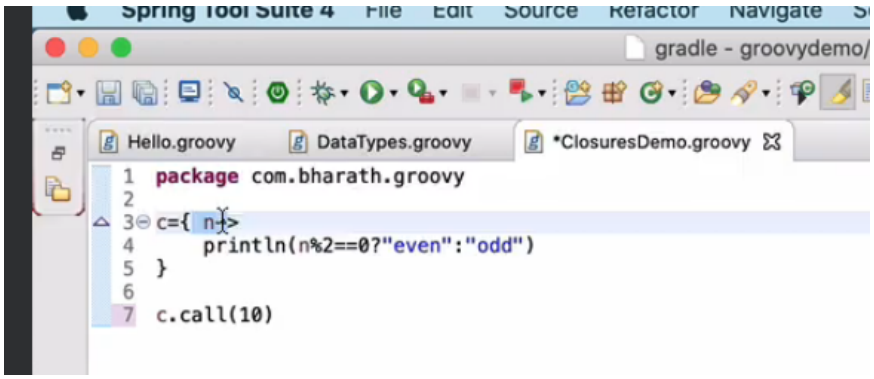


Groovy string at line number 12 and Multiline string starting at line 15 and line number 22 and 23 regular expression. ==~ is find/pattern in regular expression.

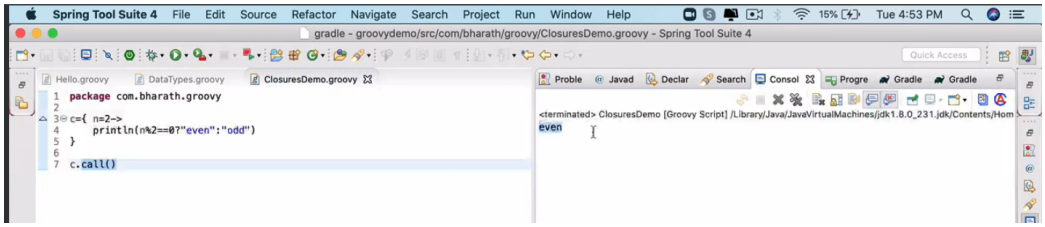


Closure : Lambda expression

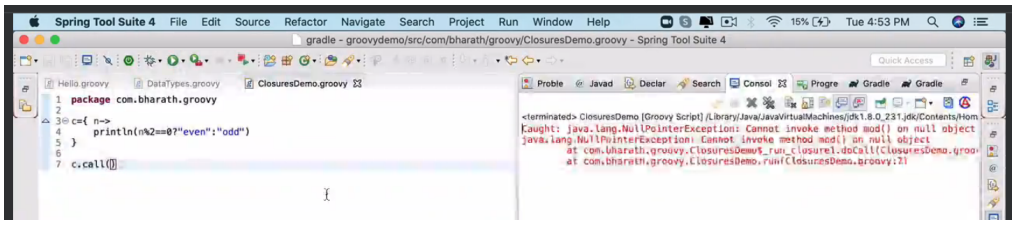




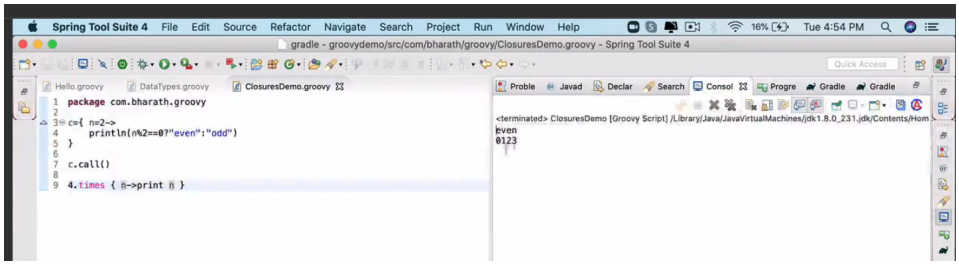
Default value to n = 2



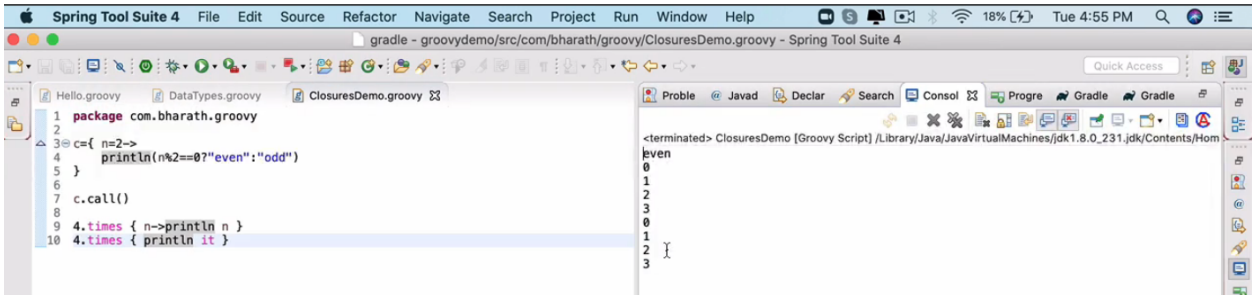
If you do not pass anything then we will get null pointer exception

