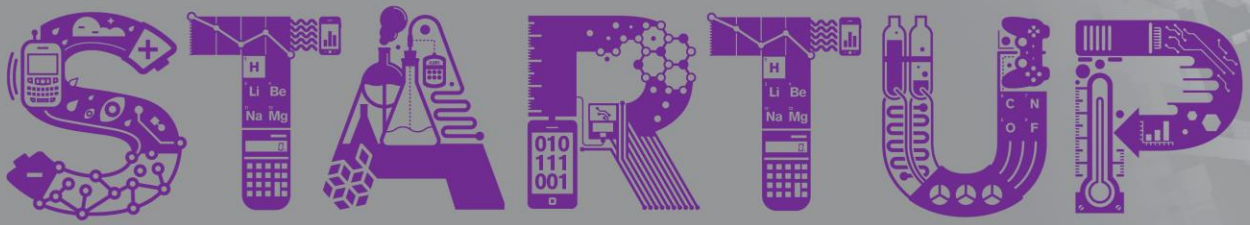
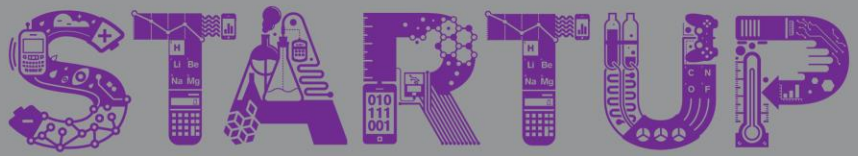
Bloomberg



Bloomberg



Python Lesson 2

Introduction to Functions

Review: Variables

```
x = 7
y = 1.2
print(x + y)
print(4 * y)
print(x * x)
```

```
8.2
4.8
49
```

```
x = "Hello, "
y = "Johnny"
print(x + y)
```

```
Hello, Johnny
```

```
name = "Johnny"
print("Hello,", name)
```

```
Hello, Johnny
```

```
name = "Sally"
print("Hello,", name)
```

```
Hello, Sally
```

What does this code do?

```
def print_greeting(name):
    print("Hello,", name)

print_greeting("Johnny")
```

Functions: Introduction

Definition: A **function** is a reusable, named piece of code that can take inputs, called parameters.

Example:

```
def print_greeting(name):
    greeting = "Hello, " + name
    print(greeting)

print_greeting("Johnny")
```

Anatomy of a Function

def tells the program you are defining a function

Functions have names, just like variables

You can pass values into the function, called **parameters**.

```
def print_greeting(name):
    greeting = "Hello, " + name
    print(greeting)

print_greeting("Johnny")
```

The function code goes here. It must all be indented the same amount.

The **:** tells the program you are starting the code block for the function

Inside the function, the parameter works like a variable.

Example 2: print is a function!

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

```
Hello World!
```

What is the parameter?

Example 3:

What does this function do?

```
def print_square_of_number(number):
    square = number * number
    print(square)
```

What happens when I run this?

```
print_square_of_number(7)
```

Example 4:

We want to write a function that prints something 4 times:

```
print_four_times("Woohoo!")
```

Output:

```
Woohoo!
Woohoo!
Woohoo!
Woohoo!
```

Example 4:

First we can set up the function definition:

```
def print_four_times(thing_to_print):
```

Example 4:

Now we can fill in the logic: What should this function do?

```
def print_four_times(thing_to_print):
    print(thing_to_print)
```

Example 4:

Now we can fill in the logic: What should this function do?

```
def print_four_times(thing_to_print):
    print(thing_to_print)
    print(thing_to_print)
```

Example 4:

Now we can fill in the logic: What should this function do?

```
def print_four_times(thing_to_print):
    print(thing_to_print)
    print(thing_to_print)
    print(thing_to_print)
```

Example 4:

Now we can fill in the logic: What should this function do?

```
def print_four_times(thing_to_print):
    print(thing_to_print)
    print(thing_to_print)
    print(thing_to_print)
    print(thing_to_print)
```

Example 6: input function

What does this code do?

```
name = input("Enter your name: ")
print("Your name is", name)
```

```
Enter your name: Sally
Your name is Sally
```

Example 7

```
def ask_and_answer():
    name = input("What's your name? ")
    print("Hello", name)
    day_of_week = input("What day of the week is it? ")
    print("Have a great", day_of_week)
```

```
What's your name? Sally
Hello Sally
What day of the week is it? Monday
Have a great Monday
```

Example 8

The computer treats user input as a string!

```
def square_number_calculator():
    number = input("What do you want to square? ")
    print(number * number)

square_number_calculator()
```

Squaring a string doesn't make sense.

What will happen if we run this function and enter 5?

```
What do you want to square? 5
Traceback (most recent call last):
  File "/home/ubuntu/workspace/introtofunctions.py", line 7, in <module>
    square_number_calculator()
  File "/home/ubuntu/workspace/introtofunctions.py", line 5, in square_number_calculator
    print(number * number)
TypeError: can't multiply sequence by non-int of type 'str'
```

Example 8

```
def square_number_calculator():
    user_input = input("What do you want to square? ")
    number = int(user_input)
    print(number * number)

square_number_calculator()
```

int is another function! It converts a string to an integer

What will happen if we run this function and enter 5?

```
What do you want to square? 5
25
```

Example 9

Functions can take more than one parameter:

```
def multiply_and_add_three(first_number, second_number):
    product = first_number * second_number
    product_plus_three = product + 3
    print("Result is", product_plus_three)
```

Functions: Practice

- Break for 30 minutes to work on exercises.

Python Explorer Game

We can use functions to make our code neater:

```
def print_menu():
    print("Welcome to the Python Explorer.")
    print("You are standing in a dark, shadowy tunnel. At your back is the south wall.")
    print("To walk around, type N to move north and S to move south.")
    print("When you find objects, try commands like look and open. Be creative!")
    print("Use the command 'look around' to look up and down the tunnel")
    print("Good luck!!")
```

Python Explorer Game

Many programs have a Main function, which is the first thing that runs. We can start to build our main function now! What does it do?

```
def main():
  current_location = 0
  print_menu()
  user_input = input("What would you like to do? ")
  print(user_input)
  print("Goodbye")
```

```
Welcome to the Python Explorer.
You are standing in a dark, shadowy tunnel. At your back is the south wall.
To walk around, type N to move north and S to move south.
When you find objects, try commands like look and open. Be creative!
Use the command 'look around' to look up and down the tunnel
Good luck!!
What would you like to do? N
N
Goodbye
```

Recap

- A function is a reusable, named piece of code.
- To write a function, start with

```
def function_name(parameter):
```

- All the code inside a function should be indented the same amount.
- We use lots of functions already: print, input, int