Learning Goals/Objectives

Be able to read, comprehend, trace, adapt and create

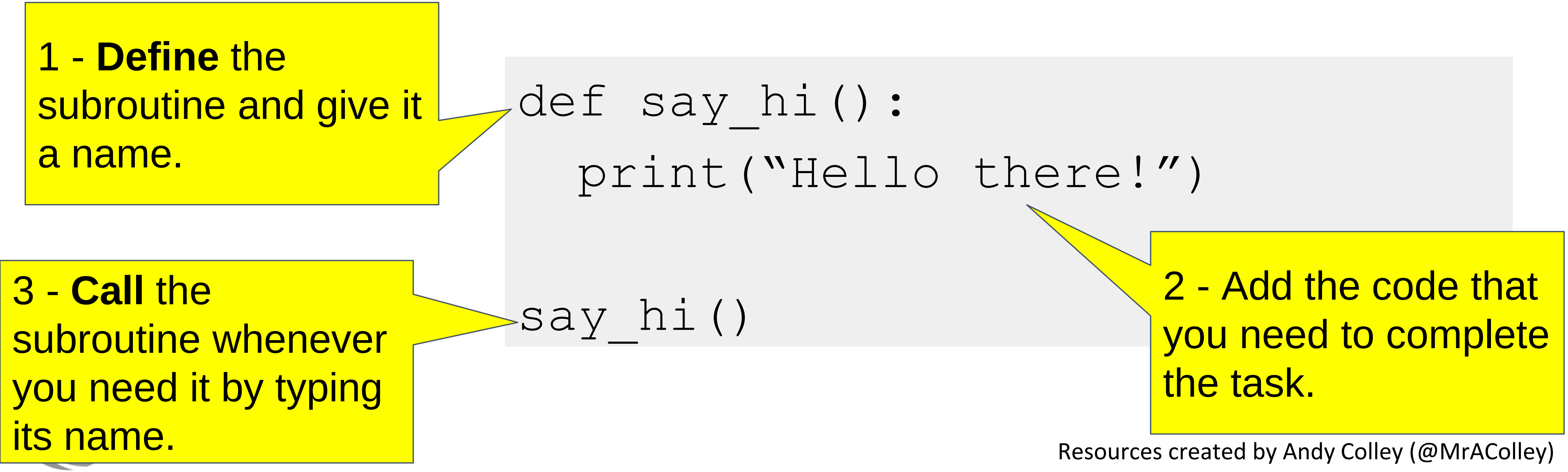
Python code that:

- 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

Resources created by Andy Colley (@MrAColley)

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.

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

Resources created by Andy Colley (@MrAColley)

