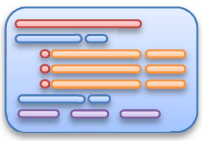


A list in Python is an ordered set of data. The following line of code will create a list called *my_list*. In this case, the list will contain integers.

***Aim:*** *To set up and edit a list.*

```
my_list = [12, 22, 351, 7812]
```

The list has four items (or elements). You can find what data is in any position in the list using the index. The indexes start at 0.

| | |
|---|---|
| `my_list[0]` | *This is 12, the first item in the list.* |
| `my_list[1]` | *This is 22.* |
| `my_list[2]` | *This is 351.* |
| `my_list[3]` | *This is 7812, the last item in the list.* |

## Task 1 – Investigating Lists

The program below creates a list of numbers set as the strings "One", "Two", "Three" and "Four". When the program is run, the console displays the output shown on the right.

```
List created
Checking list in order: One
Checking list in order: Two
Checking list in order: Three
Checking list in order: Four
The 1st item in this list is: One
The 4th item in this list is: Four
One
Two
Three
```

**a.** Recreate the program in *repl.it*, saving as '**13.1 Basic Lists**'. Use the code and the output to work out exactly what each line is doing and add comments to the program to demonstrate your understanding.

***Note:*** *This program uses 'for' loops, a type of count-controlled loop.*

```
1  numbers_list = ["One", "Two", "Three", "Four"]
2
3  print("List created")
4
5  for elem in numbers_list:
6    print("Checking list in order: " + elem )
7
8  print("The 1st item in this list is: " + numbers_list[0] )
9  print("The 4th item in this list is: " + numbers_list[3] )
10
11 for i in range(0, 3):
12   print(numbers_list[i])
```

**Run through each element of the list and display it.**

**Find the data in any position using the indexes, which start at 0 for the first element.**

**Start the variable i at 0 and add 1 each loop until you reach 3.**

**b.** There is a problem with the last section. It's only displaying "One", "Two" and "Three". This is because the range (0,3) only includes the numbers 0, 1 and 2; it stops short of the last item. Change the code so that it displays the whole list.

**c.** Edit the code so that the list includes numbers up to "Six". It should also display all six numbers when running the last section of code.

**d.** Add a line that writes "The 5th item in this list is…" to the console, as has been done with the 1st and 4th items.

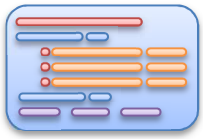
**e.** Add some blank lines to separate the different sections in the output. Use the code: `print("")`

**f.** Add a section which displays the number of items in the list. Use the code: `items_int = len(numbers_list)`

**g.** When you added items to the list, you would have had to change the last number on line 11 (in our original code above). It would be better for this to use a variable so that the change is automatic. Try the code on the right. Save your work.

```
for i in range(0, items_int):
```

# Lists (page 2)



## Task 2 – Editing Lists

Once your list has been created, you can add items to the end of it using a statement such as the following:

```
numbers_list.append("Seven")
```

You can also insert items closer to the start of your list, sort the list, change items or delete them altogether.

| | |
|---|---|
| `list.append("one")` | Add an item to the end of a list. |
| `list.insert(2,"one")` | Insert a new item into the list with the index [2]. Move everything after this down. |
| `list.remove("one")` | Removes a named item from a list. |
| `del list[2]` | Removes an item from a certain position in a list, given by its index. |
| `list.sort()` | Sorts a list. |
| `print(list)` | Outputs a whole list on one line. |

Your challenge is to use these methods and functions to create the output shown on the right.

Start with the code below and continue using one of the methods above before reprinting the list each time.

Save your program as '**13.2 Editing Lists**'.

```
['One', 'Two', 'Three']
['One', 'Two', 'Three', 'Four']
['One', 'Two', 'Five', 'Three', 'Four']
['Two', 'Five', 'Three', 'Four']
['Two', 'Three', 'Four']
['Four', 'Three', 'Two']
>
```

```
1  numbers_list = ["One", "Two", "Three"]
2
3  print(numbers_list)                          #Print the whole list
4
5  numbers_list.append("Four")                  #Add a "Four" to the end of the list
6
7  print(numbers_list)                          #Print the whole list
8
```

## Task 3 – Lists of Numbers

You can create a list using any data type, including integers and decimals. Remember that when using numbers, you do not need to put the values in quotes.

```
numbers_list = [1, 2, 3]
```

Create the program on the right and name it '**13.3 Numbers**'. The program prints out the 14 times table up to 10 times.

