

Preliminaries in R

Objects and assignment

Objects, object names, and assignment expressions

Function calls will usually produce something called an **object**.

If you just call a function, as we've been doing, then R will respond by printing out that object.

However, we'll often want to use that object some more. For example, we might want to use it as an argument later in our “conversation” with R, when we call another function later.

If you want to re-use the results of a function call later, you can **assign** that **object** to an **object name**.

This kind of expression is called an **assignment expression**.

Assignment expressions

The **gets arrow**, `<-`, is R's assignment operator. It takes whatever you've created on the right hand side of the `<-` and saves it as an object with the name you put on the left hand side of the `<-` :

Note: Generic code-- this will not work

```
[object name] <- [object]
```

Assignment expressions

For example, if I just type "Hello world", R will print it back to me, but won't save it anywhere for me to use later:

```
"Hello world"
```

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

Assignment expressions

However, if I assign it to an object, I can “refer” to that object in a later expression.

For example, the code below assigns the **object** "Hello world" the **object name** message. Later, I can just refer to this object using the name message, for example in a function call to the print function:

```
message <- "Hello world"  
print(x = message)
```

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

History of <-



Assignment expressions

When you enter an **assignment expression** like this at the R console, if everything goes right, then R will “respond” by giving you a new prompt, without any kind of message.

However, there are three ways you can check to make sure that the object was assigned to the object name:

1. Enter the object's name at the prompt and press return. The default if you do this is for R to “respond” by calling the `print` function with that object as the `x` argument.
2. Call the `ls` function (which doesn't require any arguments). This will list all the object names that have been assigned in the current R session.
3. Look in the “Environment” pane in RStudio. This also lists all the object names that have been assigned in the current R session.

Assignment expressions

Here's an example of the first two strategies:

1. Enter the object's name at the prompt and press return:

```
message
```

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

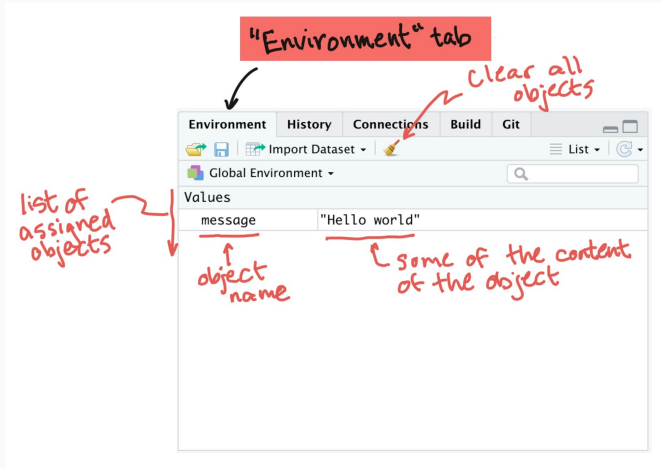
2. Call the `ls` function:

```
ls()
```

```
## [1] "message"
```

"Environment" pane

Here's an example of the third method:



Object names

There are some absolute **rules** for the names you can use for an object name:

- Use only letters, numbers, and underscores
- Don't start with anything but a letter

Assigning objects to object names

If you try to assign an object to a name that doesn't follow the “hard” rules, you'll get an error.

For example, all of these expressions will give you an error:

```
1message <- "Hello world"  
_message <- "Hello world"  
message! <- "Hello world"
```

Object names

There are also some **guidelines** for picking *good* object names:

From Hadley Wickham's R style guide

- Use lower case for variable names (message, not Message)
- Use an underscore as a separator (message_one, not messageOne)
- Avoid using names that are already defined in R (e.g., don't name an object mean, because a mean function exists)