Introduction to Python Programming

6 - SUBROUTINES.

Resources created by Andy Colley (@MrAColley)

Learning Goals/Objectives

Be able to read, comprehend, trace, adapt and create Python code that:

- Defines a **subroutine**
- Calls a subroutine
- Creates a subroutine that uses arguments
- Gets user input and use it as an argument in a subroutine
- Returns a value from a subroutine

repl.it

What Is A Subroutine?

- A subroutine gives a single name to a set of actions.
- You create a subroutine by **defining it.**
- You can use the subroutine at any time in your program by **calling** it.

1 - **Define** the subroutine and give it a name.

```
def say_hi():
    print("Hello there!")


say_hi()
```

2 - **Add the code** that you need to complete the task.
Indent code statements!

3 - **Call** the subroutine whenever you need it by typing its name.

repl.it

Naming Subroutines

- Subroutine names **do not** use camelCase
- They use all **lower case** with **underscores** between the words.
- This helps us tell the difference between subroutines and variables/lists when we are reading the code.

```
say_hi()
add_one()
get_input()
```

repl.it

Tracing Subroutines

Defining subroutines

Subroutines are **defined** at the top of your program.

But they do not run until they are **called** in the main program

```
1   def say_hi():
2       print("Why hello there!")
3
4   def offer_drink():
5       print("Would you care for a spot of tea?")
6
7   def offer_food():
8       print("Biscuit?")
9
10  def say_bye():
11      print("Cheerio then.")
12
13
14  offer_drink()
15  say_hi()
16  offer_food()
```

Main program.
Use subroutines

repl.it

Tracing Subroutines

Write down the line numbers in order that they will execute when the program is run.

```
1   def say_hi():
2     print("Why hello there!")
3
4   def offer_drink():
5     print("Would you care for a spot of tea?")
6
7   def offer_food():
8     print("Biscuit?")
9
10  def say_bye():
11    print("Cheerio then.")
12
13  print("Welcome to the hospitality program!")
14  say_hi()
15  print("what's your name?")
16  offer_drink()
17  print("Oh, lovely")
18  offer_food()
```

repl.it

Subroutines That Return A Value

Subroutines That Return a Value (Functions)

- A subroutine can **return** (send) **some data back to the main** (calling) **program**.
- When you do this, you should store the returned value in a variable in the main program.

1 - **Define** the subroutine and give it a name.

```
def adder():
        num1 = 10
        num2 = 15

        return num1 + num2

outputNum = adder()

print(outputNum)
```

2 - Add the code that you need to complete the task.

3 - Use **return** followed by the task that you want to perform

3 - **Call** the subroutine in the main program (NOT INDENTED).

Look at how the subroutine is assigned to a variable.

repl.it

Subroutines That Use Arguments

Subroutines With Arguments

- We can put data into a subroutine. To do this we use **arguments** (aka *parameters*)**.**
- You can think of arguments like variables used by the subroutine.

```
def add_five(num1):
    print(num1 + 5)

add_five(42)
```

1 - The argument name goes in brackets after the subroutine name.

2 - Put the actual data in the brackets when you call the subroutine.
This will be put into the *num1* argument and used by the subroutine.

repl.it

Subroutines With Arguments

- We can get **input from users** and use that as arguments too.

```
def add_five(num1):
 print(num1 + 5)


userInput = int(input("Enter a number"))
add_five(userInput)
```

1 - Get the user to input and save it in a variable

2 - Use the variable as the argument. This will put the data from the userInput variable into the num1 argument.

Subroutines With Arguments

- We can get **input from users** and use that as arguments too.

```
def add_five(num1):
 print(num1 + 5)


add_five(int(input("Enter a number")))
```

Independent Challenge - Calculator

- Define four subroutines - **add, subtract, multiply, divide** that add multiply etc two numbers and return the result.
- Each should have two integer number arguments.
- The user is asked to input two numbers.
- These numbers will be passed as arguments into one of the subroutines.
- The user is asked to input 1 for add, 2 for subtract etc.
- If they input 1, call the ‘add’ subroutine, input 2 calls the ‘subtract’ subroutine etc
- Output the returned result as part of a sentence.

repl.it