

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: Define the function before a function that is not *built-in* to Python can be used in a program.

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

import math

Answer:

import math

x = 0.5

result = math.sin(x)

print(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

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

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("*****", msg, "*****")  
  
print_header("Hello, world!")
```

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 a message to the screen, prefixed by five asterisks.  
  
    Parameters:  
  
    msg (str): The message to be printed.
```

Returns:

None

"""

*print("*****", msg, "*****")*

print_header("welcome")

Where within a function definition should a docstring appear?

Answer: A **docstring** within a function definition should appear after the function definition and before the start of the function body.

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):
```

```
    if a < b:
```

```
        return a
```

```
    else:
```

```
        return b
```

```
x = 3
```

```
y = 5
```

```
result = find_min(x, y)
```

```
print(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* 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:

One which provides a value for the *default argument* ;

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

and one that does not ;

```
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: Function definition would **not** be allowed 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: *Asterisk(*)* 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.

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

Answer: `print()` is commonly used built-in function, which displays output on the screen, can take a **variable number** of arguments.

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

If present, what does this prefix indicate?

Answer: This prefix indicate 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 : The name given to a small 'anonymous' function that must be defined using a single expression is 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 cube(x):  
    return x ** 3  
  
result = cube(3)  
  
print(result)
```

Exercises are completed