

4. Funciones

Índice

- [1. Definición y justificación](#)
- [2. Parámetros y retorno](#)
- [3. Regla de ámbito](#)
- [4. Funciones integradas](#)

1. Definición y justificación

Una función permite encapsular un fragmento de código de cara a su reutilización posterior.

Una función se compone principalmente de 4 partes como máximo:

- Nombre que la identifica
- Parámetros que recibe (opcional)
- Proceso que realiza
- Valor que devuelve (opcional)

En R las funciones se declaran de esta forma:

```
learn <- function() {  
  print("Estoy aprendiendo R")  
}
```

Y se la llama así:

```
learn()
```

2. Parámetros y retorno

Las funciones pueden recibir parámetros (también llamados argumentos) con distintos valores de entrada en cada llamada:

```
learnSomething <- function(something) {  
  paste("Estoy aprendiendo", something)  
}  
  
learnSomething("R a saco")
```

Una función debe recibir al ser llamada el mismo número de argumentos que los parámetros con los que se la declaró.

Si fuera necesario, pueden declararse parámetros opcionales (también llamados por defecto), con un valor predefinido si no se indica otro al llamarla:

```
learnSomethingDefault <- function(something = "algo") {  
  paste("Estoy aprendiendo", something)  
}  
  
learnSomethingDefault()  
learnSomethingDefault("R a cascoporro")
```

Las funciones también pueden devolver valores resultantes del proceso que realicen:

```
learnSomethingAndLevelUp <- function(something, level) {  
  paste("Estoy aprendiendo", something)  
  
  return (level + 1)  
}  
  
level <- 0  
level <- learnSomethingAndLevelUp("R like a pro", level)
```

Por último, R permite crear funciones anidadas (dentro de otras funciones) y también pasar funciones como argumentos.

3. Regla de ámbito

Una variable declarada fuera de una función puede ser leída dentro de la misma:

```
myText <- "genial"  
  
beFunny <- function() {  
  paste("Una ranita iba caminando, R es", myText)  
}  
  
beFunny()
```

Una variable declarada dentro de una función no podrá ser leída fuera de la misma, y solo será accesible dentro del ámbito de la función:

```
myText <- "genial"  
  
beMoreFunny <- function() {  
  myText <- "la leche,"  
  myOtherText <- "ni lo dudes"
```

```
paste("Aquella ranita seguía caminando y R es", myText, myOtherText)
}  
  
beMoreFunny()  
  
myText  
myOtherText
```

Para poder cambiar el valor de una variable declarada fuera de una función, debe referenciarse con el operador de asignación global:

```
myText <- "genial"  
  
beSuperFunny <- function() {  
  myText <- "lo mejor que me ha pasado"  
  
  paste("Otra ranita voló y R es", myText)  
}  
  
beSuperFunny()  
  
paste("Repito, ranas aparte: R es", myText)
```

4. Funciones integradas

El lenguaje ofrece un gran catálogo de funciones integradas para diversas operaciones de todo tipo, además de las que se obtienen por la importación de otros paquetes y librerías.

Algunos ejemplos son: `getwd()`, `log()`, `tolower(x)`, `range(x)`, `sum(x)`, etc.

Referencias

[Funciones](#)

[Built-in functions \(I\)](#)

[Built-in functions \(II\)](#)