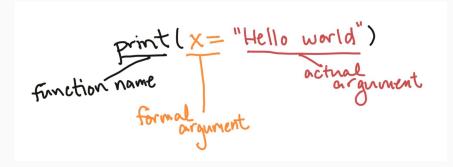
Preliminaries in R

In general, function calls in R take the following structure:

A function call forms a complete R expression, and the output will be the result of running print or show on the object that is output by the function call.

Here is an example of this structure:

```
print(x = "Hello world")
## [1] "Hello world"
```



In this example, we're **calling** a function with the **name** print. It has one **argument**, with a **formal argument** of x, which in this call we've provided the **named argument** "Hello world".

The **arguments** are how you customize the call to an R function.

For example, you can use change the named argument value to print different messages with the print function:

```
print(x = "Hello world")

## [1] "Hello world"

print(x = "Hi Fort Collins")

## [1] "Hi Fort Collins"
```

Some functions do not require any arguments. For example, the getRversion function will print out the version of R you are using.

getRversion()

```
## [1] '4.4.1'
```

Some functions will accept multiple arguments. For example, the print function allows you to specify whether the output should include quotation marks, using the quote formal argument:

```
print(x = "Hello world", quote = TRUE)

## [1] "Hello world"
print(x = "Hello world", quote = FALSE)

## [1] Hello world
```

Arguments can be required or optional.

For a required argument, if you don't provide a value for the argument when you call the function, R will respond with an error. For example, x is a **required argument** for the print function, so if you try to call the function without it, you'll get an error:

```
print()
```

```
Error in print.default() : argument "x" is
  missing, with no default
```

For an **optional argument** on the other hand, R knows a **default value** for that argument, so if you don't give it a value for that argument, it will just use the default value for that argument.

For example, for the print function, the quote argument has the default value TRUE. So if you don't specify a value for that argument, R will assume it should use quote = TRUE. That's why the following two calls give the same result:

```
print(x = "Hello world", quote = TRUE)

## [1] "Hello world"

print(x = "Hello world")

## [1] "Hello world"
```

Often, you'll want to find out more about a function, including:

- Examples of how to use the function
- Which arguments you can include for the function
- Which arguments are required versus optional
- What the default values are for optional arguments.

You can find out all this information in the function's **helpfile**, which you can access using the function?.

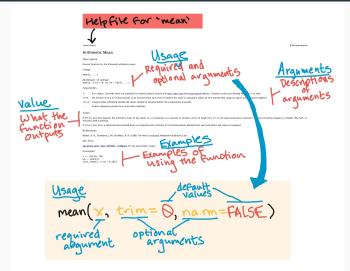
For example, the mean function will let you calculate the mean (average) of a group of numbers. To find out more about this function, at the console type:

?mean

This will open a helpfile in the "Help" pane in RStudio.



The helpfile includes sections giving the function's **usage**, **arguments**, **value**, and **examples**.



You can figure out which arguments are **required** and which are **optional** in the Usage section of the helpfile.

Operators

There's one class of functions that looks a bit different from others. These are the infix **operator** functions.

Instead using parentheses after the function name, they usually go *between* two arguments.

One common example is the + operator:

Operators

There are operators for several mathematical functions: +, -, *, /.

There are also other operators, including **logical operators** and **assignment operators**, which we'll cover later.