Lesson 3: Input/Output

Print

The main way we output information in python is through the **print()** function. We've already used this quite a bit, but here's a quick refresher:

```
print("Hello, World!")
```

We can also use the print function to output multiple things with spaces in between:

```
print("A", "B")
```

```
pschwendy@Peters-MacBook-Pro Lesson 3 % python print.py
A B
```

Input

Programs often have to take in input. In python, we can do this with the *input()* function and store the input in a variable. Let's take in an input and print it back to the user:

```
x = input()
print(x)
```

```
pschwendy@Peters-MacBook-Pro Lesson 3 % python input.py
Bananas
Bananas
```

1. I input the word "Bananas" (1st time)
2. The program prints "Bananas" back to me

You can also pass a string into the input function to preface the input with an output

```
age = input("How old are you? ")
```

Exercise 1

Create a program that asks the user for their name and then says, “Hi, [name]!” back to the user. For example:

```
What's your name? Peter
Hi, Peter!
```

Hint: use both the input() and print() functions!

Basic Operations

Another way to use variables is by performing operations. We can do this using the standard operators (+, -, *, /, %, etc.)

For example, we can add or subtract two numbers:

```
a = 17
b = 10
c = a + b   # c = 27
d = a - b   # d = 7
```

We can also add strings. This is called *string concatenation*:

```
a = "Hi, "
b = "Shane!"
c = a + b   # c = "Hi, Shane!"
```

Don’t worry about the term for now. We’ll talk more about that when we look deeper into strings.

Exercise 2

Create a program that asks the user for two numbers and outputs the sum of those two numbers. For example:

```
Give me two numbers: 1 2
3
```

Hint: Remember variable modulation

Summary

1. The **print()** function allows to output information to the command line
2. The **input()** function allows us to read input from the command line
3. Basic operations—like +, -, *, /, …—allow us to combine and edit variables