**a.** Edit the program so that it prints out the 13 times table up to 15 times.

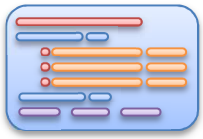
**b.** Adapt the program so that it prints out the times tables as shown. There are lots of ways of doing this.

```
1 x 13 = 13
2 x 13 = 26
3 x 13 = 39
```

```
1  numbers_list = []
2
3  for i in range(1,11):
4
5      numbers_list.append(i * 14)
6
7  print(numbers_list)
8
```

### Extension

1. Edit the program so that the user can input a number and the program displays the times table up to 15 times. Add a validation check to the input so that only numbers can be entered. You can decide if this includes decimals. Save as '**13.3 Extension**'.
2. Why do we have to include line 1 in this program? What happens if this code is removed?
3. What happens if you use code such as list.remove("One") when "One" is present multiple times in the list? Prove this.



## Task 4 – Random Questions

Your task is to build a program that selects questions randomly from a list. To produce anything random on a computer, you usually generate a random number and then work out how to use it.

***Note:*** *Before using the random functions, we need to import the 'random' module.*

**a.** Recreate the code on the right and run it several times. Look at the numbers generated. With this code, the computer produces random numbers between 0 and 9. Save the program as '**13.4 Random Qs**'.

```
1  import random
2  print(random.randint(0,9))
3
```

**b.** We are going to have a list of five questions for the program to pick from. These will have the indexes 0, 1, 2, 3 and 4. We therefore need to generate a random number between 0 and 4 inclusive and use this to pick a question. Adapt the code so that it produces a random integer between 0 and 4 and stores this in a variable called *random_int*.

**c.** Create a list which holds five questions. You can decide what questions to ask.

As before, we have started by creating an empty list and then appended each item to it. There are other ways of creating the list.

```
5  question_list=[]
6
7  question_list.append("What is thir
8  question_list.append("How many leg
9  question_list.append("What element
```

**d.** Display your random question using code like that on the right.

```
print(question_list[random_int])
```

**e.** Adapt your code so that it asks any of the five questions in a random order. Use a *for* loop to run through a process five times. Our solution is partly shown on the right.

***Note:*** *At this point, the program might ask the same question multiple times.*

```
for i in range(
  random_int =
  print(questio
```

**f.** We want to show each question only once, but still in a random order. Use code like that shown on the right to delete a question from the list once it has been asked. Run your program several times. What is the problem?

```
del question_list[random_int]
```

**g.** The issue is that once we delete a question, we only have four items left in the list. These will have the indexes 0, 1, 2 and 3. However, our random number generator is still producing numbers between 0 and 4. If it comes up with a 4, the program will generate an error because there isn't a list element with that index. We want to tell the random number generator to produce numbers between 0 and 3. And after deleting another question, we need to only generate numbers between 0 and 2. We can introduce a variable to do this for us.

To start with, give this variable the value 4. This must be before your loop starts running.

```
top_index = 4
```

**h.** Use the variable in place of the highest index when generating numbers.

```
random_int = random.randint(0,top_index)
```

**i.** And finally, reduce the value of the top index by 1 each time the loop runs.

```
top_index = top_index - 1
```

### Extension

In a copy of the program named '**13.4 Extension**', see if you can find a way of allowing users to enter an answer after each question. Count the correct answers and display this at the end. One possible solution is to have a second list with the answers (remember to delete answers as well, so that the lists stay aligned). How will you deal with answers when they contain multiple words? And is "Dog" the same answer as "dog" or "DOG"? Keep developing.

```
What is thirteen squared?
Enter your answer: 169

You scored 4 out of 5
```