

R Examples

[Basics](#)
[Functions](#)
[Countdown](#)
[User input](#)
[Random number game](#)
[Lists](#)
[Reading data](#)
[Filtering data](#)
[More Examples](#)
[How to run the code](#)
[Finding data sources](#)

Functions

```

Square <- function(x) {
  return(x^2)
}

print(Square(4))
print(Square(x=4)) # same thing
[1] 16
[1] 16

```

Defining functions

Functions can be defined like this:

```
function(parameter1, parameter2, parameter3){code}.
```

For use they are assigned to a variable (using normal `<-` assignment operator).

Return values

The `return` function is used for return values: `return(value)`. If no value is given `NULL` is returned.

Exponentiation

Raise a to the power of b: `a^b`. It's also possible to use `**` instead of `^`.

Named arguments

Named arguments work like this: `DoSomething(color="red", number=55)`.





You can also give the first arguments in order and then use named arguments:
`DoSth(value1, value2, arg4=value4, arg3=value3)`.

Automatic printing at the console

When using the console directly R automatically prints the return value of a statement if you don't assign it to a variable. This doesn't work in loops or functions.

You can use `invisible(CalculateSth())` if you don't want the return value to be printed.

Become an R Master

 <p>R for Data Science: Import, Tidy, Transf...</p> <p>\$36.06 ✓prime ★★★★★ (54)</p>	 <p>R Cookbook: Proven Recipes for Data ...</p> <p>\$31.99 ✓prime ★★★★★ (92)</p>	 <p>R Graphics Cookbook: Practical Recipes ...</p> <p>\$33.16 ✓prime ★★★★★ (82)</p>	 <p>Practical Data Science with R</p> <p>\$33.75 ✓prime ★★★★★ (33)</p>
---	--	--	---

[R Style Guide](#)
[R Language Definition \(pdf\)](#)
[R Function Info](#)
[RStudio IDE](#)

Made by Matt Zeunert

Ads by Amazon

