Handout

Introduction to Ruby 2.0
Lesson 4 - Step 4
Darren Jones

Ranges

Now we're going to look at **ranges** in IRB.

A range can be used to represent a sequence of values in order. We've already seen some examples of these when creating random numbers between two values, and for finding subsets of arrays. You can create a range by separating the start and the end of the range by two dots:

```
(1..10)
```

This will create a range of all the integers from 1 to 10, including 10. If you use three dots like this

```
(1...10)
```

you will create a range of all the integers from one to nine. So, two dots is used for an inclusive range, and three dots is used for an exclusive range, where the last value isn't included.

We can also create a range of letters. For example, we could create an alphabet of lower case letters, and assign it to the variable alphabet:

```
alphabet = (a..z)
```

Unfortunately, you can't access the individual values of a range, but you can with arrays and hashes. So if we wanted to know which letter was in position two of the range, we could try this

```
alphabet[2]
```

However, you'll get an error message because that isn't a method for accessing individual values in a range.

to_a

Fortunately there is a handy method called `to_a`, which stands for "to array", and this will convert our range into an array.

```
alphabet.to_a[2]
```