Tutorial 2

for

Ruby Exploration Version 1

Adam Ryan (14395076)

COMP47530

September 21, 2021

Contents

1 Questions **3**

1.1 Exercise 1 - Questions . . . 3

1.2 Exercise 1 - Answers . . . 4

1.3 Exercise 2 - Questions . . . 4

2 Exercise 2 - Answers **6**

2.1 Exercise 3 - Questions . . . 7

2.2 Exercise 3 - Answers . . . 7

2.3 Exercise 4 - Questions . . . 7

2.4 Exercise 4 - Answers . . . 8

2.5 Exercise 5 - Questions . . . 8

2.6 Exercise 5 - Answers . . . 8

1 Questions

1.1 Exercise 1 - Questions

In irb and each of the primitives class and instance_of? test the following to see types of object they are and explain why you get the answers you do:

1. "hello there big boy"
2. 56
3. 34.00
4. 0.222222354454365
5. ("a", "b", "c")
6. +
7. PI
8. Math::PI
9. add
10. hellow
11. hello = 8 and then check hello with class "goodbye"
12. (56 + 45.32)
13. (56 + 45)
14. 5.to_s
15. "5".to_i
16. five.to_s

1.2 Exercise 1 - Answers

Answers

1. "hello there big boy" is a string because it's a string.
2. 56 is an Integer as it has no trailing decimals and is not a string.
3. 34.00 is a float
4. 0.222222354454365 is a float as it has a decimal which is non-zero.
5. ("a", "b", "c") is an array
6. + returns a SyntaxError with .class because it does not have a class variable accessible
7. PI returns uninitialized constant PI (NameError) because it hasn't been initialised.
8. Math::PI returns a float and is 3.14...
9. add returns an undefinied local variable when checking its class
10. hellow returns an uninitialised variable as it hasn't yet been set
11. hello = 8 and then check hello with class returns an integer because we've set the hello variable to be an integer with value 8.
12. "goodbye" returns a string because it's a string.
13. (56 + 45.32) returns a float because one of the values being added is a float
14. (56 + 45) returns an integer because both values are integers
15. 5.to_s returns a string.
16. "5".to_i returns an integer
17. five.to_s returns a variable not initialised error becuase five hasn't been defined.

1.3 Exercise 2 - Questions

The following details the questions in section 2.

1. "hello there big boy".include?("boy")
2. "hello there big boy".include(" big")
3. "hello there big boy".include?(" ere")
4. What happens when you evaluate: ["a", "b", "c"] + ["d"]

5. What happens when you evaluate: ["a", "b", "c"] + "d"
6. Is there an easy way to capitalise words, so "hello" becomes "Hello" ?
7. In the same vein, make "hello" "HELLO".
8. Write a command to print out your name.
9. Write a method to print out your name.
10. Write a method to print out any name.
11. Set up the varibles, maxi, dick and twinko so that they are all assigned numbers but two of them are assigned to the same numbers. Then show with a series of equality tests which ones actually have the same value.
12. If you change the variables with the same number to be a Float and Fixnum does it change the results of the equality tests ?
13. Do a version of these test using strings rather than numbers.

2 Exercise 2 - Answers

1. "hello there big boy".include?("boy") - Returns true
2. "hello there big boy".include(" big") - Returns an error because it's the wrong function name.
3. "hello there big boy".include?(" ere") - Returns false because of the space
4. What happens when you evaluate: ["a", "b", "c"] + ["d"] - It adds all elements in the second list into the first list; that is to say it unions the two.
5. What happens when you evaluate: ["a", "b", "c"] + "d" - It returns a type error as + isn't defined for adding two different types (string and a list)
6. Is there an easy way to capitalise words, so "hello" becomes "Hello" ? - "hello".capitalize
7. In the same vein, make "hello" "HELLO". - "hello".upcase
8. Write a command to print out your name. - puts "Adam Ryan"
9. Write a method to print out your name.

   ```
   def my_name
   puts "Adam Ryan"
   end

   my_name
   ```

10. Write a method to print out any name.

    ```
    def a_name(name)
    puts name
    end

    a_name("Adam Ryan")
    ```

11. Set up the variables, maxi, dick and twinko so that they are all assigned numbers but two of them are assigned to the same numbers. Then show with a series of equality tests which ones actually have the same value.

```
maxi=1
dick=2
twinko=1

maxi===maxi Returns True
maxi===dick Returns False
maxi===twinko Returns True


dick===maxi Returns False
dick===dick Returns True
dick===twinko Returns False


twinko===maxi Returns True
twinko===dick Returns False
twinko===twinko Returns True
```

12. If you change the variables with the same number to be a Float and Fixnum does it change the results of the equality tests ? - Fixnum is depreciated, however due to rounding there can be instances where the same number as a float and fixnum will not evaluate as equal.

13. Do a version of these test using strings rather than numbers. - When the test is done as strings, setting one variable to `"5"` and another to 5 results in these two not equalling. When both are set as strings then they equal.

2.1 Exercise 3 - Questions

What's a predicate?

2.2 Exercise 3 - Answers

Predicate methods in Ruby are those which end in a question mark to highlight that you are 'asking' a question to Ruby when calling these methods. These methods return true or false (and it is breaking convention to not do so).

2.3 Exercise 4 - Questions

Define your own adding method that always adds 5 and 6 together. So, my_add_five_-to_six => 11.

2.4 Exercise 4 - Answers

```
def my_add_five_to_six
5+6
end

my_add_five_to_six
```

2.5 Exercise 5 - Questions

Put this defined method in a file and call it using the ruby command outside of irb

2.6 Exercise 5 - Answers

Copy the above, paste it into a file called eleven.rb and call the file.