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()
```

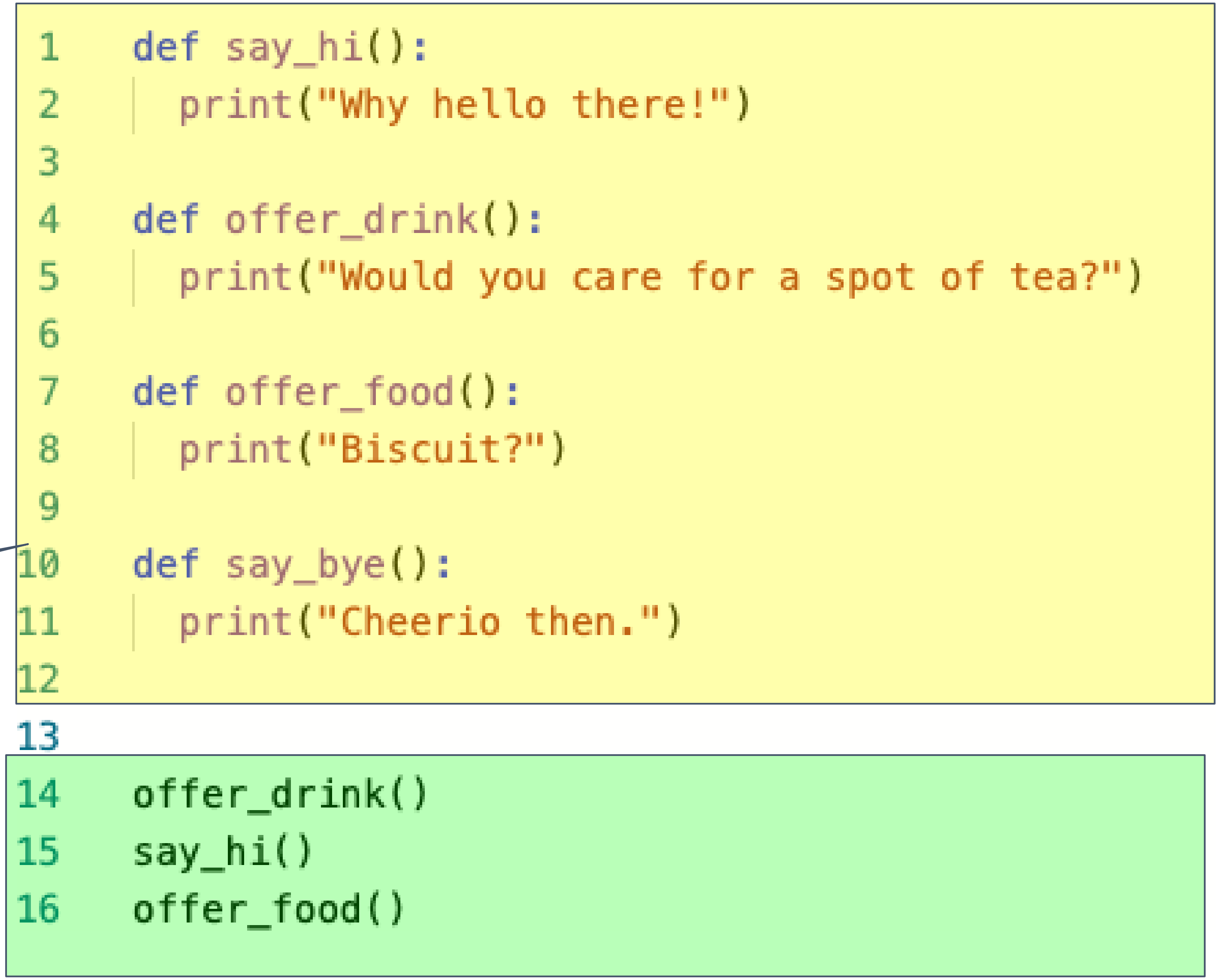
Resources created by Andy Colley (@MrAColley)

Tracing Subroutines

Subroutines are **defined** at the top of your program, but they do not run until they are **called** in the main program

Defining subroutines

Main program

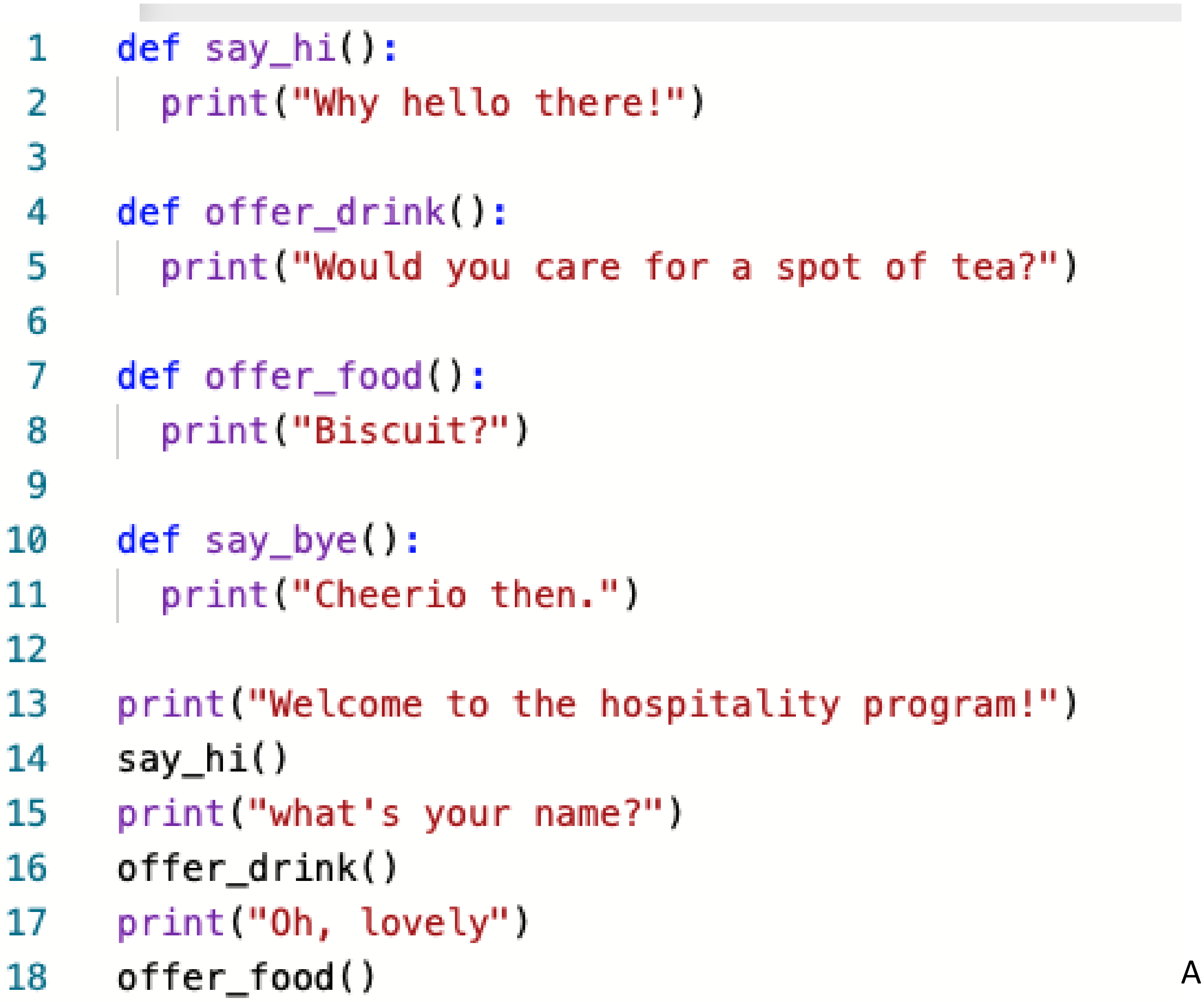


```
def say_hi():
  print("Why hello there!")

def offer_drink():
  print("Would you care for a spot of tea?")

def offer_food():
  print("Biscuit?")

def say_bye():
  print("Cheerio then.")


offer_drink()
say_hi()
offer_food()
```

Resources created by Andy Colley (@MrAColley)

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()
```

Andy Colley (@MrAColley)

Subroutines That Return A Value

Resources created by Andy Colley (@MrAColley)

Subroutines That Return a Value (Functions)

- A subroutine can return (send) some data back to the main 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.

Resources created by Andy Colley (@MrAColley)

Subroutines That Use Arguments

Resources created by Andy Colley (@MrAColley)

Subroutines With Arguments

- We can put data into a subroutine. To do this we use **arguments.**
- 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.

Resources created by Andy Colley (@MrAColley)

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")))
```

Resources created by Andy Colley (@MrAColley)

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 to add, 2 to 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.