

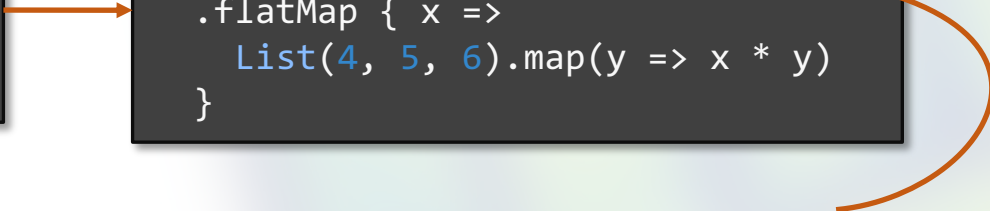
**map, flatMap,
filter, for**



For Comprehensions

```
for {  
  x <- List(1, 2, 3) if x % 2 != 0  
  y <- List(4, 5, 6)  
} yield x * y
```

```
List(1, 2, 3)  
  .withFilter(_ % 2 != 0)  
  .flatMap { x =>  
    List(4, 5, 6).map(y => x * y)  
  }
```



The *if* guards are converted into *withFilter* calls

The generators are nested using *flatMap*

The expression following *yield* is converted into a *map*

specialized definition of filter used only
within for comprehensions

For comprehensions improve code readability

Scala rocks

