



팁과 요령

Table of Contents

- 팁과 요령
 - 등가와 일치
 - 메소드와 클로저 리턴하기
 - 빌더에서 메소드 호출
 - 스크립트
 - 스크립트 포함시키기
 - 컬렉션 뒤지기
 - 정규표현식으로 텍스트 휘젓기
 - 콘솔에 진행상황 표시
 - 그루비 커맨드라인
 - 커맨드라인 옵션정의하기
 - 그루비 실행기

등가와 일치

그루비 등가
`a == b`
 그루비 일치
`a.is(b)`
 null도 비교 가능

메소드와 클로저 리턴하기

클로저 내부에서는 `return`이 클로저로부터의 `return`을 의미함.
 클로저가 인자로 전달된 메서드나 클로저를 감싸는 메서드에서의 리턴을 의미하지 않음.

```
1 [1,2,3].each {
2   print it; return
3 }
```



빌더에서 메소드 호출

```
1 new NodeBuilder().outer{
2   middle{
3     inner()
4   }
5 }
6
7 //오류
8 def produce(){
9   middle{
10    inner()
11  }
12 }
13
14 //정상
15 def builder = new NodeBuilder()
16
17 builder.outer{
18   produce()
19 }
20
21 def produce(){
22   builder.middle{
23     inner()
24   }
25 }
```



스크립트

스크립트는 한 줄씩 실행되지 않는다.
그루비 스크립트가 실행될 때는 먼저 클래스로 변환되고 클래스가 실행되어짐.
이 변환과정은 자동으로 이루어진다.

스크립트 포함시키기

그루비에서 필요한 다른 스크립트를 포함시키는 방법.

1. 의존하는 스크립트를 클래스로 정의.
2. 의존하는 스크립트를 evaluate로 실행.

```
1 Book books = new Book('Books')
2
3 books.someMethod()
```

컬렉션 뒤지기

```
1 def list = [1,2,3,4,5]
2 Collections.suffle(list)
```

정규표현식으로 텍스트 휘젓기

```
1 def text = ""
2 Scramble the inner characters of words
3 leaving the text still readable for people but not for computers
4 ""
5
6 println text.replaceAll(/Bw+B/){
7     inner ->
8     def list = inner.toList()
9     Collections.shuffle(list)
10    list.join "
11 }
```

콘솔에 진행상황 표시

```
1 def procFile(notify){
2     def names = new File('.').list()
3     names.eachWithIndex {
4         name,i ->
5         notify(i*10/names.size(),name)
6         sleep 50
7     }
8 }
9
10 procFile{
11     filled,info ->
12     print 'b' * 61
13     print '##filled + ':'*(10-filled) + '+'info.padRight(50)
14 }
15
16 /*
17 .....: .classpath
18 .....: .groovy
19 .....: .project
20 .....: .settings
21 .....: bin
22 .....:
23 .....:
24 .....:
25 .....:
26 .....:
27 .....:*/
```

```

28 | #####. src
29 |
30 | */

```

그루비 커맨드라인

표 13.3 groovy 실행 파일의 커맨드라인 옵션

옵션	인자	의미
-c, --encoding	문자 인코딩	파일의 인코딩을 명시
-d, --debug		스택 정보 전체를 출력
-e	실행할 텍스트	한 행으로 만드는 커맨드라인 스크립트
-h, --help		사용법
-i	확장자	실시간 파일 수정
-l	포트	포트를 열고서 입력되는 행들을 처리
-n		한 줄씩 처리
-p		결과를 출력하며 한 줄씩 처리
-v, --version		그루비와 JVM 버전 표시

커맨드라인 옵션정의하기

CLI를 이용하여 스크립트에서 사용할 옵션을 지정하면 분석,오류검사,오류처리를 하고 옵션값을 저장하여 스크립트에서 사용할 수 있게 해줌.

```

1 | def cli = new CliBuilder()
2 | cli.h(longOpt: 'help','usage infomation')

```

parse(arg)메소드를 이용하여 실행 명령을 분석
usage()로 프로그램 사용법을 출력.

그루비 실행기

RootLoader을 이용하여 클래스 패스를 확장 가능함.

```

1 | def loader = this.class.classLoader.rootLoader
2 | loader.addURL(File.toURL())

```