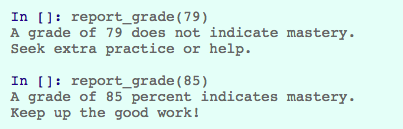
**APCSP Activity 1.3.3: Python Variables and Functions**

Learning Target: Define the problem and analyze research to create a solution to a problem.

**Step 6**

1. Copy the age\_limit\_output function code into the code editor. Execute the code, and then try the following input. Describe the output.Screen Shot 2017-10-18 at 8.40.05 AM.png

|  |
| --- |
| *# both of these statements analyzes the input and compares it to*  *# 13 and if it’s lower than 13 it will produce the first output*  *# and if it’s equal to or greater than 13 it will produce*  *# the second output*  Out []: 10 is below the age limit. Out []: Minimum age is 13  Out []: 16 is old enough. Out []: Minimum age is 13 |

1. Define a new function report\_grade(percent) that reports mastery if the argument percent is 80 or more. The beginning of the new code and the required output are both shown below.

Screen Shot 2017-10-18 at 8.42.01 AM.png

|  |
| --- |
| def report\_grade(percent):  if (type(percent) is int):  if (percent <= 100 and percent >= 0):  if (percent <= 80):  print('A grade of ' + str(percent) + ' does not indicate mastery.')  print('Seek extra practice or help.')  else:  print('A grade of ' + str(percent) + ' percent indicates mastery.')  print('Keep up the good work!')  else:  print('Enter a number from 0-100')  else:  print('Enter a number') |

**Step 8**

Define a function letter\_in\_word(guess, word) that returns True if guess is a letter in word and returns False otherwise.

|  |
| --- |
| def letter\_in\_word(guess, word):  if (type(guess) is str and type(word) is str):  if (guess in word):  return True  else:  return False  else:  print('both the guess and the word has to be a string') |

**Step 9**

Define a function hint(color, secret) that takes two parameters: a string (representing a color) and a list of strings (representing a sequence of colors). The function should print a hint telling whether the color is in string.

|  |
| --- |
| def hint(color, secret):  if (type(secret) is list):  if (type(color) is str):  if color in secret:  print('The color ' + color + ' IS in the secret sequence of colors.')  else:  print('The color' + color + ' IS NOT in the secret sequence of colors.')  else:  print('The color has to be a string')  else:  print('The secret has to be an array') |

**Conclusion Questions**

1. Describe the relationship between blocks of code indented after the colon in if, elif, and else blocks.

|  |
| --- |
| The indentations of functions and logical statements can be interpreted as boxes within boxes. For every layer of indention a new box would be added. So the base function would be a big box that contained a bunch of smaller boxes which would be all the if, elif, and else statements. And if you have multiple logical statements embed into each other it would be like having multiple boxes nested within each other. |

2. There are many operators that can be used to create Boolean expressions. List the ones you have learned about and name one more than you learn about by searching for Boolean operators on the Internet.

Any logical statement that only has a yes or no answer returns a boolean. When you have an if statement like so:

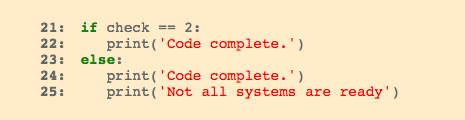
|  |
| --- |
| if (x > 3):  If (y == 1 and z => 30): |

The if statement doesn’t individually check each logical operation. An if statement only uses boolean values so if True do something and if False don’t do something. Same thing goes with multiple operations, like if y = 1 and z is = or greater than 30 it returns True but only if each individual operations are also True. You can even declare variables using logic like so:

|  |
| --- |
| x = 1 y = 10 z = (x == 1 and y == 10) |

z would be defined as True because x is = to 1 AND y is = to 10. If that operation was False z would be defined as False.

3. Steve and Latisha wrote this code:



Ira, Jayla, and Kendra are all saying it would be better to move lines 22 and 24 to a single line executing print (‘Code complete.’) just before line 21. These three students have different reasons for their opinions. Their reasons are below. Do you think each of them is right, wrong, or somewhere in between? Explain.

Ira: “It would be better to have a single print statement because that code is going to happen no matter what. The program will run slower by having it there twice.”

Jayla: “It would be better to have a single print statement because that code is going to happen no matter what. Later, if you want to change your program, you're going to have to remember to change it in two places the way the code is now.”

Kendra: “It would be better to have a single print statement because it is going to happen no matter what. That program would take up less memory if you just wrote it once.”

|  |
| --- |
| Jayla is correct because the code will happen no matter what and her reason didn’t involve the usage of processing power because it wouldn’t use more processing power because the statement is there twice. Both of the print statements are in different metaphorical boxes(explained in question 1) so they won’t ever both execute in the same run through of the code. Instead the current situation is just less ideal for the developer because if they wanted to change the print string they would have to change both of them and they might forget that there’s two of them. |