Learning Journal Unit 5

Godfrey Ouma

University of the People

CS 1101: Programming Fundamentals

Janice Block

July 20, 2023

Learning Journal Unit 5

Question 1: Modifying code to get desired result

Initial code

```
In [44]: prefix="JKLMNOPQ"
    ...: suffix="ack"
    ...:
    ...: for letter in prefix:
    ...:     print(letter+suffix)
    ...: #output
Jack
Kack
Lack
Mack
Nack
Oack
Pack
Qack
```

Modified code

```
In [51]: prefixes="JKLMNOPQ"
    ...: suffix="ack"
    ...:
    ...: for letter in prefixes:
    ...:     if letter=="O" or letter=="Q":
    ...:         print(letter+"u"+suffix)
    ...:     else:
    ...:         print(letter+suffix)
    ...: #output
Jack
Kack
Lack
Mack
Nack
Ouack
Pack
Quack

In [52]:
```

Explanation:

The initial code prints the names "Oack" and "Qack", which is not the desired result. The code is further modified by adding an “if-statement” to check if the current letter is “O” or

“Q” first before adding the suffix “ack”. The string “u” before the suffix to form "Ouack" and "Quack" if the current letter is “O” or “Q” is found in the prefixes “JKLMNOPQ”. However, the remaining letters in prefixes added to the original suffix “ack” then printed in the output to keep the other names the same. In the end, the desired result is achieved by printing "Ouack" and "Quack" together with other original names.

Question 2: Features of string slices

A slice is a section of a string. Selecting a slice is like choosing a character. The segment of the string from the "n-eth" character to the "m-eth" character, including the first but excluding the last, is often returned by the operator [n:m] used to call the index. String slices have different features as follows.

Feature 1:

Omitting the first index (before the colon) results in the slice starting at the beginning of the string (Downey, 2015). See example below:

```
In [54]: name="Godfrey"
    ...: #omit the first index (before the colon)
    ...: char=name[:3]
    ...: print(char)
    ...: #output
God
```

Explanation:

In the example above, the slice name[:3] starts at the beginning of the string text and goes up to index 3 (exclusive). Thereafter, it prints all characters from index 0 to index 2, leading to the output "God". Therefore, omitting the first index in a slice makes it default to 0, which means the slice starts at the beginning of the string.

Feature 2:

Omitting the first index (after the colon) results in the slice going up to the last character of the string (Downey, 2015). See example below:

```
In [56]: name="Godfrey"
    ...: #omit the second index(after the colon)
    ...: char=name[3:]
    ...: print(char)
    ...: #output
frey
```

Explanation:

In the example above, the slice name[3:] starts at index 3 (inclusive) and goes up to the end of the string, including all characters from index 3 to the last character. The output will be "frey". Therefore, omitting the second index in a slice makes it defaults to the end of the string, which means the slice goes up to the last character of the string.

Feature 3:

An empty string is printed in the output if the first index is greater than or equal to the second (Downey, 2015). See example below:

```
In [57]: name="Godfrey"
    ...: #If the first index is greater than or
equal to the second
    ...: char=name[3:3]
    ...: print(char)
    ...: #output


In [58]: name="Godfrey"
    ...: #If the first index is greater than or
equal to the second
    ...: char=name[3:2]
    ...: print(char)
    ...: #output
```

Explanation:

In the example above, the slice name[3:3] is taken from the string, where the first index is 3 (inclusive) and the second index is also 3 (exclusive). The first index, 3, represents the fourth character of the string (since indexing starts from 0). The second index, also 3, means the slice will stop right before the fourth character. Since the slice range [3:3] does not contain any characters, the result will be an empty string.

On the other hand, the slice name[3:2] is taken from the string, where the first index is 3 (inclusive) and the second index is 2 (exclusive). The first index, 3, represents the fourth character of the string "f" (since indexing starts from 0). The second index, 2, means the slice will stop right before the third character. Since the first index is greater than or equal to the second index (3 >= 2), the result will be an empty string because the slice cannot progress from the start index to the end index when the start index is greater than or equal to the end index.

Therefore, when the first index is greater than or equal to the second index in a slice, the result is an empty string, which contains no characters and has length 0, but other than that, it is the same as any other string.

Feature 4:

A slice with empty indices includes all characters (Downey, 2015) as shown in the example below.

```
In [60]: name="Godfrey"
    ...: #The first and the second index is empty
    ...: char=name[:]
    ...: print(char)
    ...: #output
Godfrey
```

Explanation:

In the example above, the slice name[:] is taken from the string, where the first index is not specified (empty) and the second index is also not specified (empty). This form of slice with empty indices is known as the full slice or the whole slice. When an empty first index and an empty second index used in a slice, it means all characters will be included in the string. The slice will start from the beginning of the string and go up to the end of the string, effectively creating a copy of the entire string. So, name[:] is equivalent to name[0:len(name)], where 0 is the default start index, and len(name) is the default end index (the length of the string). Therefore, a slice with empty indices includes all characters of the string "Godfrey" from the first character at index 0 to the last character at index len(name)-1, which is 7 in this case. As a result, the entire string "Godfrey" is copied and assigned to the variable char.

Reference

Downey, A. (2015). *Think Python: How to think like a computer scientist.* Green Tree Press.