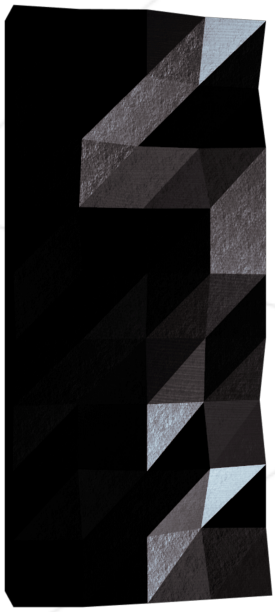

Arrays ~/~Lists

Learn to Sort

Kai kai@42.us.org

Summary: Keep a friend at your side, and study in the path of inspired ones who have given their knowledge out for free.

Contents

Chapter I

Before you Start!

Create your project folder:

1. From your project page on intra, copy the git repository link. Now, in the terminal type "git clone " and paste the link. After the link and before pressing enter, write a name for the new folder. Cloning your git repository always creates a new folder.

2. cd into the folder you just created and from now on, save your work there. Use the command "mkdir <name>" to create new folders. Put each puzzle from this project in a folder with the same name.

Chapter II

Format your Code

Each 42 challenge you turn in must adhere to the following format:

```ruby
#!/usr/bin/env ruby

# This is what my program does
# By <userid>

def function_a
 #code
end

def function_b
 #code
end

def main(ARGV)
 #main method
 function_a
 function_b
end

main(ARGV)
```

- Always begin with the "#!/usr/bin/env ruby" statement. This tells your terminal to run the program using Ruby. In python, the first line is "#!/usr/bin/env python".
- Always add a comment stating what this program is for, some hints to help others use or understand it, and your name or intra ID.
- Do not write any code outside of functions except for one line, at the end of your program, which calls the main() function.
- The (ARGV) parameter is not always needed. In Python it is sys.argv.

Reference your chosen intro to coding class to learn about functions/methods (The keyword "def" means "define function...").

Chapter III

Exercise 0: Arrays/Lists

	Exercise
Turn-in directory : `ex00`	
Files to turn in : `ex00.rb or ex00.py`	
Notes : `Ruby` Array, `Python` List, More on Lists	

Take in a set of numbers as command line arguments. Store them as an array and print out the min, max, mean, median, mode and range of the set.

```
?> ruby ex00.rb 142 6 13 36 54 4 9 78 78 102
Min: 4
Max: 142
Mean: 52.2
Median: 45
Mode: 78
Range: 138
```

Chapter IV

Exercise 1: ARR Matey

	Exercise
Turn-in directory : `ex01`	
Files to turn in : `ex01.rb or ex01.py`	
Notes : `n/a`	

- Create a script `ex01.rb` 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.

```
?> ruby ex01.rb 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 V

Grade your work!

Turn in your work by typing three commands in order:

- git add *
- git commit -m "<your comments here>"
- git push
- If you have an error during the git push, you may need to refresh your authentication ticket. Do this by typing "kinit <username>" and then typing your intra password.

Then, go to your project page and click "Set the project as finished". Next click "Subscribe to defense" and schedule two peer corrections. If you run out of correction points (check your number on your profile page!), it means you need to open correction slots and correct other people in return. :)