



How it works | R

campus.datacamp.com/courses/free-introduction-to-r/chapter-1-intro-to-basics-1?ex=1

Learn / Courses / Introduction to R

Exercise

How it works

In the editor on the right you should type R code to solve the exercises. When you hit the 'Submit Answer' button, every line of code is interpreted and executed by R and you get a message whether or not your code was correct. The output of your R code is shown in the console in the lower right corner.

R makes use of the `#` sign to add comments, so that you and others can understand what the R code is about. Just like Twitter! Comments are not run as R code, so they will not influence your result. For example, `Calculate 3 + 4` in the editor on the right is a comment.

You can also execute R commands straight in the console. This is a good way to experiment with R code, as your submission is not checked for correctness.

Instructions 100 XP

- In the editor on the right there is already some sample code. Can you see which lines are actual R code and which are comments?
- Add a line of code that calculates the sum of 6 and 12, and hit the 'Submit Answer' button.

Take Hint (-30 XP)

```
scriptR
1 # Calculate 3 + 4
2 3 + 4
3
4 # Calculate 6 + 12
5
```

R Console

DataCamp Learn

app.datacamp.com/learn

datacamp Home Learn Workspace Certification Jobs For Business Search Upgrade EN

Progress

My Library

LEARN

- Tracks
- Courses
- Practice
- Assessments
- Tutorials NEW

APPLY

- Projects
- Case Studies
- Code Alongs
- Competitions

POPULAR TOPICS NEW

- Artificial Intelligence
- Data Engineering

Getting Started (1/4)

Hey, Mukul!

Portfolio 30% complete

Review Daily XP 0 XP

LEARN

Introduction to R

0% 4 hours to go

Keep Making Progress

Already know this?

Take an assessment to verify your skill level and skip this course.

Introduction to R

PRACTICE

Introduction to Python

APPLY

Visualizing COVID-19

Build Data and AI skills 50% off

Start today. Activate 50% off unlimited learning.

Buy Now

How it works | R

campus.datacamp.com/courses/free-introduction-to-r/chapter-1-intro-to-basics-1?ex=1

Gmail MyClass UMS LPU Live LinkedIn W3 Schools LPU Colab HackerRank LeetCode Kaggle Your Home... GitHub Grammarly JoCinema - Watch... Online Java Compiler

Learn / Courses / Introduction to R

← Course Outline →

Daily XP 0

Exercise

How it works

In the editor on the right you should type R code to solve the exercises. When you hit the 'Submit Answer' button, every line of code is interpreted and executed by R and you get a message whether or not your code was correct. The output of your R code is shown in the console in the lower right corner.

R makes use of the `#` sign to add comments, so that you and others can understand what the R code is about. Just like Twitter! Comments are not run as R code, so they will not influence your result. For example, `Calculate 3 + 4` in the editor on the right is a comment.

You can also execute R commands straight in the console. This is a good way to experiment with R code, as your submission is not checked for correctness.

Instructions 100 XP

- In the editor on the right there is already some sample code. Can you see which lines are actual R code and which are comments?
- Add a line of code that calculates the sum of 6 and 12, and hit the 'Submit Answer' button.

Take Hint (-30 XP)

scriptLR

Light Mode

```
1 # Calculate 3 + 4
2 print(3 + 4)
3
4 # Calculate 6 + 12
5 print(6+12)
```

Run Code Submit Answer

R Console

>

How it works | R

campus.datacamp.com/courses/free-introduction-to-r/chapter-1-intro-to-basics-1?ex=1

Gmail MyClass UMS LPU Live LinkedIn W3 Schools LPU Colab HackerRank LeetCode Kaggle Your Home... GitHub Grammarly JoCinema - Watch... Online Java Compiler

Learn / Courses / Introduction to R

← Course Outline →

Daily XP 100

Exercise

How it works

In the editor on the right you should type R code to solve the exercises. When you hit the 'Submit Answer' button, every line of code is interpreted and executed by R and you get a message whether or not your code was correct. The output of your R code is shown in the console in the lower right corner.

