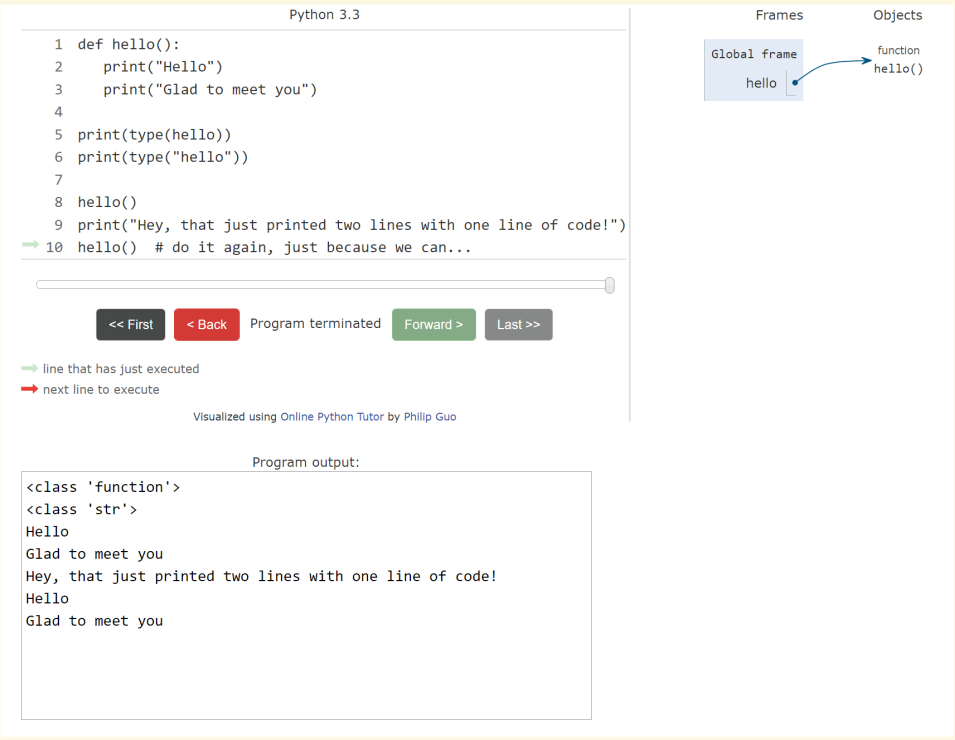
12.3. Function Invocation

Defining a new function does not make the function run. To execute the function, we need a **function call**. This is also known as a **function invocation**.

Note

This section is a review of something we learned in the beginning of the textbook.

The way to invoke a function is to refer to it by name, followed by parentheses. Since there are no parameters for the function hello, we won't need to put anything inside the parentheses when we call it. Once we've defined a function, we can call it as often as we like and its statements will be executed each time we call it.



Activity: 1 -- CodeLens: (clens11_2_1)

Let's take a closer look at what happens when you define a function and when you execute the function. Try stepping through the code above.

First, note that in Step 1, when it executes line 1, it does *not* execute lines 2 and 3. Instead, as you can see in blue "Global variables" area, it creates a variable named hello whose value is a python function object. In the diagram that object is labeled hello() with a notation above it that it is a function.

At Step 2, the next line of code to execute is line 5. Just to emphasize that hello is a variable like any other, and that functions are python objects like any other, just of a particular type, line 5 prints out the type of the object referred to by the variable hello. It's type is officially 'function'.

Line 6 is just there to remind you of the difference between referring to the variable name (function name) hello and referring to the string "hello".

At Step 4 we get to line 8, which has an invocation of the function. The way function invocation works is that the code block inside the function definition is executed in the usual way, but at the end, execution jumps to the point after the function invocation.

You can see that by following the next few steps. At Step 5, the red arrow has moved to line 2, which will execute next. We say that *control has passed* from the top-level program to the function hello. After Steps 5 and 6 print out two lines, at Step 7, control will be passed back to the point after where the invocation was started. At Step 8, that has happened.

The same process of invocation occurs again on line 10, with lines 2 and 3 getting executed a second time.

Common Mistake with Functions

It is a common mistake for beginners to forget their parenthesis after the function name. This is particularly common in the case where there parameters are not required. Because the hello function defined above does not require parameters, it's easy to forget the parenthesis. This is less common, but still possible, when trying to call functions that require parameters.

Check your understanding

func-2-1: What is a function in Python?

- (•) A. A named sequence of statements.
- ( ) B. Any sequence of statements.
- ( ) C. A mathematical expression that calculates a value.
- ( ) D. A statement of the form x = 5 + 4.

Check me Compare me

✔ Yes, a function is a named sequence of statements.

Activity: 2 -- Multiple Choice (question11_2_1)

func-2-2: What is one main purpose of a function?

- A. To improve the speed of execution
- B. To help the programmer organize programs into chunks that match how they think about the solution to the problem.
- C. All Python programs must be written using functions
- D. To calculate values.

Check me | Compare me

✔ While functions are not required, they help the programmer better think about the solution by organizing pieces of the solution into logical chunks that can be reused.

Activity: 3 -- Multiple Choice (question11_2_2)

func-2-3: How many lines will be output by executing this code?

```
def hello():
   print("Hello")
   print("Glad to meet you")
```

- A. 0
- B. 1
- C. 2

Check me | Compare me

✔ The code only defines the function. Nothing prints until the function is called.

Activity: 4 -- Multiple Choice (question11_2_3)

func-2-4: How many lines will be output by executing this code?

```
def hello():
   print("Hello")
   print("Glad to meet you")

hello()
print("It works")
hello()
hello()
```

- A. 0
- B. 1
- C. 3
- D. 4
- E. 7

Check me | Compare me

✔ Three invocations generate two lines each, plus the line "It works".

Activity: 5 -- Multiple Choice (question11_2_4)

You have attempted 6 of 5 activities on this page

12.2. Function Definition">
nction Definition">
12.4. Function Parameters">
✔ Completed. Well Done!
12.4. Function Parameters">Next Section - 12.4. Function Parameters

| Back to top