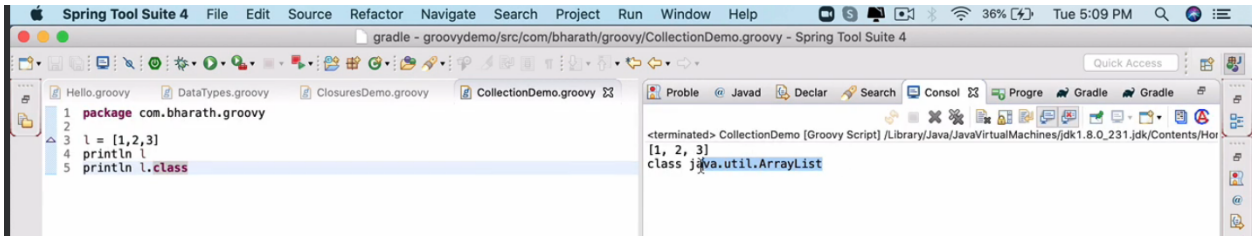


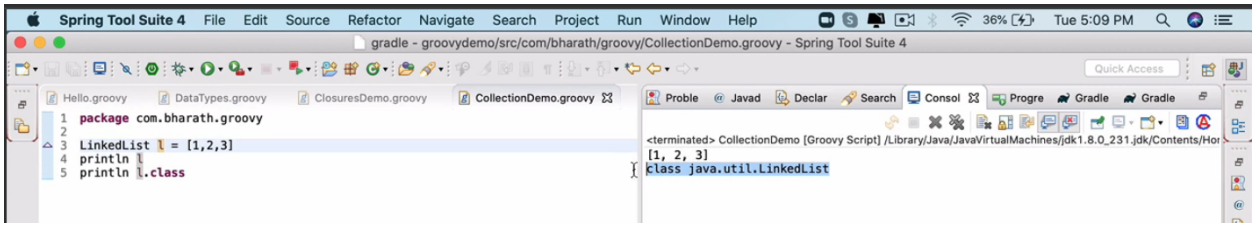
times



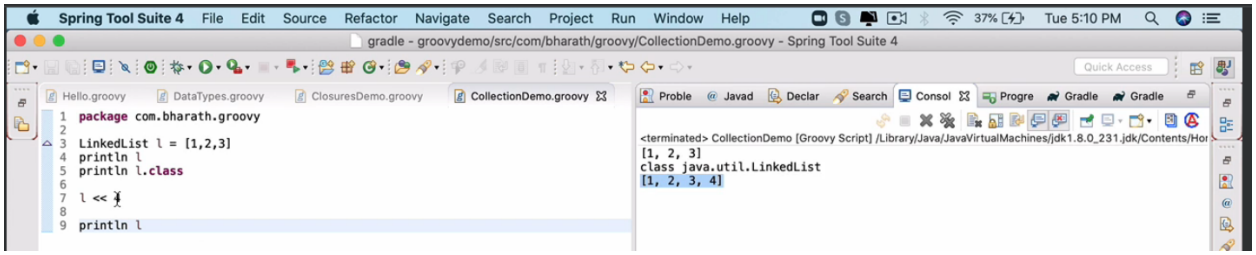
It is same as this keyword in java



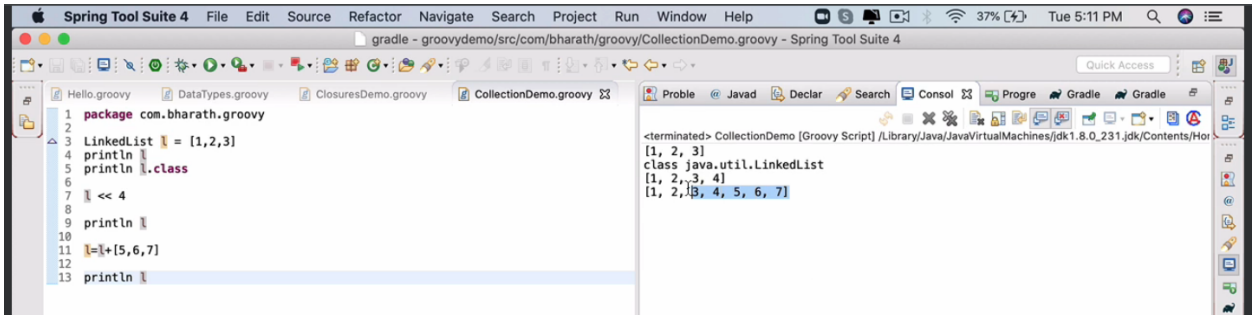




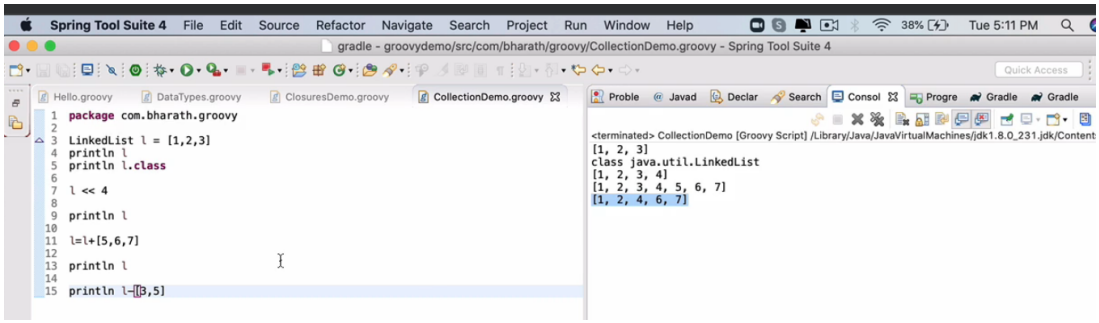
<< operator same as ruby.



