For Expressions

Lesson Objectives

- After completing this lesson, you should be able to:
  - Understand the relationship between for expressions and higher order functions
  - Describe the usage of for expressions

Composition Is Hard

- Trying to put together multiple higher order functions into a single expression is difficult
- For expressions are syntactic sugar that simplifies the work of coding a multi-stage transformation

Composing HOFs

```
scala> val myNums = 1 to 3
myNums: Range.Inclusive = Range(1, 2, 3)

scala> myNums.map(i => (1 to i).map(j => i * j))
res0: IndexedSeq[IndexedSeq[Int]] =
  Vector(Vector(1), Vector(2, 4), Vector(3, 6, 9))

scala> myNums.flatMap(i => (1 to i).map(j => i * j))
res1: IndexedSeq[Int] = Vector(1, 2, 4, 3, 6, 9)
```

For Expressions

```
scala> val myNums = 1 to 3
myNums: Range.Inclusive = Range(1, 2, 3)

scala> for {
     |   i <- myNums
     |   j <- 1 to i
     | } yield i * j
res0: IndexedSeq[Int] = Vector(1, 2, 4, 3, 6, 9)
```

Typesafe

Syntax

- Must start with the `for` keyword
- Must have generators, using the `<-` arrow
- The `yield` keyword dictates whether or not a new value is returned

Syntax

- Syntactic sugar over **`map`**, **`flatMap`**, **`withFilter`** and **`foreach`**
- Higher Order Functions have rules
  - If I **`map`** over a List, I will get a List
  - The first generator of a for expression follows the same rule
- Can have guard conditions to apply filters

Filtering

```
scala> val myNums = 1 to 3
myNums: scala.collection.immutable.Range.Inclusive =
  Range(1, 2, 3)

scala> for {
     | i <- myNums if i % 2 == 1
     | j <- 1 to i
     | } yield i * j
res0: scala.collection.immutable.IndexedSeq[Int] =
  Vector(1, 3, 6, 9)
```

Typesafe

Definitions

```
for {
  time <- times
  hours = time.hours if hours > 12
} yield (hours - 12) + "pm"
// Result: Vector[String] = Vector(1pm, 2pm)
```

Typesafe

Effectful Usages

```
for (n <- 1 to 3) println(n)
(1 to 3).foreach(n => println(n))
```

Typesafe

Lesson Summary

- Having completing this lesson, you should be able to:
  - Understand the relationship between for expressions and higher order functions
  - Describe the usage of for expressions

Typesafe