Explorations in Ruby: Practicals

Practical 1

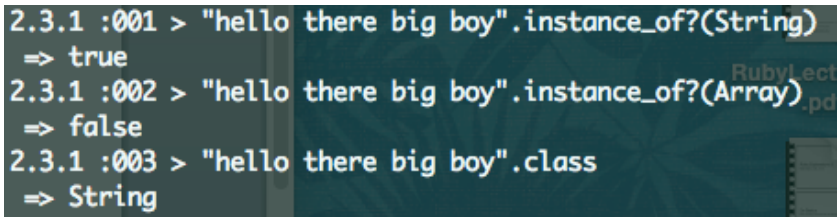
Name: Xiaosheng Liang

Date: 28th September

It is assumed that you have been through all the examples mentioned in the lecture, you need to have typed all of the stuff mentioned there (at least once) and be familiar with what was done.

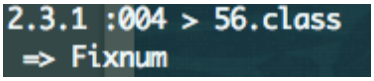
1) In irb and each of the primitives ***class*** and ***instance_of?*** test the following to see what types of object they are and explain why you get the answers you do:

   a. "hello there big boy"



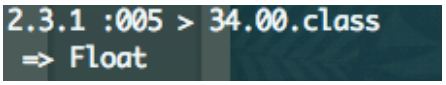
   "hello there big boy" is a String not a Array.

   b. 56



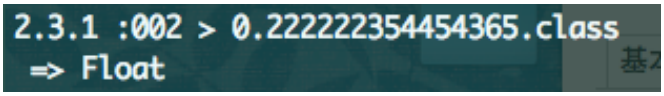
   56 is a Fixnum.

   c. 34.00



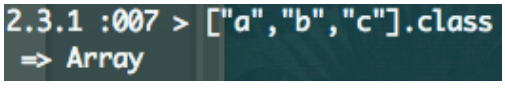
   34.00 is a Float.

   d. 0.222222354454365



   0.222222354454365 is a Float.

   e. ["a", "b", "c"]



   f. +

   It is an operation.

g. PI

PI is a constant.

h. Math::PI

```
2.3.1 :012 > Math::PI.class
 => Float
```

Math:PI is a Float.

i. add

add is a variable.

j. hellow

hello is a variable.

k. hello = 8 and then check hello with ***class***

```
2.3.1 :013 > hello.class
 => Class
2.3.1 :014 > hello=8.class
 => Fixnum
2.3.1 :015 > hello=8.instance_of?(String)
 => false
```

hello=8 is a Fixnum. hello is a Class.

l. “goodbye”

```
2.3.1 :001 > "goodbye".instance_of?(String)
 => true
2.3.1 :002 > "goodbye".instance_of?(Array)
 => false
```

“goodbye” is a String not a Array.

m. (56 + 45.32)

```
2.3.1 :006 > (56+45.32).class
 => Float
```

(56+45.32) is a Float.

n. (56 + 45)

```
2.3.1 :007 > (56+45).class
 => Fixnum
```

(56+45) is a Fixnum.

o. 5.to_s

```
2.3.1 :008 > 5.to_s.class
 => String
```

5.to_s is transforming 5 to string.

p. “5”.to_i

```
2.3.1 :009 > "5".to_i.class
 => Fixnum
```

“5”.to_i means transforming “5” to int.

q. five.to_s

five is a variable and five.to_s doesn’t belong to any class.

2) In irb what happens when you evaluate the following. Try to predict it before trying them:

a. “hello there big boy”.include?(“boy”)

```
2.3.1 :022 > "hello there big boy".include?("boy")
 => true
```

“hello there big boy” includes “boy”.

b. “hello there big boy”.include(“ big”)

```
2.3.1 :026 > "hello there big boy".include?(" big")
 => true
```

“hello there big boy” includes “ big” (space and letters).

c. “hello there big boy”.include?(“ ere”)

```
2.3.1 :025 > "hello there big boy".include?(" ere")
 => false
```

There is no space before letter “ere” in “hello there big boy”.

d. What happens when you evaluate: [“a”, “b”, “c”] + [“d”]

```
2.3.1 :027 > ["a","b","c"]+["d"]
 => ["a", "b", "c", "d"]
```

[“a”, “b”, “c”] + [“d”] is adding [“d”] in [“a”, “b”, “c”].

e. What happens when you evaluate: [“a”, “b”, “c”] + “d”

```
2.3.1 :028 > ["a","b","c"]+"d"
TypeError: no implicit conversion of String into Array
        from (irb):28
        from /Users/LiangXS/.rvm/rubies/ruby-2.3.1/bin/IRB:11:in `<main>'
```

[“a”, “b”, “c”] and “d” are different types.

f. Is there an easy way to capitalise words, so “hello” becomes “Hello” ?

```
2.3.1 :007 > name = "hello"
 => "hello"
2.3.1 :008 > name.capitalize
 => "Hello"
```

g. In the same vein, make “hello” “HELLO”.

```
2.3.1 :005 > name = "hello"
 => "hello"
2.3.1 :006 > name.upcase
 => "HELLO"
```

h. Write a command to print out your name.

```
2.3.1 :030 > puts "Xiaosheng Liang"
Xiaosheng Liang
 => nil
```

i. Write a method to print out your name.

```
1 def name_print
2     puts "Xiaosheng Liang"
3 end
4
5 name_print
```

```
dhcp-892b24ac:ruby LiangXS$ ruby name.rb
Xiaosheng Liang
```

j. Write a method to print out any name.

```
1 def print_name
2     print "The name is: ", ARGV[0]," ",ARGV[1],"\n"
3 end
4
5 print_name
```

```
dhcp-892b276a:ruby LiangXS$ ruby print_name.rb Xiaosheng Liang
The name is: Xiaosheng Liang
```

k. 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.

```
2.3.1 :011 > maxi=2
 => 2
2.3.1 :012 > dick=4
 => 4
2.3.1 :013 > twinko=2
 => 2
2.3.1 :014 > maxi==dick
 => false
2.3.1 :015 > maxi==twinko
 => true
2.3.1 :016 > dick==twinko
 => false
```

l. If you change the variables with the same number to be a Float and Fixnum does it change the results of the equality tests ?

```
2.3.1 :019 > maxi=2.0
 => 2.0
2.3.1 :020 > twinko=2
 => 2
2.3.1 :021 > maxi==twinko
 => true
```

maxi and twinko are not same type but same value.

m. Do a version of these test using strings rather than numbers.

```
2.3.1 :023 > maxi="abc"
 => "abc"
2.3.1 :024 > twinko="cba"
 => "cba"
2.3.1 :025 > dick="abc"
 => "abc"
2.3.1 :026 > maxi==dick
 => true
2.3.1 :027 > maxi==twinko
 => false
```

3) What's a predicate ?

Predict is a method which end with a question mark '?' return either 'true' or 'false'. For example:

```
2.3.1 :002 > "hello there big boy".include?(" big")
 => true
2.3.1 :003 > 7.between?(10,20)
 => false
2.3.1 :004 > "good".instance_of?(String)
 => true
```

4) Define your own adding method that always adds 5 and 6 together.

So, my_add_five_to_six => 11.

```
2.3.1 :011 > print("my_add_five_to_six=>",5+6,"\n")
my_add_five_to_six=>11
 => nil
```

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

```
1 def two_numbers_addtion
2     print("my_add_five_to_six=>",5+6,"\n")
3 end
4
5 two_numbers_addtion
```

```
dhcp-892b24ac:ruby LiangXS$ ruby addtion.rb
my_add_five_to_six=>11
```