Class 17

Week 8

Functions

Reusability is a key principle of software engineering.

Once you've written some code, you'll want to store it so you can use it again without having to copy and paste.

Functions

This is what functions are for! Functions are a way to wrap up your code so it can be used again by calling an easy shortcut.

Syntax of Functions

```
def my_function():
```

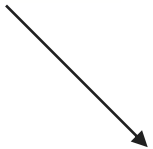
Name function.

Code that represents the function

MORE ABOUT FUNCTIONS

Function names have to follow the same rules as variable names.

```
def my_function():
```



parentheses and a colon

Indent the body of function

Calling Functions

```
def hi():
    print("Hello Mr. John! How are you
    today?")


hi()
```

Function with Parameters

```
def hi(name):
        print("Hello " + name + "! How are you today?")
```

Now you can say hi to John, James, and Maggie, like so:

hi("John")
hi("James")
hi("Maggie")

```
Hello John! How are you today?
Hello James! How are you today?
Hello Maggie! How are you today?
```

Functions With Default Values

```
def hi(name="sir"):
    print("Hello " + name + "! How are you today?")


hi()
hi("James")
```

Return Values

```
def fullname(firstname,lastname):
  print(firstname , lastname)


fullname("john","smith")
```

As you may have guessed, the return statement allows you to store the output of the function in a variable.

```
def fullname(firstname,lastname):
  return firstname,lastname


returned_value = fullname("john","smith")

print("Person name is: ", returned_value)
```

PROBLEM: There is a mathematical function f(x,y) = x^y + 100. Implement the function in Python.

PROBLEM: CHANGE THE FUNCTION BELOW TO RETURN THE GREETING INSTEAD OF PRINTING IT:

```
1. def hi():
2.     greeting = "Hello world!!!"
3.     print(greeting)
```