R makes use of the `#` sign to add comments, so that you and others can understand what the R code is about. Just like Twitter! Comments are not run as R code, so they will not influence your result. For example, `Calculate 3 + 4` in the editor on the right is a comment.

You can also execute R commands straight in the console. This is a good way to experiment with R code, as your submission is not checked for correctness.

Instructions 100 XP

Awesome! See how the console shows the result of the R code you submitted? Now that you're familiar with the interface, let's get down to R business!

PRESS ENTER TO Continue

Take Hint (-30 XP)

scriptLR

Light Mode

```
1 # Calculate 3 + 4
2 print(3 + 4)
3
4 # Calculate 6 + 12
5 print(6+12)
```

Run Code Submit Answer

R Console

```
# Calculate 6 + 12
print(6+12)
[1] 18
```

Arithmetic with R | R

campus.datacamp.com/courses/free-introduction-to-r/chapter-1-intro-to-basics-1?ex=2

Learn / Courses / Introduction to R

Course Outline

Daily XP 100

Exercise

Arithmetic with R

In its most basic form, R can be used as a simple calculator. Consider the following arithmetic operators:

- Addition: `+`
- Subtraction: `-`
- Multiplication: `*`
- Division: `/`
- Exponentiation: `^`
- Modulo: `%%`

The last two might need some explaining:

- The `^` operator raises the number to its left to the power of the number to its right: for example `3^2` is 9.
- The modulo returns the remainder of the division of the number to the left by the number on its right, for example 5 modulo 3 or `5 %% 3` is 2.

With this knowledge, follow the instructions to complete the exercise.

Instructions

100 XP

- Type `2^5` in the editor to calculate 2 to the power 5.
- Type `28 %% 6` to calculate 28 modulo 6.

scriptR

Light Mode

```
3
4 # A subtraction
5 5 - 5
6
7 # A multiplication
8 3 * 5
9
10 # A division
11 (5 + 5) / 2
12
13 # Exponentiation
14 5 ^ 5
15
16 # Modulo
17 5 %% 5
```

Run Code Submit Answer

R Console

>

Arithmetic with R | R

campus.datacamp.com/courses/free-introduction-to-r/chapter-1-intro-to-basics-1?ex=2

Learn / Courses / Introduction to R

Course Outline

Daily XP 200

Exercise

Instructions

100 XP

- Type `2^5` in the editor to calculate 2 to the power 5.
- Type `28 %% 6` to calculate 28 modulo 6.
- Submit the answer and have a look at the R output in the console.
- Note how the `#` symbol is used to add comments on the R code.

Take Hint (-20 XP)

✓

+100 XP

Incorrect Submission

Great! Head over to the next exercise.

Did you find this feedback helpful?

✓ Yes

✗ No

PRESS ENTER TO

Continue

scriptR

Light Mode

```
7 # A multiplication
8 3 * 5
9
10 # A division
11 (5 + 5) / 2
12
13 # Exponentiation
14 2 ^ 5
15
16 # Modulo
17 5 %% 5
18
19 28 %% 6
```

Run Code Submit Answer

R Console

[1] 0

28 %% 6

[1] 4

>

Variable assignment | R

campus.datacamp.com/courses/free-introduction-to-r/chapter-1-intro-to-basics-1?ex=3

Learn / Courses / Introduction to R

Course Outline

Daily XP 300

Exercise

Variable assignment

A basic concept in (statistical) programming is called a variable.

A variable allows you to store a value (e.g. 42) or an object (e.g. a function description) in R. You can then later call the variable's name to easily access the value or the object that is stored within it.

You can assign a value 42 to a variable `x` with the command:

+100 XP

Good job! Have you noticed that R does not print the value of a variable to the console when you did the assignment? `x <- 42` did not generate any output, because R assumes that you will be needing this variable in the future. Otherwise you wouldn't have stored the value in a variable in the first place, right? Proceed to the next exercise!

