

Martin Fowler's closure examples in Groovy

Martin Fowler wrote an article in his [Bliki on Closures](#). He uses Ruby as demonstration language for closures. On this page (nearly) the same example is written in Groovy:

```
def managers(emps) {  
  emps.findAll { e -> e.isManager() }  
}
```

```
def highPaid(emps) {  
  threshold = 150  
  emps.findAll { e -> e.salary > threshold }  
}
```

```
def paidMore(amount) {  
  { e -> e.salary > amount }  
}
```

```
def highPaid = paidMore(150)  
println highPaid(emps[0])
```

```
new File(filename).withReader{ reader -> doSomethingWith(reader) }
```

The whole example with class *Employee* (with an dispensible, but convenient, because boosting readability, *toString()* method), an example list with four employees and some explaining assertions (Dierk would call this *Inline Unittests*):

```
class Employee {
  def name, salary
  boolean manager
  String toString() { return name }
}

def emps = [new Employee(name:'Guillaume', manager:true, salary:200),
  new Employee(name:'Graeme', manager:true, salary:200),
  new Employee(name:'Dierk', manager:false, salary:151),
  new Employee(name:'Bernd', manager:false, salary:50)]

def managers(emps) {
  emps.findAll { e -> e.isManager() }
}

assert emps[0..1] == managers(emps) // [Guillaume, Graeme]

def highPaid(emps) {
  threshold = 150
  emps.findAll { e -> e.salary > threshold }
}

assert emps[0..2] == highPaid(emps) // [Guillaume, Graeme, Dierk]

def paidMore(amount) {
  { e -> e.salary > amount }
}
def highPaid = paidMore(150)

assert highPaid(emps[0]) // true
assert emps[0..2] == emps.findAll(highPaid)

def filename = 'test.txt'
new File(filename).withReader{ reader -> doSomethingWith(reader) }

def readersText
def doSomethingWith(reader) { readersText = reader.text }

assert new File(filename).text == readersText
```