Parseltongue Piscine - Day01

Strings, Input-Output, Equality, Integers, If/Else

Kai kai@42.us.org

Summary: Learn about the following: Strings, Printing, Variable Assignment, Format strings aka String Interpolation, Input, String methods, If-Else, Index of a string, testing equality with ==, and Integers.

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: So Long and Thanks for All the Fish

	So Long and Thanks for All the Fish
Files to turn in : `00_so_long.py`	
Notes : `n/a`	

You're going to write a Python program. That means you write Python code in a text editor, and save it as a file called 00_so_long.py. Whenever you want to see what your code does, run it by typing "python 00_so_long.py" in Terminal after navigating to the folder which contains that file.

The idea of this project is to write an ad-lib. The first thing your program should do is define six different variables and fill each variable with a string of your choice that matches the description.

```
animal_plural = "monkeys"
emotion = "perturbed"
character_trait = "wholesomeness"
adjective = "vivacious"

print("So long and thanks for all the {}".format(animal_plural))
# Continue writing your own code here :)
```

Next, your program should print the first half of the "So Long And Thanks for All The Fish" song with your ad-lib words filled in. Use the syntax in the example above for filling in variables to the lines you print out.

Here's a template for how to fill in the lyrics:

```
So long and thanks for all the {animal_plural}
So {emotion} that it should come to this
We tried to warn you all but oh dear?

You may not share our {character_trait}
```

```
Which might explain your disrespect
For all the {adjective} wonders that
grow around you

So long, so long and thanks
for all the {animal_plural}
```

And here's an example of the output:

```
So long and thanks for all the monkeys
So perturbed that it should come to this
We tried to warn you all but oh dear?

You may not share our wholesomeness
Which might explain your disrespect
For all the vivacious wonders that
grow around you

So long, so long and thanks
for all the monkeys
```

After this first half has been printed, create one new `variable` and re-use the other variables from above. Assign a new value to each of the reused variables...

```
animal_2 = # fill in your own values!
emotion =
character_trait =
adjective =
```

And then, print the second half, with your variables in the spots like this:

```
The world's about to be destroyed
There's no point getting all {emotion}
Lie back and let the planet dissolve

Despite those nets of {animal_2} fleets
We thought that most of you were {emotion}
Especially {adjective} tots and your
pregnant women

So long, so long, so long, so long, so long
So long, so long, so long, so long, so long

So long, so long and thanks
for all the {animal_plural}
```

Chapter III

Exercise 1: Ready Player One

	Ready Player One
Files to turn in : `01_player.rb or 01_player.py`	
Notes : `Built-in Functions`, `System`	

- Create a script `01_player.py` which asks your name and waits for you to type something in. After you type an input, then it assigns that input to a variable, and prints out another message to greet you by your name.

  Here is an example of the program running:

```
?> python 01_player.py
Hello hacker, what is your name?
?> Parzival
Welcome, Parzival!
```

Chapter IV

Exercise 2: Daenerys

	Daenerys
Files to turn in : `02_daenerys.rb or 02_daenerys.py`	
Notes : `Control Flow`	

- Create a program called `02_daenerys.py` 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."

```
?> python 02_daenerys.py
Who goes there?
?> DHTFHNUQARFMQMKGSPRLRSKBCMD
Welcome, Daenerys.
```

```
?> python 02_daenerys.py
Who goes there?
?> Dany
Dany who?
```

```
?> python 02_daenerys.py
Who goes there?
?> Jaqen H'gar
Move along, now.
```

Chapter V

Exercise 3: Palindrome

	Palindrome
Files to turn in : `02_palindrome.rb or 02_palindrome.py`	
Notes : `The goal of this project is to learn to work through strings by accessing individual letters through their index. You will also need to research substring substitutions.`	

Create a program called `03_palindrome.py` which takes in a sentance, word, or number and tells you if it is a palindrome or not.
Here's a few things to think about:

- The phrase "A man, a plan, a canal: Panama." is a valid palindrome. You don't want your program to ignore because of the spaces and punctuation marks, do you? You will have to find a way to handle spaces, capitalization, and punctuation.

- How can you run tests of equality on specific letters in a string?

- Your program will be nicer quality if it prints out a prompt, like "Enter the text which may be a palindrome:", and then a readable answer, like "So cool! This text IS a palindrome." At the bare minimum, it must print out 'true' or 'false'.