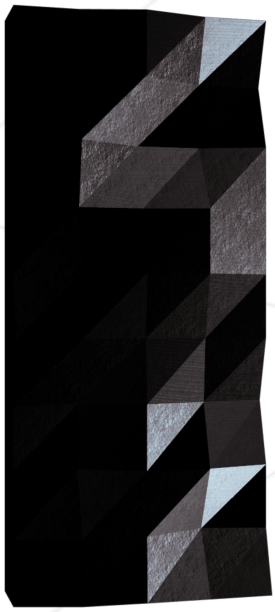

Hacker, Baby

Show us your skillz

Kai kai@42.us.org

Summary: This project is the doorway into the full pathway of HackHighSchool programming challenges. We are checking to see if you know a few things - how to repeat, how to make decisions, how to encapsulate code. Can you take data into your program; check the docs for options; discover built in data structures; and use logic to structure a program?

Contents

Chapter I

Introduction

Welcome to HackHighSchool!

Take a deep breath, look around. This is your oasis for expressing creativity through code. Here is a place for hanging out with friends who have the skills to make things happen. Take a moment to caress the edges of your lovely iMac - aren't they wonderful? Take care of the shared lab space so that we can share it with many people to come; that includes especially, most importantly, no drinks on the table. Keep only water in the lab and keep it on the floor. Capiche.

The first checkpoint of HackHighSchool is to learn the basic controls of how to use a programming language. It's like learning how to drive a bicycle or a car: there's a few features shared between all vehicles. Turn right, turn left, move forward, slow down, stop.

We'd love for you to be bilingual or even multilingual programmers. Many of the projects we are going to offer are open-language: you can turn in a solution in any programming language you like, so long as it outputs the right program behavior. But our first stop is Ruby. We'll give everyone a shared experience by learning some Ruby syntax to start it all off.

Chapter II

Vogsphere

Quick refresher:

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, 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.

3. Each day, turn in your work so far 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.

This is how your directory structure should look:

```
?> pwd
~/kai/hackhighschool
?> ls
hacker_baby
first_day
knowledge_libre
?> cd hacker_baby & ls
ex00
ex01
ex02
```

```
ex03
ex04
ex05
ex06
?> cd ex00 & ls
ex00.rb
```

Inside the project folder are six folders names ex00 ... ex06. Inside each ex0X folder is a .rb or .py file with the same name.

Chapter III

Guidelines

- This project is corrected by Moulinette, our computer grader.
- Read the examples carefully. Your output should match the examples exactly as well as fulfill the instructions.

You can retry the project as many times as you need until you get it right! You don't need to go through the peer correction process to see your result.

Don't use any libraries or gems that you have to install yourself; they won't be available on the computer that grades you.

Chapter IV

Exercise 00 : What is your name

	Exercise 00
description	
Turn-in directory : $ex00/$	
Files to turn in : `ex00.rb`	
Allowed functions : `All`	
Notes : `n/a`	

- Create a script `ex00.rb` which asks your name and greets you with it.

```
?> ruby ex00.rb
Hello hacker, what is your name?
?> O'Brian
Welcome, O'Brian.
```

Chapter V

Exercise 01 : Who Goes There

	Exercise 01
description	
Turn-in directory : $ex01/$	
Files to turn in : `ex01.rb`	
Allowed functions : `All`	
Notes : `n/a`	

- Create a script `ex01.rb` which asks your name and only greets you if your name is “Daenerys of the House Targaryen, the First of Her Name, The Unburnt, Queen of the Andals, the Rhoynar and the First Men, Queen of Meereen, Khaleesi of the Great Grass Sea, Protector of the Realm, Lady Regnant of the Seven Kingdoms, Breaker of Chains and Mother of Dragons” or "DHTFHNUQARFMQMKGSPRLRSKBCMD" for short.

- Otherwise, if your name is "Dany", the program replies "Dany who?".

- For any other name, the program replies "Move along, now."

```
?> ruby ex01.rb
Who goes there?
?> DHTFHNUQARFMQMKGSPRLRSKBCMD
Welcome, Daenerys.
```

```
?> ruby ex01.rb
Who goes there?
?> Dany
Dany who?
```

```
?> ruby ex01.rb
Who goes there?
?> Jaqen H'gar
Move along, now.
```

Chapter VI

Exercise 02 : ARR Matey

	Exercise 02
description	
Turn-in directory : *ex*02/	
Files to turn in : `ex02.rb`	
Allowed functions : `All`	
Notes : `n/a`	

- Create a script `ex02.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 ex02.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 VII

Exercise 03 : Conditional Sum

	Exercise 03
description	
Turn-in directory : $ex03/$	
Files to turn in : `ex03.rb`	
Allowed functions : `All`	
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 `ex03.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 ex03.rb 42
408
?>
```

```
?> ruby ex03.rb 420
40950
?>
```

```
?> ruby ex03.rb 4242
4198308
?>
```

```
?> ruby ex03.rb -10
-23
?>
```

```
?> ruby ex03.rb 0
0
?>
```

Chapter VIII

Exercise 04 : Prime Suspects

	Exercise 04
description	
Turn-in directory : *ex*04/	
Files to turn in : `ex04.rb`	
Allowed functions : `All`	
Notes : `n/a`	

- Create a script `ex04.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 ex04.rb 29
29
?>
```

```
?> ruby ex04.rb 242
2,11,11
?>
```

```
?> ruby ex04.rb 60
2,2,3,5
?>
```

```
?> ruby ex04.rb nineteen ninety six
?>
```

```
?> ruby ex04.rb 1
1
?>
```

Chapter IX

Exercise 05 : Memorize

	Exercise 05
description	
Turn-in directory : $ex05/$	
Files to turn in : `ex05.rb`	
Allowed functions : `All`	
Notes : `Use the capitols.txt file provided on the project page. You do not need to turn that file in, but you can include it in your repository.`	

- Create a script `ex05.rb` which reads in the provided comma-delimited file of US States and capitals and stores this information in a hashtable.
- Next, on an infinite loop, print "Ready: " and wait for the user to enter the name of a state or capital. For each query print out the associated capital or state and go back to Ready state.
- The program exits when the user types "Done". If the input is invalid, answer "nil".

```
?> ruby ex05.rb capitals.txt
Ready: Arizona
Phoenix
Ready: Montana
Helena
Ready: MacaroniAndCheese
nil
Ready: Pierre
South Dakota
Ready: Done
?>
```

Chapter X

Exercise 06 : Do You Even 101010

	Exercise 06
description	
Turn-in directory : *ex*06/	
Files to turn in : `ex06.rb`	
Allowed functions : `All`	
Notes : `n/a`	

- Create a script `ex06.rb` which takes in a number in base 10 and prints out its equivalent in base 2 (binary).

```
?> ruby ex06.rb 94555
10111000101011011
?>
```

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