Today's Topics:

- Error messages
- Functions
- Math module

LEE WILKINS

OFFICE : 6C.9

CONTACT: MIO

420-SN1-RE PROGRAMMING IN SCIENCE

A series of steps that we've already told the computer how to do

When we need the computer to make the same "recipe" multiple times, we can simply call the function instead of telling it all the steps again.

FUNCTIONS

Functions can have **arguments** or **parameters**)

Arguments can be required or optional

The number of arguments depends on the function

Some functions may also require arguments to be a certain type

Functions can also give us something they end (**return** a value)

We’ve already used some built-in functions like print() and type(), but now we're going to write our own

FUNCTIONS

def means we are defining the function

This is your function name

Note the syntax, you always need () so you know its a function

```
def my_function():
    print("this is my function")
```

Function content is indented

Your function ends, here, but it wont be called until you do this:

```
my_function()
```

This is where your function actually runs. This is called calling the function.

FUNCTION SYNTAX

I can pass a parameter through my function. It will be used by this name only within the scope of the function.

Whatever I pass through the function will be placed here

```python
def my_function_param(my_string):
    print(my_string)

my_function_param("this is a string")
my_function_param("I can print anything here")
my_function_param("This function is so useful!")
```

PASSING PARAMETERS

We can call the same function to produce different results

You can create variables inside your function. They only exist inside the function and not int he rest of your program

You can pas multiple parameters through your function

```
def calculate_square_area(width, height):
    area = width * height
    print(f'The area of a square with the width of {width} inches ad the height of 
{height} inches is {area} inches^2')

calculate_square_area(2,4)
# The area of a square with the width of 2 inches ad the height of 4 inches is 8 inches^2
calculate_square_area(5,10)
# The area of a square with the width of 5 inches ad the height of 10 inches is 50 
inches^2
calculate_square_area(100,3)
# The area of a square with the width of 100 inches ad the height of 3 inches is 300 
inches^2
```

MULTIPLE PARAMETERS

The function ends when
the indentation ends

```
def calculate_square_area(width, height):
    area = width * height
    print(f'The area of a square with the width of {width} inches ad the height of 
{height} inches is {area} inches^2')

print(area)

# NameError: name 'area' is not defined
```

Area only exists inside the function. This idea is called scope

SCOPE

```
def return_calculate_square_area(width, height):
    area = width * height
    return(area)

print(return_calculate_square_area(2,4))

my_area = return_calculate_square_area(2,4)
```

The word return is used t
give back a value

In this example we are
returning the area, either
printing it or storing it

RETURN

Current best practice is to use snake_case

However, you will also see camelCase and flatcase sometimes

e.g. analyzeData(), endswith()

Python doesn't care, but humans do!

Humans who might read your code: your future coworkers, your future self, your teacher who will be grading you (me!)

Get into the habit of following best practices

Most importantly: be consistent!

A NOTE ON FUNCTION NAMES...

NEXT CLASS: MORE FUNCTIONS, DOCUMENTATION AND A PRACTICE QUIZ

NEXT CLASS: