**APCSP Activity 1.3.7: For Loops**

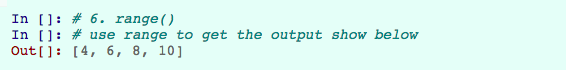
Learning Target: Define the problem and analyze research to create a solution to a problem.

**Step 6**

Examine the code below.



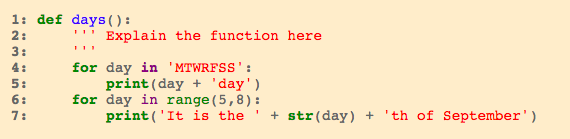
Write code with range() that will return the list shown here:



|  |
| --- |
| range(4, 12, 2) |

**Step 7**

Explain the code below:



Screen Shot 2017-10-25 at 2.41.55 PM.png

The first for loop runs through line 5 for every character in the specified string which in this case is 7, one for each day of the week. So it would concatenate the current character and the ‘day’ string and then print it for every day of the week. The next for loop runs through the loop 3 times because the difference between 5 and 8 is 3 and then it would concatenate and print a string using the current number in the loop.

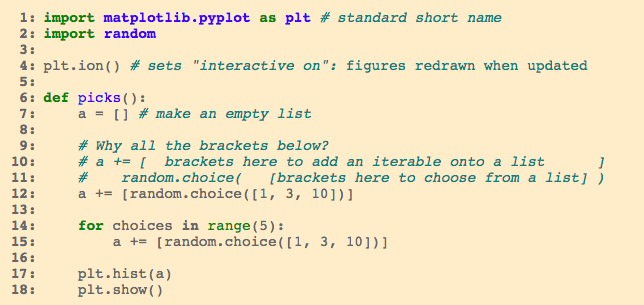
**Step 8**

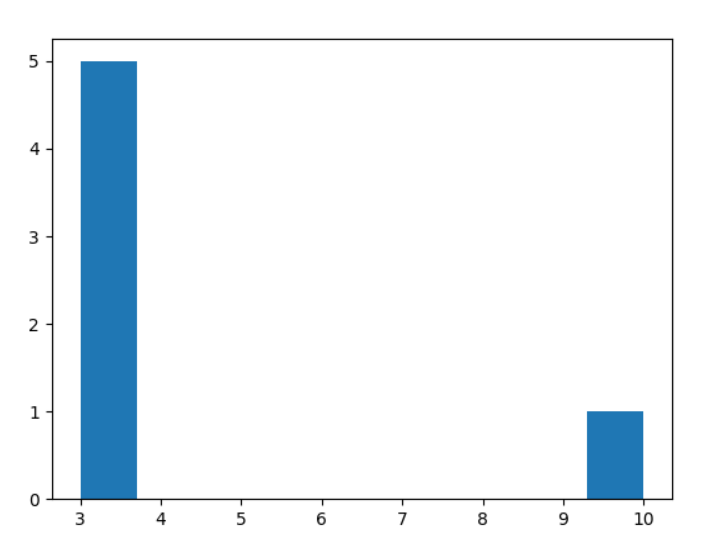
List five built-in *Python* functions. Explain the purpose of each function.

|  |  |
| --- | --- |
| **Function Name** | Purpose |
| print(str) | To print the specified string to the command line |
| eval(str) | To run a string as if it was standard python code |
| hex(int) | To convert an integer to a hexadecimal |
| raw\_input(str) | To obtain user input in the form of a string |
| range(int, int, int) | To Return an array following a specific pattern from start to stop |

**Step 9**

Execute the code below. Provide a screenshot of the output.





