



# Parseltongue Piscine - Day01

## Input, Output, and Variable Types

Kai [kai@42.us.org](mailto:kai@42.us.org)

*Summary: Learn about the following: Strings, Printing, Variables, Input, Integers and Floats.*

# Contents

<b>I</b>	<b>Don't Panic!</b>	<b>3</b>
<b>II</b>	<b>Variables, statements, and expressions</b>	<b>4</b>
II.1	FOPP Chapter 2 . . . . .	4
II.2	Exercise 0: So Long and Thanks for All the Fish . . . . .	4
II.3	Exercise 1: Ready Player One . . . . .	6
<b>III</b>	<b>Numeric Types</b>	<b>7</b>
III.1	FOPP Chapter 3 . . . . .	7
III.2	Exercise 2: Working with Numbers . . . . .	7
III.3	Exercise 3: 101010 . . . . .	8



Eat, Sleep, Code, Repeat.

# Chapter I

## Don't Panic!

Remember the three commandments of 42:

1. Ask the person on the left of you
2. Ask the person on the right of you
3. Read the Manual (i.e. the [documentation](#))!


# Chapter II

## Variables, statements, and expressions

### II.1 FOPP Chapter 2

Go to [Runestone: Fundamentals of Python](#) and complete section 2 before the first exercise.

### II.2 Exercise 0: So Long and Thanks for All the Fish

	So Long and Thanks for All the Fish
Topics to study : Variables, Printing, String Manipulation	
Files to turn in : 00_so_long.py	
Notes : <a href="#">Variables</a> , <a href="#">Printing</a> , <a href="#">String Manipulation</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}
```

## II.3 Exercise 1: Ready Player One

	Ready Player One
Topics to study : Input	
Files to turn in : 01_player.py	
Notes : <a href="#">Built-in Functions</a> , <a href="#">System</a> , <a href="#">input</a>	

- 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 III

## Numeric Types

### III.1 FOPP Chapter 3

Go to [Runestone: Fundamentals of Python](#) and complete section 3 before the following exercises.

### III.2 Exercise 2: Working with Numbers

	Ready Player One
Topics to study : Variable types, floats	
Files to turn in : 02_numtypes.rb or 02_numtypes.py	
Notes : <a href="#">Numeric Types</a> , <a href="#">Built-in Functions</a>	


Write a program that asks you your age, then tells you how old you will be in 5 years, and how old you will be when you have lived 10 times as long as you have now. Then, divide your age by 3, and print out both the whole number quotient and the remainder.

`Input()` will bring the age into your program as a String. You'll have to change its type to make it into a number with which you can perform math.

```
?> python 02_numtypes.py
Hello hacker, what is your age?
?> 14
In five years you will be 19.
When you live ten times as long you will be 140.
Your age divided by 3 equals 4 remainder 2.
```



### III.3 Exercise 3: 101010

	101010
Topics to study : Base systems (Binary, Hexadecimal, Octal)	
Files to turn in : 03_basicbases.py	
Notes : <a href="#">Binary and Hexidecimal Conversions</a>	

- Reference the documentation about [Python3 Built-in Functions](#) to create a script, `03_basicbases.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
Enter a number:
?> 94555
0b10111000101011011
0o270533
0x1715B
?>
```