Exercises - Session 2

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!

<u>Recap Session 1</u>

1. You have two strings: `"Ruby is"` and `"great!"`.

   Combine them to one string which says: `"Ruby is great!"`

2. Strings and Numbers are two different things. The string `"42"` and the number `42` are not the same.

   a) You can ask any object what class it belongs to, by using the `.class` method.

   What does it say for the string `"42"` and the number `42`?

   b) Find out how you can make the number `42` out of the string `"42"`.

   How can you make a string out of a number?

<u>Advanced Command Line</u>

1. Create a file called "program.rb". Create a copy in the same directory called "program-copy.rb".
2. Create a directory called "ruby-files" and move both files into the directory.
3. Change to the directory "ruby-files" and copy both files to the parent directory.
4. What does the argument "-n" do when you call "cp -n"? How can you find the answer by using the command line?

<u>Variables</u>

For this exercise you should create a file with your text editor (for example called variables-exercise.rb). You will write your Ruby code in this file instead of IRB. You can then run your Ruby program (or script) with `ruby variables-exercise.rb` from the command line.

Hint 1: To output the value of something on screen, use `puts`, for example `puts "Hello World!"`. Of course you can also use `puts` with variables!
Hint 2: To allow the user to enter something in your program, use `gets`. For example `my_input = gets` will wait for the user to input something on the keyboard, followed by the return key. It will then save the text the user entered in the variable called `my_input`.

1. Let's write an interactive program. Here's what it should do: we want to provide a Fahrenheit to Celsius temperature converter. This is an example how the program flow could be:

```
Thank you for your interest in the metric system.
Please enter a temperature in Fahrenheit: 100
You entered 100.0° F, let me convert that for you…
It's 37.77777777777778° C. Goodbye!
```

2. Write a similar program that converts the other way around (°C to °F).

Booleans

1. What happens if you compare strings with the comparison operators < and >? What rules are applied here?

2. Check what the result of `"1" == 1` is. Do you understand it? If not, let the coach explain it to you!

3. Write a program that lets the user input a string and then outputs `true` if the string is the same read forwards and backwards. Otherwise it should output `false`. Hint: Use `gets.chomp` for reading user input.

Arrays

1. Write a program that lets the user input 3 strings. Save those strings in an array. Then print the first letter of each of the strings.

2. Find out how to do each of these things with an array (by googling or looking it up in the Ruby documentation):
    a. delete all elements in the array
    b. turn around the order of the elements in the array
    c. sort the array
    d. randomise the order of the array
    e. delete duplicate entries from the array

    Try each of these in IRB.

Optional Part

1. How can you turn the string `"what is happening?"` into `"?gnineppah si tahW"`?

2. How can you copy a directory with all it's content?

3. What does the `chomp` in `gets.chomp` do?

4. What's the difference between array methods with or without exclamation mark? For example

    `shuffle` vs. `shuffle!` To figure this one out, try assigning the result to a variable.