For each iteration of your for loop, yield generates a value which will be remembered. It's like the for loop has a buffer you can't see, and for each iteration of your for loop, another item is added to that buffer. **When your for loop finishes running, it will return this collection of all the yielded values**. The type of the collection that is returned is the same type that you were iterating over, so a Map yields a Map, a List yields a List, and so on.

Also, **note that the initial collection is not changed**; the for/yield construct creates a **new** collection according to the algorithm you specify.

scala> for (e <- a **if e > 2**) yield e

res1: Array[Int] = Array(3, 4, 5)

def scalaFiles =

for {

file <- filesHere

if file.isFile

if file.getName.endsWith(".scala")

} yield file

**The concept of guards will be explained later**