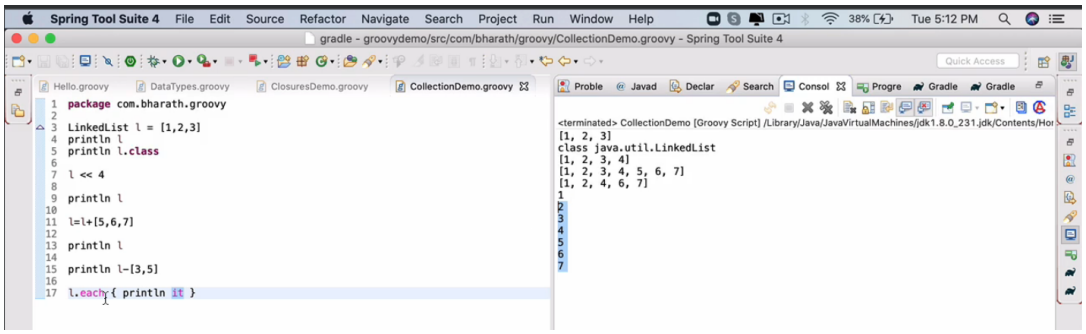
plus

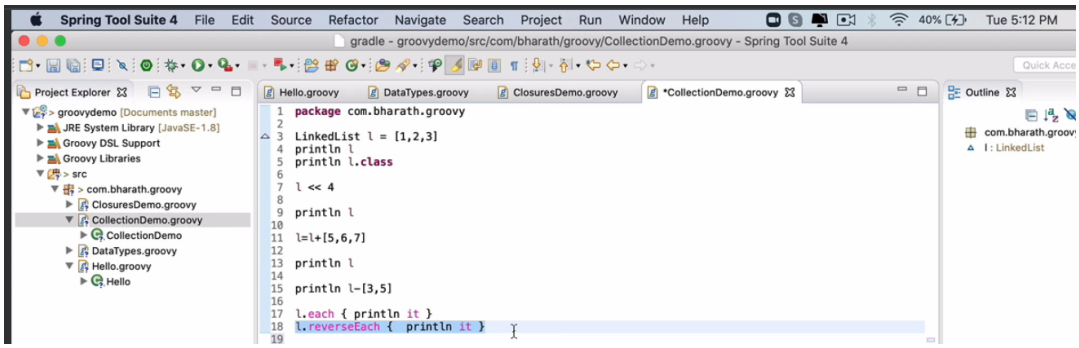


minus

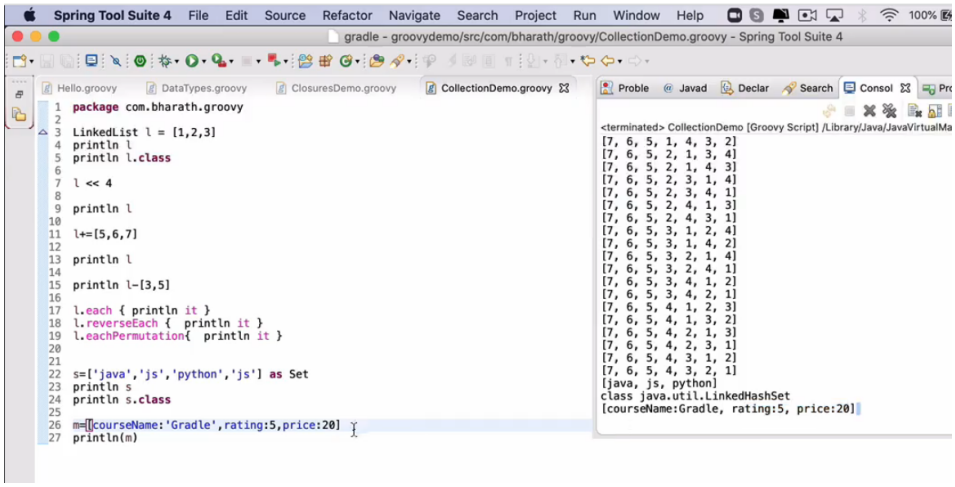


each





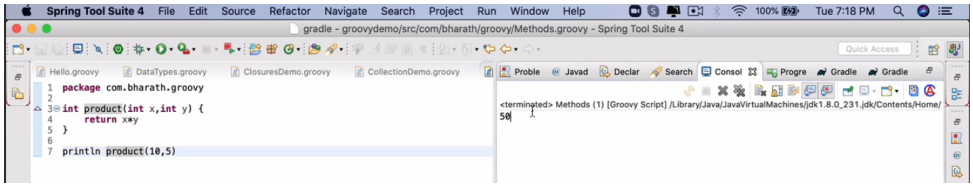
All permutations and similarly for set we have explicitly specify as shown below otherwise it will be treated as arraylist and map.

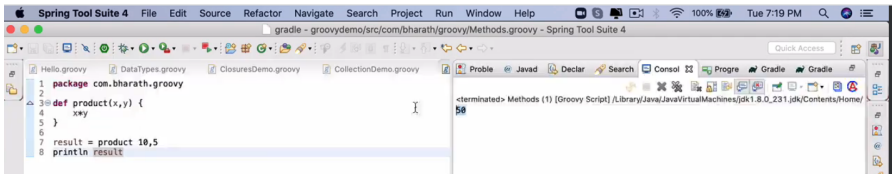


```
println l

l+=[5,6,7]

println l

println l-[3,5]

l.each { println it }
l.reverseEach {  println it }
l.eachPermutation{  println it }


s=['java','js','python','js'] as Set
