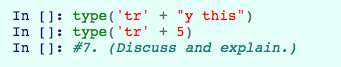
**APCSP Activity 1.3.5: Strings**

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

**Step 7**

One of the following two inputs will produce an error. Try this, discuss both outputs with your partner, and summarize your discussion.

The first type check won’t return an error because ‘tr’ and “y this” are both strings regardless of what type of quotes they have. The second type check however will return an error because ‘tr’ and 5 are not both strings so they can’t concatenate.

**Step 8**

Identify the individual character referenced by the code below.

|  |
| --- |
| slogan = 'My school is the best' |

|  |  |
| --- | --- |
| **Code** | **Reference Character** |
| slogan[0] | M |
| slogan[2] | SPACE |
| slogan[8] | l |
| slogan[26] | NULL / UNDEFINED |
| slogan[-2] | NULL / UNDEFINED |

**Step 9**

Slice the code for the variable ‘slogan’ to return ‘best’, omitting the end index.

|  |
| --- |
| slogan = 'My school is the best' return slogan[17:21] |

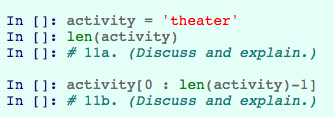
**Step 10**

Using slicing and concatenation to create your own sentence, pasting together two strings.

|  |
| --- |
| str1 = 'the programming language JavaScript is far more superior than Python' str2 = 'JavaScript is much better than Python'  jsString = str1[25:35] pyString = str1[62:68]  str3 = str2[10:31]  return pyString + str3 + jsString |

**Step 11**

Explain the output of the following inputs:

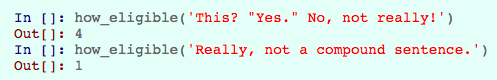


The len function just returns the length of the input string so in this case len(activity) would return 5 because the string ‘theater’ has 6 characters and python starts counting at 0.

The next line of code which is activity[0 : len(activity)-1] is just a string splice which starts at position 0 which is the first character and ends at the length of the string - 1 so it would just remove the last character in the string.

**Step 13**

A social media site offers a contest to write a humorous short paragraph. A constraint on the creative format: the entry must include a question, a quote, a compound sentence, and an exclamation. These would contain the characters ?, ", ,, and !, respectively.

Create a function how\_eligible(essay) that returns 0 to 4, equal to the number of these four characters that the essay included. As pair programmers, generate ideas for how to solve this problem, strategize, and then code and test iteratively.

|  |
| --- |
| def how\_eligible(essay):  score = 0  if (type(essay) is str):  if ('"' in essay):  score = score + 1  if ('?' in essay):  score = score + 1  if ('!' in essay):  score = score + 1  if (',' in essay):  score = score + 1  return score |