Quiz on Functions and Procedures ( / 20 )

( / 3 )

|  |  |
| --- | --- |
| **List 3 functions that are built into Python** | **List 3 procedures that are built into Python** |
| len() | print() |
| range() | L.append() |
| int() | L.remove() |

Below is the definition of a new command called **myNewCommand,** followed by a call of it. ( / 3 )

|  |
| --- |
| **def** **myNewCommand**( num1, num2, num3 ):  #does some stuff  #does some more stuff  myNewCommand(4, 7, -2) |
| Is **myNewCommand** a procedure or a function?  Procedure |
| Explain how you know. (Spelling and grammar count.)  I know it is a procedure because “myNewCommand” doesn’t have a return statement and therefore doesn’t return a value to the user, making it a procedure. |

In the definition of **myNewCommand**, *num1*,*num2* and *num3,* are the ( / 1 )

|  |
| --- |
| parameters |

In the call **myNewCommand**( 4, 7, -2 ), the **4**, **7** and **-2** are the ( / 1 )

|  |
| --- |
| arguments |

In the box below, write a Python function called **getVolumeOfCone**(r, h)that returns the volume of a cone that has radius r and height h. If you wish, you may first write the function in IDLE, then paste it into the box. You may look up the formula for the volume of a cone online if you’ve forgotten. ( / 6 )

|  |  |
| --- | --- |
| def getVolumeOfCone(r, h):  return (pi\*r\*\*2) \*(h/3) | Note:  Don’t worry about colors. Just type. Use the space bar to show indentation. |

Write some Python code that uses your function getVolumeOfCone to find the volume of a cone that has radius 6 and height 8, and then prints out the answer. ( / 2 )

|  |
| --- |
| Vol = getVolumeOfCone(6, 8)  print(“The volume of the cone with radius 6 and height 8 is: “ +Vol) |

In the box below, write a Python procedure called **drawSquare**( xCentre, yCentre, size, col )   
that draws a square on the screen, where (xCentre, yCentre) is the centre of the square, size is the square’s side length, and col is the square’s colour. ( / 4 )

|  |
| --- |
| tk = Tk()  screen = Canvas(tk, width = 800, height = 800, bg = “white”)  screen.pack()  def drawSquare(xCentre, yCentre, size, col):  screen.create\_rectangle(xCentre-size, yCenter-size, xCenter+size, yCenter-size, fill = col, outline = col) |

To make sure it’s working, you may code it in IDLE and then try calling **drawSquare**(200, 600, 100, “blue”). This should draw a blue square that’s centered at (200, 600) and has side length 100.