println s
println s.class

m=[courseName:'Gradle',rating:5,price:20]
println(m)
m.each { k,v->
    println k
    println v
}

println m.courseName
println m['courseName']
println m.get('courseName')

m['review']="Its Awesome"
println m.get('review')
```

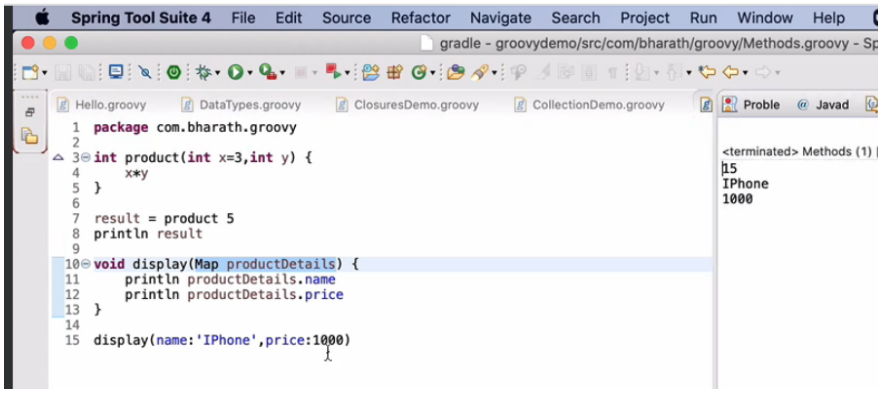
```
[7, 6, 5, 4, 2, 3, 1]
[7, 6, 5, 4, 3, 1, 2]
[7, 6, 5, 4, 3, 2, 1]
[java, js, python]
class java.util.LinkedHashSet
[courseName:Gradle, rating:5, price:20]
courseName
Gradle
rating
5
price
20
Gradle
Gradle
Gradle
Its Awesome
```

