Lists

```
sum(0:9)
append(LETTERS[1:13],letters[14:26])
c(1,6,4,9)*2
something <- c(1,4,letters[2])  # indices start at one, you get (1,4
length(something)
```

```
sum(0:9)
[1] 45
> append(LETTERS[1:13],letters[14:26])
 [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "n" "o"
[16] "p" "q" "r" "s" "t" "u" "v" "w" "x" "y" "z"
> c(1,6,4,9)*2
[1]  2 12  8 18
> something <- c(1,4,letters[2]) #indices start at one, you get (1,4
> length(something)
[1] 3
```

Generating lists

You can use a colon to generate a list of numbers:

```
-5:5
```

```
[1] -5 -4 -3 -2 -1  0  1  2  3  4  5
```

The c function

The `c` function **combine**s the parameters into a list and converts them to the same type.

```
c("test",3)
```

```
[1] "test" "3"
```

```
typeof("3")
```

```
[1] "character"
```

Here `3` is converted to a string.

The seq function

`seq` generates more complex regular **sequence**s:

```
> seq(from=1,to=4,by=.6)
```

```
[1] 1.0 1.6 2.2 2.8 3.4 4.0
```

Accessing list members

List members can be accessed using brackets as in most languages: `(3:5)[2]`.

This returns `4` because **indices start with 1**.

You can also extract multiple list members from a list: `letters[2:4]` returns `[1] "b" "c" "d"`.

Appending to lists

You can use the `append` function for this. Its return value has to be reassigned to the variable.

By default the new value is appended at the end of the list. You can use the `after` argument to change that:

R Style Guide
R Language Definition (pdf)
R Function Info
RStudio IDE

```
a <- 1:4
append(a,2.4,after=2)
```

```
[1] 1.0 2.0 2.4 3.0 4.0
```

Operating on lists

R allows you to easily operate on all list values at once.

```
c(1,2,3) + 3
```

This and the `apply` function allow you to avoid most for loops.

```
[1] 4 5 6
```

Predefined lists

Lists for letters and month names are predefined:

```
letters
```

```
[1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o"
[16] "p" "q" "r" "s" "t" "u" "v" "w" "x" "y" "z"
```

```
LETTERS
```

```
[1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O"
[16] "P" "Q" "R" "S" "T" "U" "V" "W" "X" "Y" "Z"
```

```
month.abb
```

```
[1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep" "Oct"
[11] "Nov" "Dec"
```

```
month.name
```

```
[1] "January"   "February"  "March"     "April"     "May"
 [6] "June"      "July"      "August"    "September" "October"
[11] "November"  "December"
```

What are vectors?

Vectors are lists in which all elements have the same type. For example, the `c` function creates a vector.