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.

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

Answer:

Before a function that is not built-in to Python can be used in a program, it must be defined or imported.

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

```
import math
```

Answer:

To call the sin() function,
math.sin(x)

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

```
from math import sqrt
```

Answer:

To call the sqrt(),
sqrt(x)

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

Answer:

Python Standard Library

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

Answer:

def keyword is used in Python to define a new 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(f“*****”, msg)
```

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

Answer:

```
def print_header(msg):
    print ("=" * len(msg))
    print (msg)
    print ("=" * len(msg))
```

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

Answer:

A docstring should appear immediately after the function definition and also should be the first statement inside the function body.
For example;

```
def function_name (parameters):
    """
    This is where the docstring goes.
    It should describe what the function does, its parameters, and return value.
    """
    # Function code here
```

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:

Return statement should appear within a function’s code block to cause a specific value to 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):
   return a if a < b else b

# Example usage of the function:
print(find_min(5, 8))
 # Output: 5
```

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 could be described as being a default argument.

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

Answer:

Two example calls to the shouldContinue function:
1. Call with a value for the default argument (answer=True):
   shouldContinue("Do you want to continue?", answer=True)
2. Call without providing a value for the default argument (uses answer=False)
   shouldContinue("Do you want to continue?")

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

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

Answer:

The default arguments must appear after non-default arguments, so the given function would not be allowed.

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:

The 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 is asterisk(*).

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

Answer:

The commonly used built-in function that displays output on the screen and can take a variable number of arguments is 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:

The ** prefix in a function parameter indicates that the function can accept arbitrary keyword arguments.

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

Answer:

The name given to a small 'anonymous' function that must be defined using a single expression is a 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
print(cube(3))
```

Exercises are complete

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