

Errors most often occur when code is used in a way that it is not intended to be used. For example adding two strings together produces the following error:

```
1 "hello" + "world"
2 Error in "hello" + "world": non-numeric argument to binary operator
```

The + operator is essentially a function that takes two numbers as arguments and finds their sum. Since neither "hello" nor "world" are numbers, the R interpreter produces an error. Errors will stop the execution of your program, and they will (hopefully) print an error message to the R console.

In R there are two other constructs in R which are both related to errors: warnings and messages. Warnings are meant to indicate that something seems to have gone wrong in your program which should be inspected. Here's a simple example of a warning being generated:

```
1 as.numeric(c("5", "6", "seven"))
2 Warning: NAs introduced by coercion
3 [1] 5 6 NA
4
```

The as.numeric() function attempts to convert each string in c("5", "6", "seven") into a number, however it is impossible to convert "seven", so a warning is generated. Execution of the code is not halted, and an NA is produced in place of "seven" instead of a number.

Messages simply print text to the R console, though they are generated by an underlying mechanism that is similar to how errors and warning are generated. Here's a small function that will generate a message:

```
1 f <- function(){
2   message("This is a message.")
3 }
4
5 f()
6 This is a message.
7
```

Mark as completed