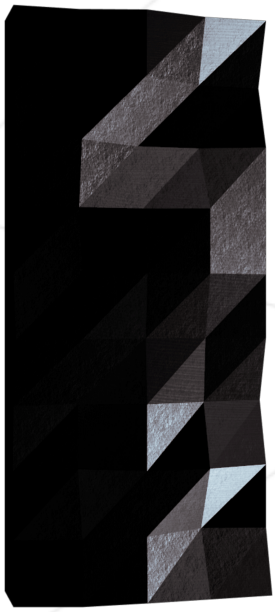

Numbers

Learn to Count

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: Numbers and Casting

	Exercise
Turn-in directory : `ex00`	
Files to turn in : `ex00.rb or ex00.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.

```
?> ruby ex00.rb 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 IV

Exercise 1: Conditional Sum

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

- From Project Euler, a great resource for programming practice: https://projecteuler.net/problem=1
- If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The sum of these multiples is 23.
- Create a script `ex01.rb` which finds the sum of all the multiples of 3 or 5 below the number given as a command line argument.
- For this version, if given a negative number you must also find the sum of all multiples of 3 and 5 between that number and zero.

```
?> ruby ex01.rb 42
408
?>
```

```
?> ruby ex01.rb 420
40950
?>
```

```
?> ruby ex01.rb 4242
4198308
?>
```

```
?> ruby ex01.rb -10
-23
?>
```

```
?> ruby ex01.rb 0
0
?>
```

Chapter V

Exercise 2: Prime Suspects

	Exercise
Turn-in directory : `ex02`	
Files to turn in : `ex02.rb or ex02.py`	
Notes : `n/a`	

- Create a script `ex02.rb` that prints the prime factors, in increasing order, of the number given as an argument.
- If the input given is not a number or is less than one, print only a newline.

```
?> ruby ex02.rb 29
29
?>
```

```
?> ruby ex02.rb 242
2,11,11
?>
```

```
?> ruby ex02.rb 60
2,2,3,5
?>
```

```
?> ruby ex02.rb nineteen ninety six
?>
```

```
?> ruby ex02.rb 1
1
?>
```

Chapter VI

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. :)