Methods

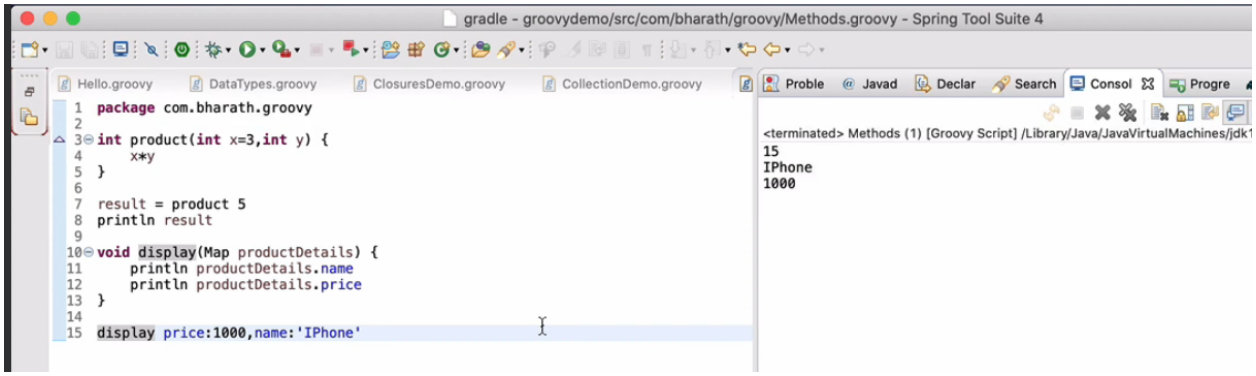




Default value passing to method and passing map example.

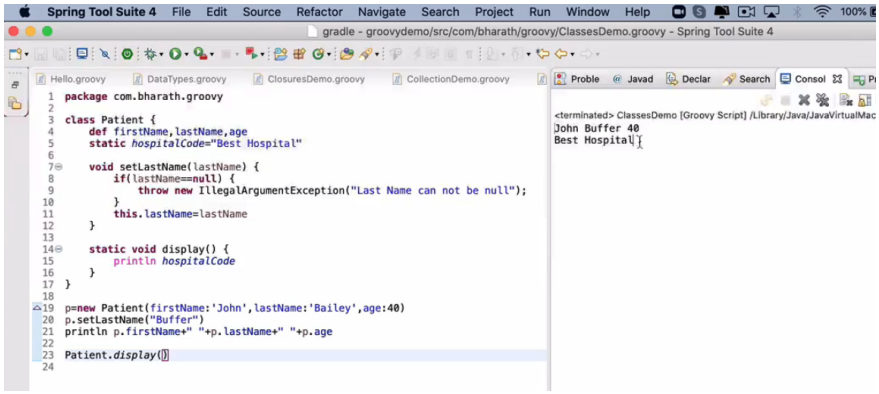


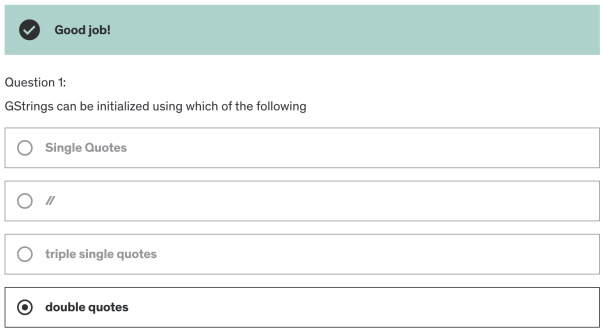
Without parentheses also we can pass

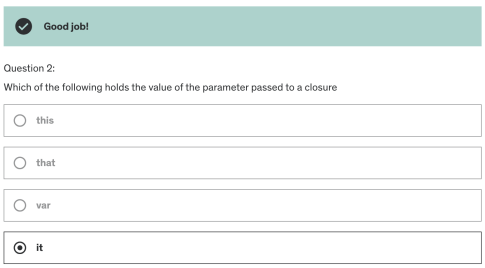


Classes and objects

We do not need to create argument constructor and setters and getters explicitly.
How a static variable is declared and can be called is also provided in the example below.







Good job!

Question 3:

Closures are similar to which of the following in java

- classes
- interfaces
- **lambdas**
- abstract classes

Good job!

Question 4:

Which of the following is used to define a List type

- ()
- **[]**
- {}
- <>

Good job!

Question 5:

Which of the following should be used between each key and value in a map

- ,
- **:**
- ;
- /

Good job!

Question 6:

Groovy supports named parameters using which of the following

- lists
- sets
- tree
- **map**