Take hint (-30 XP)

Continue

scriptLR

1 # Assign the value 42 to x
2 x <- 42
3
4 # Print out the value of the variable x
5 x

R Console

Print out the value of the variable x

x

[1] 42

>

Variable assignment (2) | R

campus.datacamp.com/courses/free-introduction-to-r/chapter-1-intro-to-basics-1?ex=4

Learn / Courses / Introduction to R

Course Outline

Daily XP 300

Exercise

Variable assignment (2)

Suppose you have a fruit basket with five apples. As a data analyst in training, you want to store the number of apples in a variable with the name `my_apples`.

Instructions 100 XP

- Type the following code in the editor: `my_apples <- 5`. This will assign the value 5 to `my_apples`.
- Type: `my_apples` below the second comment. This will print out the value of `my_apples`.
- Submit your answer, and look at the output: you see that the number 5 is printed. So R now links the variable `my_apples` to the value 5.

Take Hint (-30 XP)

scriptLR

1 # Assign the value 5 to the variable my_apples
2
3
4 # Print out the value of the variable my_apples
5

R Console

>

Variable assignment (3) | R

campus.datacamp.com/courses/free-introduction-to-r/chapter-1-intro-to-basics-1?ex=5

GmailMyClassUMSLPU LiveLinkedInW3 SchoolsLPU ColabHackerRankLeetCodeKaggle Your Home...GitHubGrammarlyJioCinema - Watch...Online Java Compiler

Learn / Courses / Introduction to R

← Course Outline →

Daily XP 400

Exercise

Variable assignment (3)

Every tasty fruit basket needs oranges, so you decide to add six oranges. As a data analyst, your reflex is to immediately create the variable `my_oranges` and assign the value 6 to it. Next, you want to calculate how many pieces of fruit you have in total. Since you have given meaningful names to these values, you can now code this in a clear way:

```
my_apples + my_oranges
```

Instructions100 XP

- Assign to `my_oranges` the value 6.
- Add the variables `my_apples` and `my_oranges` and have R simply print the result.
- Assign the result of adding `my_apples` and `my_oranges` to a new variable `my_fruit`.

Take Hint (-30 XP)

scriptLR

Light Mode

```
1 # Assign a value to the variables my_apples and my_oranges
2 my_apples <- 5
3 my_oranges <- 6
4
5 # Add these two variables together
6 my_fruit <- my_apples+my_oranges
7
8 # Create the variable my_fruit
9 my_fruit
```

Run CodeSubmit Answer

R Console

```
my_apples <- my_apples+my_oranges
# Create the variable my_fruit
my_fruit
[1] 11
>
```

Variable assignment (3) | R

campus.datacamp.com/courses/free-introduction-to-r/chapter-1-intro-to-basics-1?ex=5

GmailMyClassUMSLPU LiveLinkedInW3 SchoolsLPU ColabHackerRankLeetCodeKaggle Your Home...GitHubGrammarlyJioCinema - Watch...Online Java Compiler

Learn / Courses / Introduction to R

← Course Outline →

Daily XP 400

Exercise

Variable assignment (3)

Every tasty fruit basket needs oranges, so you decide to add six oranges. As a data analyst, your reflex is to immediately create the variable `my_oranges` and assign the value 6 to it. Next, you want to calculate how many pieces of fruit you have in total. Since you have given meaningful names to these values, you can now code this in a clear way.

```
my_apples + my_oranges
```

+100 XP

Nice one! The great advantage of doing calculations with variables is reusability. If you just change `my_apples` to equal 12 instead of 5 and rerun the script, `my_fruit` will automatically update as well. Continue to the next exercise.

