More Lists and More Functions Practice

GWC SIP 2019

28 June 2019

1 Lists

1.1 Turn the string into a list of characters.

(i.e. fill in the provided blanks with the appropriate characters)

```
st = "friday"  --> ["f"|"r"|"i"|"d"|"a"|"y"]
```

1.2 Write the appropriate outputs on the right.

```
len(st)                 # OUTPUT: 6
st[0]                   # OUTPUT: "f"
st[3]                   # OUTPUT: "d"
st.index("r")           # OUTPUT: 1
```

1.3 Write the appropriate output on the right.

```
st = "friday"            # OUTPUT:
for c in st:             # f
    print(c)             # r
    #print(c, end="")    # i
                         # d
                         # a
                         # y
                         #
                         #
```

1.4 Step through the loop and mark if that word is printed.

```
for num in range(5):          #  num  |  blue  |  orange  |  red
    if (num % 2 == 0):        # ----------------------------------
        print("blue")         #   0       x
    elif (num % 3 == 0):      #   1                          x
        print("orange")       #   2       x
    else:                     #   3                  x
        print("red")          #   4       x
```

1.5 range(3) corresponds to which values?

```
    (a) [1, 2, 3]
    #(b) [0, 1, 2]
    (c) [1, 2]
    (d) [0, 1, 2, 3]
```

1.6 Let s = "happy". range(len(s)) corresponds to which values?

```
    (a) [0, 1, 2, 3]
    (b) [1, 2, 3, 4, 5, 6]
    (c) [1, 2, 3, 4, 5]
    #(d) [0, 1, 2, 3, 4]
```

2 Functions

2.1 Let s = "happy".

2.1.1 Draw the string s as a sequence of characters.

```
s = "happy"  --> ["h"|"a"|"p"|"p"|"y"]
```

2.1.2 Fill in the blanks in the below code to ignore all instances of the letter "p" and build a new string.

```
def ignore_letter(ch, word):
    myword = ""      # empty string "" --> [""]
```

```
    for letter in word:
        if ( letter != ch ):
            myword += letter
    return myword

def main():
    s = "happy"
    new_word = ignore_letter ("p", s)
    print("new word: " , new_word)
```

2.2 Let mylist = [14, 3, 7, 26]. Fill in the blanks below to sum up all of the values in mylist.

```
def sum_list(alist):
    total = 0
    for value in alist:
        total += value
    return total
```

2.3 Let mylist = [14, 3, 7, 26].

2.3.1 Fill in the blanks below to build a new list containing the indices for all of the even values in mylist.

```
def only_evens(alist):
    # local scope --> alist # mylist doesn't exist here
    new_list = []
    for index in range( len( alist ) ):
        if ( alist [ index ] % 2 == 0 ):
            new_list.append( index )
    return new_list

def main():
    mylist = [14, 3, 7, 26]
    evens = only_evens(mylist)
```

2.3.2 What is the final value of evens?

```
def main():
    mylist = [14, 3, 7, 26]
    evens = only_evens(mylist)
...
evens = [14, 26]
or
evens = [0, 3]

# append examples
l = [1, 2, 3]

l.append(4)
# now l = [1, 2, 3, 4]
```