Introduction to Python Programming

6 - SUBROUTINES.



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

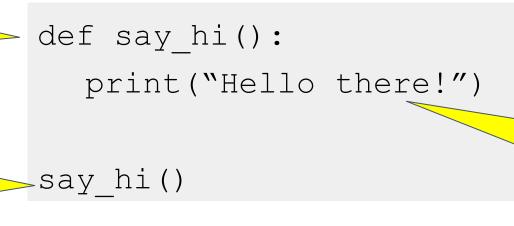


What Is A Subroutine?

- 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.

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



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

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()
```



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

Main program.
Use subroutines

```
repl;it
```

```
def say_hi():
       print("Why hello there!")
     def offer_drink():
       print("Would you care for a spot of tea?")
     def offer_food():
       print("Biscuit?")
     def say_bye():
       print("Cheerio then.")
13
     offer_drink()
14
15
     say_hi()
     offer_food()
16
```

Tracing Subroutines

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

```
def say_hi():
       print("Why hello there!")
 2
 3
     def offer_drink():
 4
 5
       print("Would you care for a spot of tea?")
 6
     def offer_food():
       print("Biscuit?")
 8
 9
     def say bye():
10
11
       print("Cheerio then.")
12
     print("Welcome to the hospitality program!")
13
     say_hi()
14
15
     print("what's your name?")
16
     offer_drink()
17
     print("0h, lovely")
18
     offer_food()
```



Subroutines That Return A Value



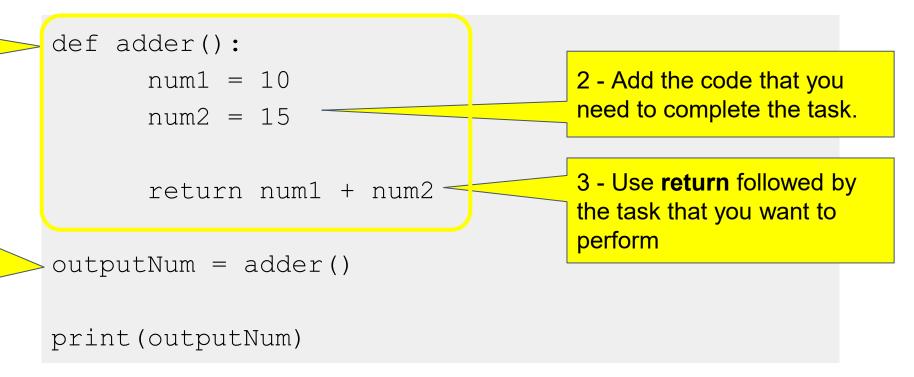
Subroutines That Return a Value (Functions)

- 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.

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

Look at how the subroutine is assigned to a variable.



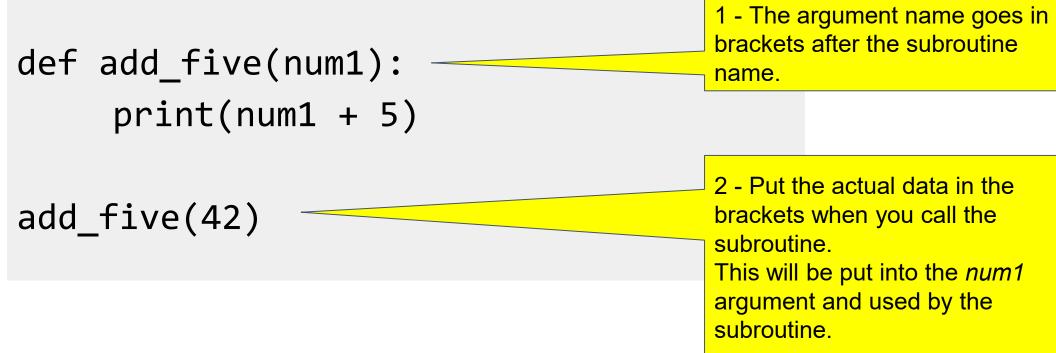


Subroutines That Use Arguments



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.





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)

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.

