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:

Import

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:

sqrt()

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

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)
```

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):
    """
    Prints the provided message with five asterisk characters (*****)
    as a prefix.

    Parameters:
    msg (str): The message to be printed.

    Example:
    print_header("Welcome to Python!")  # Output: ***** Welcome to Python!
    """
    print("*****", msg)
```

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

Answer:

It should immediately after the function header .

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

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=int(input("Enter a number:"))
    b=int(input("Ennter another number:"))
    print(a if a>b else b)
find_min()
```

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

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

Answer:

print(shouldContinue("Do you want to continue?"))
print(shouldContinue("Do you want to continue?", True))

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

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

Answer:

Default arguments must come after non default arguments.

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:

Single asterisk(*)

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

If present, what does this prefix indicate?

Answer:

It indicate that the function can accept an arbitrary number of keyword arguments which are passed as 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
Print(cube)
```

Exercises are complete

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