LISTS

A list is a collection of items in a particular order. You can make a list that includes the letters of the alphabet, number, the names of all the people in your family, etc. You can put anything you want into a list and the items in your list don’t have to be related in any particular way.

In Python, square brackets [ ] indicate a list and the individual elements in the list are separated by commas.

Example

```
wizard_list = [‘spider legs’, ‘toe of frog’, ‘snail tongue’, ‘bat wing’,
‘slug butter’, ‘snake dandruff’,  ‘bear burp’]

>>> print(wizard_list)

['spider legs', 'toe of frog', 'snail tongue', 'bat wing', 'slug butter',
‘snake dandruff’, 'bear burp']
```

```
Lists start at index position 0, so the first item in a list is 0, the second
is 1 , the third is 2 and so on.
```

```
>>> print(wizard_list[0]

spider legs

>>> print(wizard_list[0].title())

Spider legs
```

How to output all the elements of the list, one at a time?

Method 1 (not so efficient)

```
>>> print(wizard_list[0])

spider legs

>>> print(wizard_list[1])

toe of frog

>>> print(wizard_list[2])

snail tongue

>>> print(wizard_list[3])

bat wing

>>> print(wizard_list[4])

slug butter
```

```
>>> print (wizard_list[5])
snake dandruff
>>> print (wizard_list[6])
bear burp
```

Method 2 (using a loop – better!)

```
for i in range(0, 7):
      print(wizard_list[i])
```

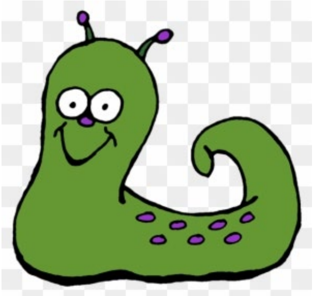
What if you don’t know how long the list is?

Use the **len()** function. The len() function returns the length of a string or list.

```
for i in range (0, len(wizard_list)):
      print(wizard_list[i])
```

Another way to output the values in a list:

```
for i in wizard_list:
        print(i)
>>>
spider legs
toe of frog
snail tongue
bat wing
slug butter
snake dandruff
bear burp
```



This code is a way of saying, “For each item in wizard_list, store the value in the variable i, and then print the contents of that variable.

Example 2

```
hugehairypants = ['huge', 'hairy', 'pants']

for i in hugehairypants:

       print(i)

>>>

huge

hairy

pants
```

Problems

1. Create a list containing six different sandwich ingredients, such as the following:

ingredients = [‘lettuce’, ’tomatoes’, ‘onions’, ‘pickles’, ‘mayo’, ‘mustard’]

Now, create a loop that prints out the list (including the numbers):

e.g.,

1. lettuce

2 tomatoes

3 onions

4 pickles

5 mayo

6 mustard

2. Create a list with at least 6 items. You can put anything you want in the list, e.g., favorite movies, favorite food, favorite games, etc. Next, write the code to output the values in the list in reverse order.

   For example, if your list looked like this: [‘donuts’, ‘ice-cream’, ‘waffles’, ‘cookies’, ‘hot dogs’, ‘cup cakes’], your output should look like the following:

   cup cakes
   hot dogs
   cookies
   waffles
   ice-cream

donuts

Adding Items to a List

To add items to a list, we use the append function.

For example, to add the item newt eye to the wizard’s shopping list, do this:

```
>>> wizard_list.append(‘newt eye’)

>>> print(wizard_list)

[‘spider legs’, ‘toe of frog’, ‘snail tongue’, ‘bat wing’, ‘slug butter’,
‘snake dandruff’, ‘bear burp’, ‘newt eye’]
```

You can keep adding more items to the wizard’s list in the same way, e.g.,

```
>>> wizard_list.append(‘hemlock’)

>>> wizard_list.append(‘swamp gas’)

Now the wizard’s list looks like this:

>>> print(wizard_list)

[‘spider legs’, ‘toe of frog’, ‘snail tongue’, ‘bat wing’, ‘slug butter’,
‘snake dandruff’, ‘bear burp’, ‘newt eye’, ‘hemlock’, ‘swamp gas’]
```

Removing Items from a List

To remove items from a list, use the **del** command (short for delete).

For example, to remove the sixth item in the wizard’s list, snake dandruff, do this:

```
>>> del wizard_list[5]

>>> print(wizard_list)

[‘spider legs’, ‘toe of frog’, ‘snail tongue’, ‘bat wing’, ‘slug butter’,
‘bear burp’, ‘newt eye’, ‘hemlock’, ‘swamp gas’]
```

Remember that positions start at zero, so wizard_list[5] actually refers to the sixth item in the list

Searching for an Item in a List

The following example shows how you can search for a particular item in a list. In this example, we wish to see if the number three is in the list. If we find the number, then we break out of the loop (as we don’t need to keep searching, assuming that each number in the list is unique).

Example 1

```
my_list = [1, 2, 3, 4, 5]
item_to_search = 3
found = False

for item in my_list:
    if item == item_to_search:
        found = True
        break

if found:
    print("Item found in the
list!")
else:
    print("Item not found in the list.")
```

Example 2

The following example shows how you can search for all occurrences of a particular number in a list. In this example, we wish to know how if the name “Ria” occurs in the list and in which positions in the list it occurs.

```
name_list = [“Jose”, “Alex”, “Ria”, “Emily”, “Mimi”, “Julio”, “Sandy”, “Ria”,
“Dani”]
item_to_search = “Ria”
found = False

for i in range (0, len(name_list)):
    if item == item_to_search:
        found = True
        print(item_to_search+” found in position “+str(i)+”\n”)

if !found:

    print(item_to_search+” not found in the list”)
```

Example 2

This example is another solution to the problem but here we append all the indexes where the name is found to a new list and then print out the new list.

```
name_list = [“Jose”, “Alex”, “Ria”, “Emily”, “Mimi”, “Julio”, “Sandy”, “Ria”,
“Dani”]

position_list = []

item_to_search = “Ria”

found = False


for i in range (0, len(name_list)):

    if item == item_to_search:

        found = True

        position_list.append(i)


if found:

    print(item_to_search+” was found in the following locations:”)

    for index in position_list:

        print(index)

else:

    print(item_to_search+” not found in the list”)
```

Practice Questions

1. Write a program that prompts the user to enter a list of numbers. The program should then count and display the number of elements in the list.

2. Write a program that prompts the user to enter a list of names. The program should reverse the order of the elements in the list and display the reversed list.

3. Write a program that prompts the user to enter a list of numbers. The program should calculate and display the sum of all the numbers in the list.