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:*

|  |
| --- |
| Firstly define the function, call the function and import the function or module. |

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

import math *Answer:*

|  |
| --- |
| import math  result = math.sin(0.45)  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  result = sqrt(25)  print(result) |

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

*Answer:*

|  |
| --- |
| Common library are :- math, Boolean, etc |

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

*Answer:*

|  |
| --- |
| ‘Def’ is the key word to define a function in python. |

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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 = "Hello, this is a header"  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:*

|  |
| --- |
| def print\_header(msg):  """  Prints a header  with the  provided message.  """  print("\*\*\*\*\*", msg, "\*\*\*\*\*") |

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

*Answer:*

|  |
| --- |
| Docstring appear as the first statement in a function or module in python. |

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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 function is used to code block. |

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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 values.  """  return min(a, b)  value1 = 8  value2 = 4  min\_value = find\_min(value1, value2)  print(f"The smallest value between {value1} and {value2} is: {min\_value}") |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** 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 is 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:*

|  |
| --- |
| Default argument is prompt that provide value and answer does not need any value. |

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

def do\_something(prefix="Message", prompt, answer=False): # function body...

*Answer:*

|  |
| --- |
| Default function defines that, the ‘prompt’ parameter comes before the ‘prefix’ parameter, |

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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 (“\*”) |

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

*Answer:*

|  |
| --- |
| Print() function display out put on the screen. |

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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:*

|  |
| --- |
| \*\* is the prefix |

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

*Answer:*

|  |
| --- |
| Lambda parameters is the name given to a small ‘anonymous’ function that must be defined using a single expression. |

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(9)  print(result) |

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Exercises are complete**

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