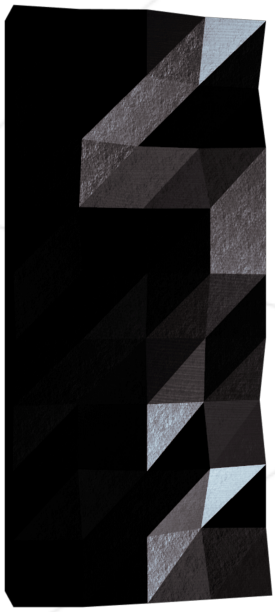

Parseltongue Piscine - Day02

Arrays; Numbers; Booleans

Kai kai@42.us.org

Summary: Learn about arrays, indexing, for loops, ARGV, floats, integers, booleans, numeric bases, and irb

Contents

Eat, Sleep, Code, Repeat.

Chapter I

Don’t Panic!

Confused about how to begin? Not sure what the PDF means?

Don’t worry, we are not perfect :)

And the PDFs are challenges - not walkthroughs. Team up with your group, your partner, your mentor and your other peers to decipher what to do.

You are the master of your own destiny! Go forward and code the world !

Chapter II

Exercise 0: Brainstorm

	Brainstorm
Topics to study :	
Files to turn in : `00_brainstorm.rb or 00_brainstorm.py`	
Notes : `n/a`	

Create a program called `00_brainstorm.py` which imitates a version of the game Scattegories. Here are some goals for it:

- Your game should have a built in list or tuple which contains a selection of categories. When the game begins, choose one of the categories from the list randomly. (Research: How to choose a random number between 0 and n?)
- Once the random category is chosen, the program displays it and loops 10 times to collect input from the user.
- Every answer the user gives is added to another list, the answers lists.
- After 10 entries, display each answer that was given.
- The answers should be displayed neatly. Print them in a column that is centered in the middle of the terminal, with a blank line between each answer.
- Bonus: Time how long it takes the user to enter 10 items for that category, and display the elapsed time at the end.
- Bonus: Draw a nice box around the answer table output.

Chapter III

Exercise 1: ARR Matey

	ARR, Matey
Topics to study :	
Files to turn in : `01_arrmatey.rb or 01_arrmatey.py`	
Notes :	

- Create a script `01_arrmatey.py` which takes a sentence worth of command-line arguments, splits them into an array, and then prints them each out on a different line along with the corresponding index of the array.

- Next, sort the array by word length and reverse it, printing just the words in descending order of length.

```
?> python 01_arrmatey.py ruby-doc.org shows comprehensive functions with arrays and strings :\)
Argv of 0 is ruby-doc.org
Argv of 1 is shows
Argv of 2 is comprehensive
Argv of 3 is functions
Argv of 4 is with
Argv of 5 is arrays
Argv of 6 is and
Argv of 7 is strings
Argv of 8 is :)
comprehensive
ruby-doc.org
functions
strings
arrays
shows
with
and
:)
?>
```

Chapter IV

Exercise 2: Numeric Types

	ARR, Matey
Topics to study :	
Files to turn in : `02_numtypes.rb or 02_numtypes.py`	
Notes : `Ruby` Numeric `Python` Numeric Types	

Write a program that takes two numbers as command line arguments.

They will come into your program as Strings. Find a way to turn them into numbers with which you can perform math.

Divide the first number by the second one and print out both the integer quotient and the remainder.

Then, your program should declare and initialize four variables of different numeric types. Print them out and use the built-in functions type() or .class to print the variable type of each.

```
?> python 02\_numtypes.py 142 6
142 divided by 6 equals 23 remainder 4
Variable a contains : 10  which is of type: Integer
Variable b contains: 56.99  which is of type: Float
Variable c contains: 2+3i  which is of type: Complex
...
```

Chapter V

Exercise 3: 101010

	101010
Topics to study :	
Files to turn in : `03_binary.rb or 03_binary.py`	
Notes :	

- Create a script `03_binary.py` which takes in a number in base 10 and prints out its equivalent in base 2 (binary), in base 8 (octal), and in base 16 (hexadecimal).

```
?> python 03_binary.py 94555
10111000101011011
270533
1715B
?>
```

There are two types of people in the world: those who understand binary, and those who donut.

Real hint: there is a built in function for this... search around!

Chapter VI

Exercise 4: George Bool

	George Bool
Topics to study :	
Files to turn in : `04_bool.rb or 04_bool.py`	
Notes : `Ruby nilClass, TrueClass, FalseClass Python Constants, Types`	

Using the three provided arrays:

Ruby
[false,true,true,nil,true,nil,nil,false,false,nil,true,false]
["or","or","or","==","!=","==","and","==","!=","and","!=","and"]
[false,false,nil,nil,true,true,false,true,nil,false,true,nil]

Python
["False",True,True,None,True,None,None,False,False,None,True,False]
["or","or","or","==","!=","==","and","==","!=","and","!=","and"]
[False,False,None,None,True,True,False,True,None,False,True,None]

Write a program which tests boolean logic by combining one element from each array into an equation and printing the full equation with its result. You can either use the arrays as given or have your program randomize them.

```
?> ruby 04_bool.rb
false or false => false
true or false => true
true or false => true
nil == nil => true
true != true => false
...
```

There are several ways to do this. Sophisticated ways include .send or getattr(). Simpler ways involve using a series of if statements to perform the operation requested. Practicing these equations in the terminal by typing "irb" or "python" would be a helpful way to start.