

# 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:

```
We need to define a function  
def function_name:
```

---

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

```
import math
```

Answer:

```
math.sin()
```

---

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

```
from math import sqrt
```

Answer:

```
From math import sqrt  
x=4  
Result=sqrt(x)  
print(result)
```

---

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
```

---

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)
print_header("hello")
```

---

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

*Answer:*

```
Def function:
    """Here the we print five asterisks and the msg together"""
Docstring basically explains the purpose of a function
```

---

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

*Answer:*

```
Just below where function is defined
```

---

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
def get_length(string):
    """
    Get the length of a string.

    This function takes a string as an argument and returns its length.

    :param string: The string to get the length of.
    :type string: str
    :return: The length of the string.
    :rtype: int
    """
    return len(string)

string_length = get_length("Hello, world!")
print("The length of the string is:", string_length)
```

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 min(a,b):
    if a>b:
        print("the minimum value is",b)
    else:
        print("the minimum value is",a)
min(4,5)
```

---

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:

Default argument is the answer because it needs to be defined

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

Answer:

shouldContinue("Do you want to continue?",True) provides a value for default argument  
shouldContinue("Do you want to continue") does not.

---

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

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

Answer:

Because the default argument prefix is defined before the non default argument prompt

---

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:

\* is placed

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, they are keyword arguments

---

If present, what does this prefix indicate?

Answer:

It indicates that the function accepts a variable number of keyword arguments.

---

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:

```
Def cubert(x):  
    Return x**3  
result=cubert(3)  
print(result)
```

---

**Exercises are complete**

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