

Environments are data structures in R that have special properties with regard to their role in how R code is executed and how memory in R is organized. You may not realize it but you're probably already familiar with one environment called the global environment. Environments formalize relationships between variable names and values. When you enter `x <- 55` into the R console what you're saying is: assign the value of 55 to a variable called x, and store this assignment in the global environment. The global environment is therefore where most R users do most of their programming and analysis.

You can create a new environment using `new.env()`. You can assign variables in that environment in a similar way to assigning a named element of a list, or you can use `assign()`. You can retrieve the value of a variable just like you would retrieve the named element of a list, or you can use `get()`. Notice that `assign()` and `get()` are opposites:

```
1 my_new_env <- new.env()
2 my_new_env$x <- 4
3 my_new_env$x
4 [1] 4
5
6 assign("y", 9, envir = my_new_env)
7 get("y", envir = my_new_env)
8 [1] 9
9 my_new_env$y
10 [1] 9
```

You can get all of the variable names that have been assigned in an environment using `ls()`, you can remove an association between a variable name and a value using `rm()`, and you can check if a variable name has been assigned in an environment using `exists()`:

```
1 ls(my_new_env)
2 [1] "x" "y"
3 rm(y, envir = my_new_env)
4 exists("y", envir = my_new_env)
5 [1] TRUE
6 exists("x", envir = my_new_env)
7 [1] TRUE
8 my_new_env$x
9 [1] 4
10 my_new_env$y
11 NULL
```

Environments are organized in parent/child relationships such that every environment keeps track of its parent, but parents are unaware of which environments are their children. Usually the relationships between environments is not something you should try to directly control. You can see the parents of the global environment using the `search()` function:

```
1 search()
2 [1] ".GlobalEnv" "package:magrittr"
3 [3] "package:tidyr" "package:microbenchmark"
4 [5] "package:purrr" "package:dplyr"
5 [7] "package:readr" "package:stats"
6 [9] "package:graphics" "package:grDevices"
7 [11] "package:utils" "package:datasets"
8 [13] "Autoloads" "package:base"
```

As you can see `package:magrittr` is the parent of `.GlobalEnv`, and `package:tidyr` is parent of `package:magrittr`, and so on. In general the parent of `.GlobalEnv` is always the last package that was loaded using `library()`. Notice that after I load the `ggplot2` package, that package becomes the parent of `.GlobalEnv`:

```
1 library(ggplot2)
2 search()
3 [1] ".GlobalEnv" "package:ggplot2"
4 [3] "package:magrittr" "package:tidyr"
```