

Queries with For

Queries with For

- The for notation is essentially equivalent to the common operations of query languages for databases
- Example: Suppose that we have a database books, represented as a list of books:

case class Book(title: String, authors: List[String])

Suppose we also have a database of books, which is represented as a list of books:

```
val books: List[Book] = List(  
  Book(title = "Structure and Interpretation of Computer Programs",  
    authors = List("Abelson, Harald", "Sussman, Gerald J.")),  
  Book(title = "Introduction to Functional Programming",  
    authors = List("Bird, Richard", "Wadler, Phil")),  
  Book(title = "Effective Java",  
    authors = List("Bloch, Joshua")),  
  Book(title = "Java Puzzlers",  
    authors = List("Bloch, Joshua", "Gafter, Neal")),  
  Book(title = "Programming in Scala",  
    authors = List("Odersky, Martin", "Spoon, Lex", "Venners, Bill")))
```

Queries

- To find the titles of books whose author's name is "Bird":
for (b <- books; a <- b.authors if a startsWith "Bird, ") yield b.title
- To find all the books which have the word "Program" in the title:
for (b <- books if b.title indexOf "Program" >= 0) yield b.title
- To find the names of all authors who have written at least two books present in the database:

```
for {  
  b1 <- books;  
  b2 <- books;  
  if b1 != b2;  
  a1 <- b1.authors;  
  a2 <- b2.authors;  
  if a1 == a2  
} yield a1
```

If we have two different books written by the same author("Bloch, Joshua"), we will get his name twice in the result. The reason why is that we have two generators which go over books and each pair of books will show up twice(once with the argument swapped).

How can we avoid this? – One easy way to avoid it would be instead of just demanding that the two books are different, to demand that the title of the first book must be lexicographically smaller than the second one.

But if we have three different books with the same author, we would get the name three times in the result, because we have three possible pairs of two books out of these three books.

Solution: *We must remove duplicate authors who are in the results at least twice.
This can be achieved by using the distinct method on sequences:*

```
{ for {  
  b1 <- books  
  b2 <- books  
  if b1.title < b2.title  
    a1 <- b1.authors  
    a2 <- b2.authors  
    if a1 == a2  
  } yield a1  
}.distinct
```

Better alternative: *Compute with sets instead of sequences:*

```
val bookSet = books.toSet  
for {  
  b1 <- bookSet  
  b2 <- bookSet  
  if b1.title < b2.title  
    a1 <- b1.authors  
    a2 <- b2.authors  
    if a1 == a2  
  } yield a1
```