

# CheatSheet: Jenkins & Groovy

## LANGUAGES

- PDF Link: [cheatsheet-jenkins-groovy-A4.pdf](#), Category: languages
- Blog URL: <https://cheatsheet.dennyzhang.com/cheatsheet-jenkins-groovy-A4>
- Related posts: Jenkins CheatSheet, #denny-cheatsheets

File me Issues or star this repo.

## 1.1 Jenkins Basic

Name	Comment
Deploy Jenkins via docker	<a href="https://hub.docker.com/r/jenkins/jenkins/">https://hub.docker.com/r/jenkins/jenkins/</a>

## 1.2 Jenkins Pipeline

Name	Comment
Specify parameter to run jobs	<code>build job:'job1', parameters:[string(name:'name1', value:va1)]</code>
Run job in different agents	<code>node(\$agent_label) {...}</code>
Ask for user input	<code>stage('stage2'){ input "OK to go?" }</code>
Actively fail current pipeline job	<code>error("Build failed because of this and that..")</code>
Keep going when previous stage has failed	<code>keep-going-with-errors.groovy</code>
Send slack notification in pipeline	<code>slack-notification.groovy</code>
Pass parameter across jenkins jobs	<code>jenkinsfile-pass-parameter.groovy</code>
Set timeout & retry	<code>jenkinsfile-timeout-retry.groovy</code>
Use finally to do cleanup	<code>jenkinsfile-finally.groovy</code>
Run jenkins jobs in a sequential way	<code>jenkinsfile-sequentially.groovy</code>
Run jenkins jobs in parallel	<code>jenkinsfile-parallely.groovy</code>

## 1.3 Config Jenkins Via Groovy

Name	Comment
Set timezone for jenkins	<code>timezone.groovy</code>
Create jenkins views and add jobs to it	<code>jenkins-views.groovy</code>
Configure default view	<code>jenkins-views.groovy</code>
Add default jobs	Copy jobs config.xml to <code>/usr/share/jenkins/ref/jobs/</code>
Jenkins skip wizzard when initialization	<code>-Djenkins.install.runSetupWizard=false</code>
Install jenkins plugins	<code>/usr/local/bin/install-plugins.sh &lt; /usr/share/jenkins/ref/plugins.txt</code>
Configure jenkins slave Node	

## 1.4 Groovy Basic

Name	Comment
Read file content as a variable	<code>def env = System.getenv(), def content = readFile("/tmp/test.txt")</code>
Print stdout	<code>echo 'Action is done', println "Hello World"</code>
Use boolean parameter	<code>if (istru == "false") {...}</code>
Basic integer caculation	<code>def a = 3, b = 7; println "\$a + \$b = \${a + b}"</code>
Get environment variables	<code>get-env.groovy</code>
Run groovy online	SaaS: Groovy Web console
Run groovy script from Jenkins	Link: Jenkins Script Console
Reference	Link: Apache Groovy

## 1.5 Groovy String

Name	Comment
Convert list to string	<code>1.join(";")</code>
Create string with multi-lines	<code>multi-line-string.groovy</code>
Convert string to list	<code>split-string.groovy</code>
Convert string to json	<code>string-to-json.groovy</code>

## 1.6 Groovy Array

Name	Comment
Iterate a list	<code>for(item in [1,2,3,4]){ println item }</code>
Iterate a list	<code>(1..3).each { println "Number \${it}" }</code>
Add item to list	<code>def alist = [10, 9, 8]; alist &lt;&lt; 7</code>
List size	<code>def alist = [10, 9, 8]; alist.size()</code>
Split string with delimiter	<code>'1128-2'.tokenize( '-' )</code>

## 1.7 Groovy Dictionary

Name	Comment
Create a map	<code>def m = ['fruit':'Apple', 'veggie':'Carrot']</code>
Add an item to map	<code>m.put('denny','hello')</code>
Check if key exists	<code>m.containsKey('key1')</code>
Loop a map	<code>loop-map.groovy</code>

## 1.8 Groovy Network

Name	Comment
Get hostname by ip	<code>get-ip-by-hostname.groovy</code>
validate user input: ip address	<code>assert ip_address.matches("\\d{1,3}\\. \\d{1,3}\\. \\d{1,3}\\. \\d{1,3}")</code>

## 1.9 Jenkins Groovy

Name	Command
Print message to stdout	See <code>print.groovy</code>
Get environment variables	See <code>get-env.groovy</code>
Convert string to list	See <code>split-string.groovy</code>
Create string with multi-lines	See <code>multi-line-string.groovy</code>
Loop a map	See <code>loop-map.groovy</code>
Get hostname by ip	See <code>get-ip-by-hostname.groovy</code>
Convert list to string	<code>l.join(";"), link: groovy list</code>

## 1.10 Jenkins Plugins

Name	Command
Backup jenkins	ThinBackup Plugin
Load the ssh key	Credentials Plugin
Add timestamp to job output	Timestamping Plugin
Create dashboard	Dashboard View Plugin
Abort if job takes too long	Build-timeout Plugin
Retry failed a job	Naginator Plugin
Backup job configuration	JobConfigHistory Plugin

## 1.11 More Resources

<http://groovy-lang.org/documentation.html#gettingstarted>

<https://jenkins.io/doc/>

License: Code is licensed under MIT License.