**Step 10: Complete at least one of the following:**

1. Define a function roll\_hundred\_pair() that produces a histogram of the results of 100 rolls of two 6-sided dice.

|  |
| --- |
| def roll\_hundred\_pair():  dice = []  for i in range(100):  dice.append(random.randint(1, 6))  plt.hist(dice)  plt.show() |

1. Define a function dice(n) that returns the sum of a random roll of n 6-sided dice. Example output shown here:

Screen Shot 2017-10-25 at 2.49.05 PM.png

|  |
| --- |
| def dice(n):  dice = []  total = 0  for r in range(n):  dice.append(random.randint(1, 6))  for i in range(len(dice)):  total += dice[i]  print('Roll was ' + total) |

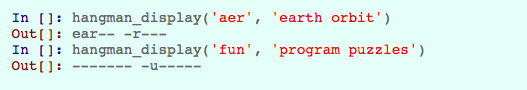
**Step 11: Complete at least one of the following:**

1. Define a function hangman\_display(guessed, secret) that returns the string a hangman player would see. The arguments are:

guessed: letters guessed so far

secret: the full secret word or phrase

*Hint*: Start with the null string and add onto it one character at a time.



|  |
| --- |
| def hangman\_display(guessed, secret):  output = ''  for s in range(len(secret)):  character = ''  if (secret[s] == ' '):  character = ' '  else:  for g in range(len(guessed)):  if (secret[s] == guessed[g]):  character = secret[s]  if (character == ''):  character = '-'  output += character  return output |

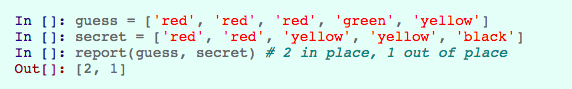
1. A lottery ticket contains five *unique* numbers. (A set of unique numbers does not contain repeated elements.) The winning combination of this lottery is chosen by picking five unique numbers. Define a function matches(ticket, winners) that takes two lists and returns an integer that says how many numbers the two lfor ists have in common.

Screen Shot 2017-10-25 at 2.54.05 PM.png

|  |
| --- |
| def matches(ticket, winners):  matches = 0  if (type(ticket) == list) and (type(winners) == list):  if (len(ticket) == len(winners)):  for t in range(len(ticket)):  for w in range(len(winners)):  if (ticket[t] == winners[w]):  matches += 1  if (matches == len(ticket)):  print('You just won the lottery!')  else:  print('There\'s ' + str(matches) + ' number(s) in common') |

1. In MasterMind, one player has a secret code made from a sequence of colored pegs. Another player tries to guess the sequence. The player with the secret reports how many colors in the guess are in the secret and also reports whether the correct colors are in the correct place.

Write a function report(guess, secret) that takes two lists and returns a 2-element list [number\_right\_ place, number\_wrong\_place].



|  |
| --- |
| def report(guess, secret):  inPlace = 0  colors = []  if (type(guess) == list) and (type(secret) == list):  if (len(guess) == len(secret)):  for i in range(len(guess)):  if (guess[i] == secret[i]):  inPlace += 1  for s in range(len(secret)):  for g in range(len(guess)):  if (guess[g] == secret[s]):  colors.append(guess[g])  colors = set(colors)  return (str(inPlace) + ' values in the same place and ' +  str(len(colors)) + ' colors in common') |

**Conclusion Questions**

1. Sometimes code using an iterative loop can be written without a loop, simply repeating the iterated code over and over as separate lines in the program. Explain the disadvantages of developing a program this way.

It would be extremely tedious and if you wanted to add any more values to iterate through you would just have to add more lines of code. If you were using a for loop to iterate through an array, the entire process would be completely automatic so all you would have to do is add values into the array instead of manually hard coding each value yourself.

2. Name a large collection across which you might iterate.

Pretty much any dynamic array that has the possibility of values being added onto it, like for example an array of users emails on a website would be used in array iteration when emailing those emails in bulk. And as long as the email is in the array it will always be sent the specified email because the program iterates through the entire array.

3. What is the relationship between iteration and the analysis of a large set of data?

Lots of programs analyze and compare data during array iteration. If you have a database full of large amounts of data and you want to for example query all of the strings in the database that has contains the substring of ‘Thomas Achatz’, you would iterate through every value in the database and then for every string that is encountered it would then parse the string in search for the specified substring. It would then output an array that includes all the strings in the database that include ‘Thomas Achatz’.