TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Nuevo León

Algoritmos y lenguaje de programación.

Unidad 4

Maestro: Juan Pablo Rosas Baldazo

Alumno: Graciela Sarahi Hernández Bautista

No. De control: 18480025

Funciones

A diferencia de otros lenguajes de programación procedurales, como C, Java, y PHP, en R las funciones constituyen una clase. Por ejemplo, los objetos de esa clase pueden ser asignados a variables; podría darse el caso, incluso, de armar una lista cuyos elementos fueran funciones.
La forma de crear funciones como una herramienta para agrupar varias operaciones.

La sintaxis para la creación de una función es como sigue:
variable <- **function**(arg_1, arg_2, ..., arg_n) expression

Se trata de una asignación de un valor: la función, a una variable. A partir de esa definición, la variable se puede utilizar como el nombre de la función. En R, toda expresión tiene un valor, así que el valor de la expresión será lo que la función regresará cuando se aplique. En seguida se muestra un ejemplo.

```
hipotenusa <- function(x, y) {
sqrt(x^2 + y^2)
}
class(hipotenusa)
## [1] "function"
```

Para utilizar esta función lo hacemos con:

```
hipotenusa(3, 4)
## [1] 5
```

Los argumentos de la función tienen nombres, y esos se pueden usar en el llamado a la función, cambiando incluso el orden en que aparecen en la definición de la función.

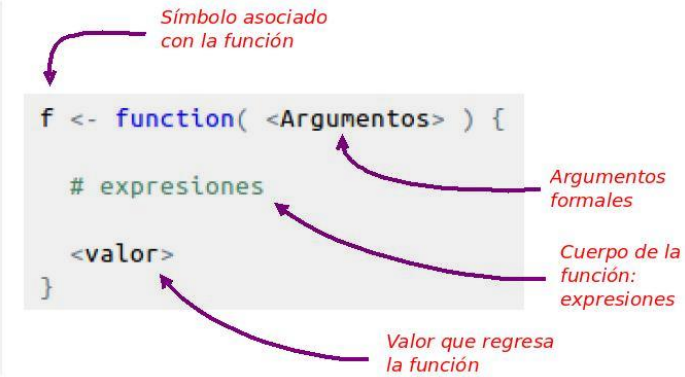
```
hipotenusa(y = 4, x = 3)
## [1] 5
```

Otra característica es que las funciones, en su definición, pueden tener valores asignados por defecto o en ausencia cuando es llamada la función:

```
hipotenusa <- function(x=3, y=4) { # valores por ausencia
return( sqrt( x^2 + y^2 ) )
}
# Llamamos a la función con argumentos "ausentes"
hipotenusa()
## [1] 5
```

Estructura formal de las funciones

Existen dos momentos importantes en la vida de una función: la definición de la función, que básicamente ocurre una vez, y la ejecución de la función, que puede ocurrir un sinnúmero de ocasiones y en contextos muy diversos. Las nociones de argumentos formales, variables y valor de regreso en el interior de una función, deben ser examinadas desde la perspectiva de esos dos momentos.



Argumentos y valor de resultado de una función

Los argumentos de una función se enfocan desde las perspectivas de los dos momentos de la vida de una función: su definición y su ejecución. En el momento de la definición de la función se ven principalmente como lo que se denomina argumentos formales, mientras que en el momento de la ejecución, se considera cómo esos argumentos formales se asocian con un valor particular a partir de sus respectivos argumentos verdaderos o efectivos, que son los que se especifican o se pasan, por así decirlo, en el instante de invocar la función.

Los argumentos formales son uno de los principales medios por el cual la función se comunica con el ambiente exterior a ella; esto es, el sitio dónde la función es invocada para ejecutarse, y consiste de una lista de símbolos, que pudieran tener o no asociado algún valor a utilizar en caso de no asignársele alguno en el momento de su invocación. Al momento de ejecutarse la función, los argumentos formales se asociarán con algún valor, típicamente, a partir de los argumentos verdaderos que se proveen en la invocación de la función, o de los valores por omisión o de default, que se hayan indicado en la definición de la función.

```
# Definición -- Versión 1
MiFunc.v1 <- function (x, yyy, z=5, t) {
w <- x + yyy + z
w
}
```

```r
# Definición -- Versión 2
MiFunc.v2 <- function (x, yyy, z=5, t) {
w <- x + yyy + z
return (w)
3.1416 # Este código nunca se ejecuta
}
# Definición -- Versión 3
MiFunc.v3 <- function (x, yyy, z=5, t) {
x + yyy + z
}
# Ejecuciones:
MiFunc.v1(1,2,3) # Ejecución 1
## [1] 6
MiFunc.v2(1,2) # Ejecución 2
## [1] 8
MiFunc.v3(1) # Ejecución 3
## Error: el argumento "yyy" está ausente, sin valor por omisión
MiFunc.v3(z=3, yyy=2, x=1) # Ejecución 4
## [1] 6
MiFunc.v2(1, y=2) # Ejecución 5
## [1] 8
MiFunc.v1(1, z=3) # Ejecución 6
## Error: el argumento "yyy" está ausente, sin valor por omisión
```

Bibliografía.

Santana, J. & Mateos, E. (2014). *El arte de programar en R: un lenguaje para la estadística.* México: Instituto Mexicano de Tecnología del Agua.