# Assignment #4

# Lists in Python

Free imaginary Python t-shirt With every assignment completed

In Python, **lists** are a special data type. They allow us to group pieces of data together **in order** under a common name. We create them so we can keep information **organized** and **easily accessible**.

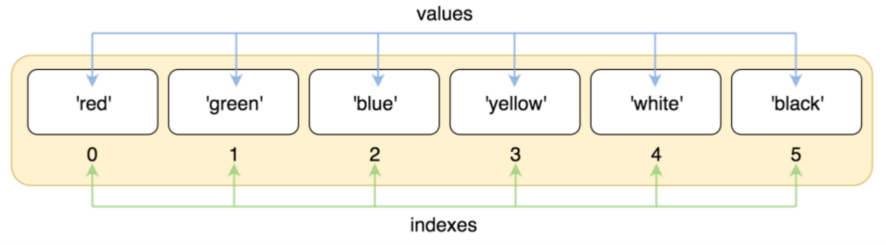
Examples of creating **Lists** in python:

```
School_Grades = [87, 91, 95, 81, 63, 50]

names = ["Sara", "David", "Warner", "Sandy"]

student_info = ["Sara", 1, "Chemistry"]
```

Lists (or **Arrays** as they are sometimes called in other programming languages) are an **essential** part of computer programming.  You can't get very far without them.

The great thing about lists is that they help **keep track of where things are**.  Each **element** in the **list** has its **own address** or "**index**" that we can use to find stuff in our list.

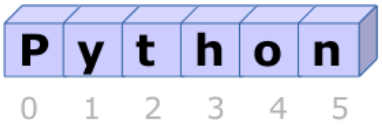
**Example**:



```
color_list=["red", "green", "blue", "yellow", "white", "black"]

print(color_list[0])="red"
print(color_list[1])="green"
print(color_list[5])="black"
```

Notice that each element in the list has it's own address that you can use to access the element.

**Important: first box (index)** is automatically labelled **0** and then we have box 1, box 2 box 3 etc.



# Exercise #1

Try it out:

Enter the following code into Replit. Then run the program and *look carefully at what is happing*. **Type out the code rather than cutting and pasting**. Trust me, cut and paste will cause you to miss important syntax.

```
my_list = [5, 12, 13, 14] # the list contains all integer values
print(my_list)

print(my_list[0])
print(my_list[1])
```

Now try: (type in the code and run it in Replit)

```
color_list=["Red", "Blue", "Green", "Black"]

print(color_list[0]) # Return the First Element
print(color_list[0],color_list[3]) # Print First and Last Elements
```

Now try **Adding an element** to a list (very useful and important!)

```
color_list=["Red", "Blue", "Green", "Black"]
print(color_list)

color_list.append("Yellow") # append adds another element to the end of a list

print(color_list)
['Red', 'Blue', 'Green', 'Black', 'Yellow'] # yellow has been added to the end
```

Now try:

```
color_list=["Red", "Blue", "Green", "Black"]
print(color_list)

new_color=input("Type in a color you would like to add to the list: ")

color_list.append(new_color) # adds another new_color to the end of a list

print(color_list)
['Red', 'Blue', 'Green', 'Black', 'Yellow'] # yellow has been added to the end
```

From the previous exercise you will see that creating a lists is easy. And grabbing particular items in the list using their **index** is easy to. You will see later that this way of organizing information can be super useful.

Lists have many **built in functions** that help us **sort** them, **add** to them, **delete** items, **count** items etc. The following exercises will take you through some of the more important functions you need to work with lists.

# **Exercise #2** User builds a list

Creating a list is easy enough, but most often we will want take data from user or external file and put it in a list.

Below is an example of how we can take **many items from a user** and put it neatly into a list for future use (**type** the code into Repit, and run the program, and **make sure you know how it works**):

**While LOOP**
While the variable “friend” is **not equal** to “end” the program will execute the indented lines of code repeatedly.
Note: **!=** means **not equal to**

```
Friend_List = []
friend= " "

while friend !="end":
    friend=input("Enter a friend to add to the list. If done type end:")
    Friend_List.append(friend)
    # user keeps adding friends to Friend_list

print("Here is a list of your friends!")
print(Friend_List)
```

Another way to get a user to enter a list of data. Try it out:

```
input_string = input("Enter a list elements separated by space ")

userList = input_string.split()
print("user list is: ")
print(userList)
```

### Common **list** functions:

These **functions** are tools included in python that help you manipulate lists.