Take Hint (-30 XP)

PRESS ENTER TO

Continue

scriptLR

Light Mode

```
1 # Assign a value to the variables my_apples and my_oranges
2 my_apples <- 5
3 my_oranges <- 6
4
5 # Add these two variables together
6 my_fruit <- my_apples+my_oranges
7
8 # Create the variable my_fruit
9 my_fruit
```

Reset to sample codeRun CodeSubmit Answer

R Console

```
my_apples <- my_apples+my_oranges
# Create the variable my_fruit
my_fruit
[1] 11
>
```

Apples and oranges | R

campus.datacamp.com/courses/free-introduction-to-r/chapter-1-intro-to-basics-1?ex=6

GmailMyClassUMSLPU LiveLinkedInW3 SchoolsLPU ColabHackerRankLeetCodeKaggle Your Home...GitHubGrammarlyJioCinema - Watch...Online Java Compiler

Learn / Courses / Introduction to R

← Course Outline →

Daily XP 500

Exercise

Apples and oranges

Common knowledge tells you not to add apples and oranges. But hey, that is what you just did, no ->? The `my_apples` and `my_oranges` variables both contained a number in the previous exercise. The `+` operator works with numeric variables in R. If you really tried to add "apples" and "oranges", and assigned a text value to the variable `my_oranges` (see the editor), you would be trying to assign the addition of a numeric and a character variable to the variable `my_fruit`. This is not possible.

Instructions100 XP

- Submit the answer and read the error message. Make sure to understand why this did not work.
- Adjust the code so that R knows you have 6 oranges and thus a fruit basket with 11 pieces of fruit.

Take Hint (-30 XP)

scriptLR

Light Mode

```
1 # Assign a value to the variable my_apples
2 my_apples <- 5
3
4 # Fix the assignment of my_oranges
5 my_oranges <- 6
6
7 # Create the variable my_fruit and print it out
8 my_fruit <- my_apples + my_oranges
9 my_fruit
```

Run CodeSubmit Answer

R Console

>

Apples and oranges | R

campus.datacamp.com/courses/free-introduction-to-r/chapter-1-intro-to-basics-1?ex=6

GmailMyClassUMSLPU LiveLinkedInW3 SchoolsLPU ColabHackerRankLeetCodeKaggle Your Home...GitHubGrammarlyJioCinema - Watch...Online Java Compiler

Learn / Courses / Introduction to R

← Course Outline →

Daily XP 513

Exercise

Apples and oranges

Common knowledge tells you not to add apples and oranges. But hey, that is what you just did, no ->? The `my_apples` and `my_oranges` variables both contained a number in the previous exercise. The `+` operator works with numeric variables in R. If you really tried to add "apples" and "oranges", and assigned a text value to the variable `my_oranges` (see the editor), you would be trying to assign the addition of a numeric and a character variable to the variable `my_fruit`. This is not possible.

Instructions100 XP

✓+100 XP

Awesome, keep up the good work! Continue to the next exercise.

Adjust the code so that R knows you have 6 oranges and thus a fruit basket with 11 pieces of fruit.

Take Hint (-30 XP)

PRESS ENTER TOContinue

scriptLR

Light Mode

```
1 # Assign a value to the variable my_apples
2 my_apples <- 5
3
4 # Fix the assignment of my_oranges
5 my_oranges <- 6
6
7 # Create the variable my_fruit and print it out
8 my_fruit <- my_apples + my_oranges
9 my_fruit
```

Run CodeSubmit Answer

R Console

Create the variable my_fruit and print it out

my_fruit <- my_apples + my_oranges

my_fruit

[1] 11

>

Basic data types in R | R

campus.datacamp.com/courses/free-introduction-to-r/chapter-1-intro-to-basics-1?ex=7

Learn / Courses / Introduction to R

Course Outline

Daily XP 600

Exercise

Basic data types in R

R works with numerous data types. Some of the most basic types to get started are:

- Decimal values like 4.5 are called **numerics**.
- Whole numbers like 4 are called **Integers**. Integers are also numerics.
- Boolean values (TRUE or FALSE) are called **logical**.
- Text (or string) values are called **characters**.

Note how the quotation marks in the editor indicate that "some text" is a string.

Instructions

100 XP

Change the value of the:

- my_numeric variable to 42.
- my_character variable to "universe". Note that the quotation marks indicate that "universe" is a character.
- my_logical variable to FALSE.

