

Groovy 3 New Features

Lambdas and More

Contact Info

Ken Kousen

Kousen IT, Inc.

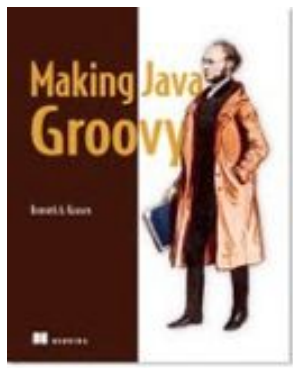
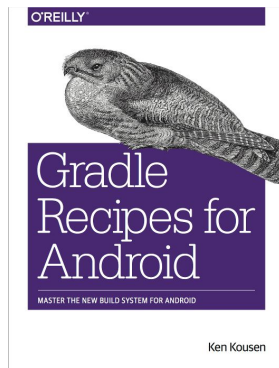
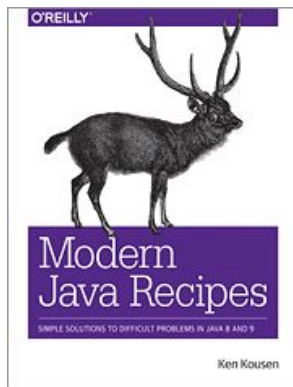
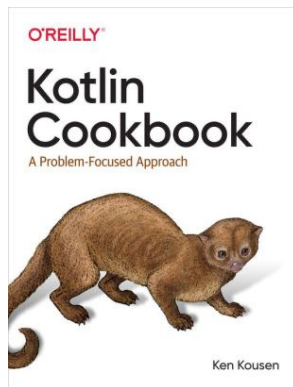
ken.kousen@kousenit.com

<http://www.kousenit.com>

<http://kousenit.org> (blog)

[@kenkousen](https://twitter.com/kenkousen) (twitter)

<https://kenkousen.substack.com> (newsletter)



GitHub Repositories

Most examples in:

https://github.com/kousen/groovy_3_examples

Some others (including cat pictures) in:

<https://github.com/kousen/IntroGroovy>

Release notes for Groovy 3.0

<http://groovy-lang.org/releasesnotes/groovy-3.0.html>

Note: documentation largely by Paul King

Examples in these slides reused from those docs

Parrot Parser

Flexible Parrot parser

- `do-while` loops
- `lambdas`, method references, constructor references
- `try-with-resources`
- Java array initializers
- default methods in interfaces

do-while loop

```
1 // classic Java-style do..while loop
2 def count = 5
3 def fact = 1
4 do {
5     fact *= count--
6 } while(count > 1)
7 assert fact == 120
```

Old-style Java for loop

```
1 def facts = []  
2 def count = 5  
3 for (int fact = 1, i = 1; i <= count; i++, fact *= i) {  
4     facts << fact  
5 }  
6 assert facts == [1, 2, 6, 24, 120]
```

Java-style array initialization

```
1 def primes = new int[] {2, 3, 5, 7, 11}
2 assert primes.size() == 5 && primes.sum() == 28
3 assert primes.class.name == '[I'
4
5 def pets = new String[] {'cat', 'dog'}
6 assert pets.size() == 2 && pets.sum() == 'catdog'
7 assert pets.class.name == '[Ljava.lang.String;'
8
9 // traditional Groovy alternative still supported
10 String[] groovyBooks = [ 'Groovy in Action', 'Making Java Groovy' ]
11 assert groovyBooks.every{ it.contains('Groovy') }
```


Java-style lambda syntax

```
1 (1..10).forEach(e -> { println e })
2
3 assert (1..10).stream()
4         .filter(e -> e % 2 == 0)
5         .map(e -> e * 2)
6         .toList() == [4, 8, 12, 16, 20]
```

Method references

```
1 import java.util.stream.Stream
2
3 // class::staticMethod
4 assert ['1', '2', '3'] ==
5         Stream.of(1, 2, 3)
6                 .map(String::valueOf)
7                 .toList()
8
9 // class::instanceMethod
10 assert ['A', 'B', 'C'] ==
11         ['a', 'b', 'c'].stream()
12                 .map(String::toUpperCase)
13                 .toList()
```

Operators

`!in` and `!instanceof`

```
assert 42 !instanceof LocalDate
```

```
assert 4 !in [1, 3, 5, 7]
```

Elvis assignment

```
1 import groovy.transform.ToString
2
3 @ToString
4 class Element {
5     String name
6     int atomicNumber
7 }
8
9 def he = new Element(name: 'Helium')
10 he.with {
11     name = name ?: 'Hydrogen'    // existing Elvis operator
12     atomicNumber ?= 2            // new Elvis assignment shorthand
13 }
14 assert he.toString() == 'Element(Helium, 2)'
```

Identity comparison operators

```
1 import groovy.transform.EqualsAndHashCode
2
3 @EqualsAndHashCode
4 class Creature { String type }
5
6 def cat = new Creature(type: 'cat')
7 def copyCat = cat
8 def lion = new Creature(type: 'cat')
9
10 assert cat.equals(lion) // Java logical equality
11 assert cat == lion     // Groovy shorthand operator
12
13 assert cat.is(copyCat) // Groovy identity
14 assert cat === copyCat // operator shorthand
15 assert cat != lion     // negated operator shorthand
```

Safe indexing

```
1 String[] array = ['a', 'b']
2 assert 'b' == array?[1]           // get using normal array index
3 array?[1] = 'c'                   // set using normal array index
4 assert 'c' == array?[1]
5
6 array = null
7 assert null == array?[1]          // return null for all index values
8 array?[1] = 'c'                   // quietly ignore attempt to set value
9 assert null == array?[1]
10
11 def personInfo = [name: 'Daniel.Sun', location: 'Shanghai']
12 assert 'Daniel.Sun' == personInfo?['name'] // get using normal map index
13 personInfo?['name'] = 'sunlan'           // set using normal map index
14 assert 'sunlan' == personInfo?['name']
15
16 personInfo = null
17 assert null == personInfo?['name']        // return null for all map values
18 personInfo?['name'] = 'sunlan'           // quietly ignore attempt to set value
19 assert null == personInfo?['name']
```

var reserved type

```
1 var two = 2
2 IntFunction<Integer> twice = (final var x) -> x * two
3 assert [1, 2, 3].collect{ twice.apply(it) } == [2, 4, 6]
```

// Java 10

// Java 11

Interface default methods

- Can add default methods to interfaces, as in Java 8+
- Still probably better to use traits

New GDK methods

- `average()` on arrays and iterables
- `takeBetween()` on `String` (including Groovy strings) and `CharSequence`
- `shuffle()` and `shuffled()` on arrays and iterables
- `minus()` on `LocalDate`

AST transforms

- @NullCheck

Split packages

- JPMS requires classes in distinct modules have distinct package names
- Groovy has not been "modularized" according to Java 9+
- Groovy 3 moving toward compliance
- Target for compliance is Groovy 4
- Some packages have been renamed and classes moved
 - See release notes for details

Conclusions

- Big improvement is support for Java 8+ syntax
- Others are small, but interesting
- Elvis assignment
- Identity operator ===
- Not with in and instanceof
- Safe indexing of arrays
- do-while loop
- Groovy is still awesome