**Environments in R Programming**

The environment is a virtual space that is triggered when an interpreter of a programming language is launched. Simply, the environment is a collection of all the objects, variables, and functions. Or, Environment can be assumed as a top-level object that contains the set of names/variables associated with some values. In this article, let us discuss creating a new environment in [R programming](https://www.geeksforgeeks.org/r-language/r-programming-language-introduction/), list all environments, removing a variable from the environment, searching for a variable or function among environments and function environments with the help of examples.

**Why the Environment Differ from the List?**

* Every object in an environment has a name.
* The environment has a parent environment.
* Environments follow reference semantics.

**Create a New Environment**

An environment in R programming can be created using **new.env()** function. Further, the variables can be accessed using **$ or [[ ]]** operator. But, each variable is stored in different memory locations. There are four special environments: **globalenv(), baseenv(), emptyenv() and environment()**

**Syntax:** new.env(hash = TRUE)

**Parameters:**   
**hash:** indicates logical value. If TRUE, environments uses a hash table

To know about more optional parameters, use below command in console: **help("new.env")**

**Example:**

# R program to illustrate

# Environments in R

# Create new environment

newEnv <- new.env()

# Assigning variables

newEnv$x <- 1

newEnv$y <- "GFG"

newEnv$z <- 1:10

# Print

print(newEnv$z)

**Output:**

[1] 1 2 3 4 5 6 7 8 9 10

**List all Environments**

Every environment has a parent environment but there is an empty environment that does not have any parent environment. All the environments can be listed using [**ls()**](https://www.geeksforgeeks.org/r-language/list-all-the-objects-present-in-the-current-working-directory-in-r-programming-ls-function/) function and [**search()**](https://www.geeksforgeeks.org/r-language/get-a-list-of-all-the-attached-packages-in-r-programming-search-function/) function. **ls()** function also list out all the bindings of the variables in a particular environment.

**Syntax:**

ls()   
search()

***Parameters:***  
*These functions need no argument*

**Example:**

# R program to illustrate

# Environments in R

# Prints all the bindings and environments

# attached to Global Environment

ls()

# Prints bindings of newEnv

ls(newEnv)

# Lists all the environments of the parent environment

search()

**Output:**

[1] "al" "e" "e1" "f" "newEnv" "pts" "x" "y"

[9] "z"

[1] "x" "y" "z"

[1] ".GlobalEnv" "package:stats" "package:graphics"

[4] "package:grDevices" "package:utils" "package:datasets"

[7] "package:methods" "Autoloads" "package:base"

**Removing a Variable From an Environment**

A variable in an environment is deleted using [**rm()**](https://www.geeksforgeeks.org/r-language/remove-objects-from-memory-in-r-programming-rm-function/) function. It is different from deleting entries from lists as entries in lists are set as NULL to be deleted. But, using **rm()** function, bindings are removed from the environment.

**Syntax:** rm(...)

**Parameters:**   
**...:** indicates list of objects

**Example:**

# R program to illustrate

# Environments in R

# Remove newEnv

rm(newEnv)

# List

ls()

**Output:**

[1] "al" "e" "e1" "f" "pts" "x" "y" "z"

**Search a Variable or Function Among Environments**

A variable or a function can be searched in R programming by using **where()** function among all the environments and packages present. **where()** function is present in **pryr** package. This function takes only two arguments, the name of the object to search for and the environment from where to start the search.

**Syntax:** where(name)

**Parameters:**   
**name:** indicates object to look for

**Example:**

# R program to illustrate

# Environments in R

# Install pryr package

install.packages("pryr")

# Load the package

library(pryr)

# Search

where("x")

where("mode")

**Output:**

<environment: R\_GlobalEnv>

<environment: base>