Note that R is case sensitive!

Take Hint (-50 XP)

scriptLR

Light Mode

```
1 # Change my_numeric to be 42
2 my_numeric <- 42
3
4 # Change my_character to be "universe"
5 my_character <- "universe"
6
7 # Change my_logical to be FALSE
8 my_logical <- FALSE
```

Run Code Submit Answer

R Console

Basic data types in R | R

campus.datacamp.com/courses/free-introduction-to-r/chapter-1-intro-to-basics-1?ex=7

Learn / Courses / Introduction to R

Course Outline

Daily XP 618

Exercises

Basic data types in R

R works with numerous data types. Some of the most basic types to get started are:

- Decimal values like 4.5 are called **numerics**.
- Whole numbers like 4 are called **Integers**. Integers are also numerics.
- Boolean values (TRUE or FALSE) are called **logical**.
- Text (or string) values are called **characters**.

Note how the quotation marks in the editor indicate that "some text" is a string.

Instructions

+100 XP

Great work! Continue to the next exercise.

Change the value of the:

- my_numeric variable to 42.
- my_character variable to "universe". Note that the quotation marks indicate that "universe" is a character.
- my_logical variable to FALSE.

Note that R is case sensitive!

Take Hint (-50 XP)

scriptLR

Light Mode

```
1 # Change my_numeric to be 42
2 my_numeric <- 42
3
4 # Change my_character to be "universe"
5 my_character <- "universe"
6
7 # Change my_logical to be FALSE
8 my_logical <- FALSE
```

Run Code Submit Answer

R Console

What's that data type? | R

campus.datacamp.com/courses/free-introduction-to-r/chapter-1-intro-to-basics-1?ex=8

GmailMyClassUMSLPU LiveLinkedInW3 SchoolsLPU ColabHackerRankLeetCodeKaggle Your Home...GitHubGrammarlyJioCinema - Watch...Online Java Compiler

Learn / Courses / Introduction to R

← Course Outline →

Daily XP 700

Exercise

What's that data type?

Do you remember that when you added `5 + "six"`, you got an error due to a mismatch in data types? You can avoid such embarrassing situations by checking the data type of a variable beforehand. You can do this with the `class()` function, as the code in the editor shows.

Instructions100 XP

Complete the code in the editor and also print out the classes of `my_character` and `my_logical`.

Take Hint (-30 XP)

scriptLR

1 # Declare variables of different types
2 my_numeric <- 42
3 my_character <- "universe"
4 my_logical <- FALSE
5
6 # Check class of my_numeric
7 class(my_numeric)
8
9 # Check class of my_character
10 class(my_character)
11
12 # Check class of my_logical
13 class(my_logical)

Run CodeSubmit Answer

R Console

>

What's that data type? | R

campus.datacamp.com/courses/free-introduction-to-r/chapter-1-intro-to-basics-1?ex=8

GmailMyClassUMSLPU LiveLinkedInW3 SchoolsLPU ColabHackerRankLeetCodeKaggle Your Home...GitHubGrammarlyJioCinema - Watch...Online Java Compiler

Learn / Courses / Introduction to R

← Course Outline →

Daily XP 765

Exercise

What's that data type?

Do you remember that when you added `5 + "six"`, you got an error due to a mismatch in data types? You can avoid such embarrassing situations by checking the data type of a variable beforehand. You can do this with the `class()` function, as the code in the editor shows.

Instructions100 XP

Complete the code in the editor and also print out the classes of `my_character` and `my_logical`.

Take Hint (-30 XP)

scriptLR

1 # Declare variables of different types
2 my_numeric <- 42
3 my_character <- "universe"
4 my_logical <- FALSE
5
6 # Check class of my_numeric
7 class(my_numeric)
8
9 # Check class of my_character
10 class(my_character)
11
12 # Check class of my_logical
13 class(my_logical)

Run CodeSubmit Answer

R Console

>
[1] "numeric"
>
[1] "character"
>
[1] "logical"

Continue