

Exercises - Session 6



In case you get stuck anywhere, don't be afraid to ask the coaches! They are here to help and will gladly explain everything to you!

Take notes during the exercises. Even if you never look at them again, they will help you memorise things!

Recap previous sessions

1. Write a method called `get_user_number_input`. When you call the method, it should ask the user for a number with a nice text. The method should then return this number as a number, not a string.

Use your method in the example code:

```
def get_user_number_input
  # your code goes here...
end
```

```
number = get_user_number_input
puts "The user entered the number #{number} and it's a #{number.class}"
```

When you execute it, you should get (user input in **bold**):

```
Hi, please enter a number: 42
The user entered the number 42 and it's a Integer
```

Boolean Operators

2. The following exercise might be a bit dry, but it's quite important in the daily life of a programmer.

Imagine the condition is part of your code, like this:

```
if condition
  puts "A"
else
  puts "B"
end
```

Also, we have the following variables:

```
list = [2, 3, 4]
title = "Ruby Monstas"
```

Analyze the following conditions and note their return type like in the first example. Also write what the code would execute in the if statement above.

Fill this out without trying it on your computer. After that, check your answers with irb.

condition	result (return value)	puts ...
<code>1 < 2</code>	true	A
<code>title.include?("Ruby")</code>		
<code>list.length == 3</code>		
<code>"test" == 1</code>		
<code>true false</code>		
<code>true && false</code>		
<code>1 < 2 1 > 2</code>		
<code>list.length > 3 && title.length == 12</code>		
<code>!(list.length == 3)</code>		
<code>!(list[1] == 3 10 != 12)</code>		
<code>1 == 1 && !("testing" == 1 1 == 0)</code>		
<code>3 != 4 && !("A" != "a" "Ruby" == "Ruby")</code>		

3. The following is a truth table. For more examples of truth tables, refer to the cheat sheet that belongs to this session. We have two variables called one and two. These are the input of our truth table. The third column contains the output of our truth table. E.g. for the inputs one = true and two = false, the condition results in true.

Input one	Input two	Output condition
false	false	false
true	false	true
false	true	true
true	true	false

- How would you describe this condition with words? When is it true and when is it false?
- Write a formula that satisfies this truth table. You're only allowed to use these elements:
 - one
 - two
 - &&
 - ||
 - !

4. Consider this boolean condition:

```
false && x
```

You can tell what the result will be without knowing the value of `x`. Why?

Optional Part

1. Write methods for the boolean operators `||`, `&&` and `!`. Here are the method signatures, you just need to fill in the bodies:

```
def my_not(a)
  # your code goes here...
end
```

```
def my_and(a, b)
  # your code goes here...
end
```

```
def my_or(a, b)
  # your code goes here...
end
```

When you execute

```
puts my_not(my_and(my_or(true, false), my_or(false, true)))
```

after defining these methods, your program should print `false` as a result.