Functions

Exercises

Week 4

Prior to attempting these exercises ensure you have read the lecture notes and/or viewed the video, and followed the practical. You may wish to use the Python interpreter in interactive mode to help work out the solutions to some of the questions.

Download and store this document within your own filespace, so the contents can be edited. You will be able to refer to it during the test in Week 6.

Enter your answers directly into the highlighted boxes.

For more information about the module delivery, assessment and feedback please refer to the module within the MyBeckett portal.

©2021 Mark Dixon / Tony Jenkins

What must be done before a function that is not *built-in* to Python can be used in a program?

Answer:

Functions which are not built-in must be imported before they can be used within our programs. For example: ‘form math import sqrt’ to access ‘sqrt’ function.

Given the following `import` statement, how would a call to the `sin()` function be made?

```
import math
```

Answer:

```
import math

result = math.sin(30) # Calling the sin() function

print("Sine of 30 degrees:", result)
```

Given the following `import` statement, how would a call to the `sqrt()` function be made?

```
from math import sqrt
```

Answer:

```
from math import sqrt

result = sqrt(25) # Calling sqrt() function

print("Square root of 25:", result)
```

What is the name of the common library that is available with all Python distributions?

Answer:

The common library that is available with all Python distributions is the **Python Standard Library**.

What keyword is used in Python to define a new function?

Answer:

def keyword is used. For example: def my_function():

Write some Python code that defines a function called `print_header(msg)`. This should output the value provided by the '`msg`' parameter to the screen (prefixed by five asterisk '`*****`') characters.

Answer:

```
def print_header(msg):
    print("*****", msg)

message_to_print = "Mr Wick is here!"
print_header(message_to_print)
```

In the answer box below give an example of what the **docstring** may look like for the `print_header(msg)` function.

Answer:

```
"""Prints a message prefixed by five asterisks"""
```

Where within a function definition should a **docstring** appear?

Answer:

docstring should appear as the first statement after the function header (the line with def).

What statement should appear within a function's code block to cause a specific value to be passed back to the caller of the function?

Answer:

The **'return'** statement is used within a function's code block to specify a value that should be passed back to the caller of the function.

Write some Python code that defines a function called `find_min(a,b)` that returns the smallest of the two given parameter values.

Answer:

```
def find_min(a, b):
    """Returns the smallest of two given parameters."""
    return min(a, b)

result = find_min(8.356, 8.35)
print("The minimum value is:", result)
```

Given the following function definition, which of the *formal parameters* could be described as being a **default argument**?

```
def shouldContinue(prompt, answer=False):
      # function body...
```

Answer:

answer = false because in this case, the default value for the 'answer' parameter is 'False'.

Provide two example calls to the above function, one which provides a value for the *default argument*, and one that does not.

Answer:

```
def shouldContinue(prompt, answer=False):
   print("Reporter:", prompt)
   print("Maid:", answer)

shouldContinue("Is he Mr. John Wick?", True)
shouldContinue("Is he a weapon dealer?")
```

State why following function definition would **not** be allowed.

```
def do_something(prefix="Message", prompt, answer=False):
      # function body...
```

Answer:

Default arguments can only be specified to the right of parameters that do not have default value. The parameter **prefix** has a default value, but the parameter **prompt** comes after it and does not have a default value. This would result in a **SyntaxError.**

What single character is placed directly before the name of a *formal parameter*, to indicate that a variable number of actual parameters can be passed when the function is called?

Answer:

Asterisk (*). For example: def my_function(arg1, *args):

What commonly used built-in function, which displays output on the screen, can take a **variable number** of arguments?

Answer:

print()

Is it valid for a function's parameter name to be prefixed by two asterisk characters '`**`' as shown below?

```
def send_output(**details):
      # function body...
```

Answer:

Yes, it is valid for a function's parameter name to be prefixed by two asterisk characters **.

If present, what does this prefix indicate?

Answer:

** prefix syntax indicates that the **details function can accept any number of arguments and those arguments are bundled into a dictionary.

What is the name given to a small 'anonymous' function that must be defined using a single expression?

Answer:

lambda function

Give an example of such a function that calculates the *cube* of a given number (i.e. the value of the number raised to the power of three) -

Answer:

```
cube = lambda x: x ** 3

result = cube(4)
print("Cube of 4:", result)
```

Exercises are complete

Save this logbook with your answers. Then ask your tutor to check your responses to each question.