**`len(list)`**
Gives the total length of the list.

**`max(list)`**
Returns item from the list with max value.

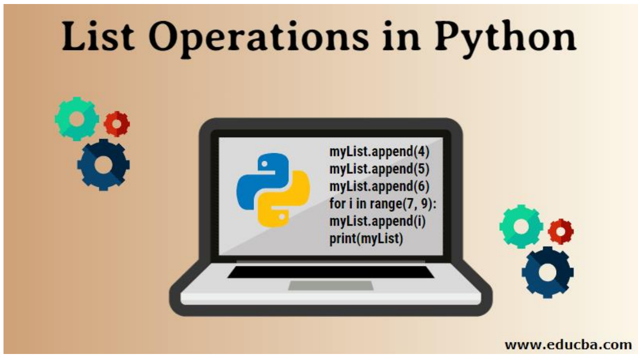
**`min(list)`**
Returns item from the list with min value.

**`list.append(element)`**
Adds element to the end of a list.

**`list.count(element)`**
Returns count of how many times element occurs in list

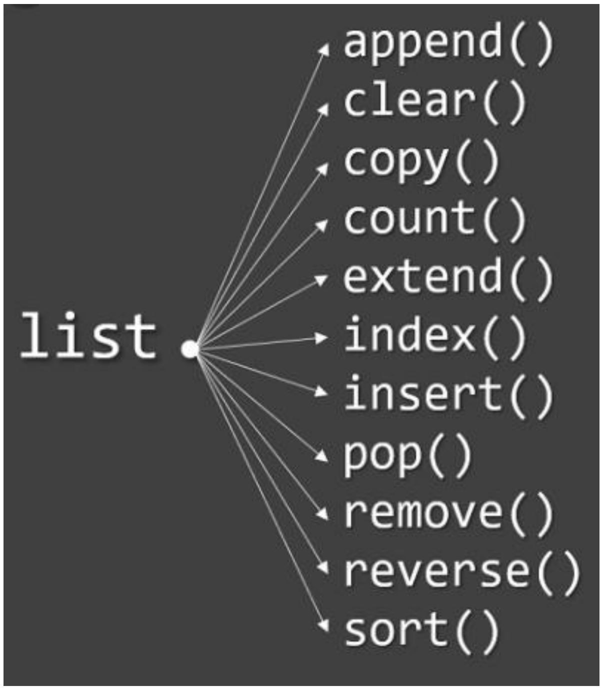
**`list.remove(element)`**
Removes element from list

**`list.reverse()`**
Reverses the order of elements in a list



```
myList.append(4)
myList.append(5)
myList.append(6)
for i in range(7, 9):
myList.append(i)
print(myList)
```

www.educba.com



**`list.index(element)`**
returns the index box of which the element of interest is contained.  If there is more than one item with value x, this function will return the first index location.

# Exercise #3

Write a series *mini* programs. One for **each** of the functions listed above in the **grey boxes** (not the black ones).  Each program should use the list method to manipulate a list according to its function.
Use the # to put comment after each mini program to explain what each method does.
Examples on the next page…**type** them in and then add to them…you should have 8 mini programs total when your done

## Example#1

```
color_list=["Red", "Blue", "Green", "Black", "Blue", "White"]
x = len(color_list)
print(x)
# the len() function should return 6 (The length of the list).
```

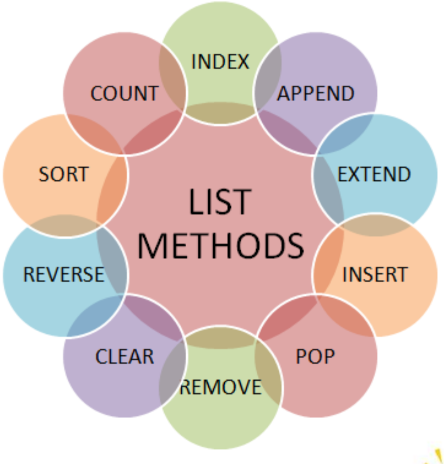
## Example#2

```
my_list = [1, 2, 3, 7, 9, 234, 311, 45, 67, ,12]
x = max(my_list)
print('the largest number in your list is')
print(x)
# the max() function should return the maximum value in the list
```

**You need to create 6 more. Of these**, one for each of the following:

```
 min()
.append()
.count()
.remove()
.reverse()
.index()
```

Hand in all 8 for **Exercise#3**.  Make sure you are using the # *and explaining each function.*



## Exercise#4

**Modify** the code below to create a program that can do the following *repeatedly* do the following:

a) Take a number from a user
b) Check to see if it is **even** or **odd**
c) Put the number to either one of two **lists**: The **even** list or The **odd** list.
d) When the user types “0”  the program will end and print out both **lists** with an appropriate title for each.

```
Odds = []
Evens = []
num = 1

while num !=0:
    num=int(input("give me a number! Enter 0 to exit:"))
    if num%2=0:
       Evens.append(num)

# note the code above is not correct or complete. You will have to modify it according to the instructions above to get full marks
```

# Exercise#5

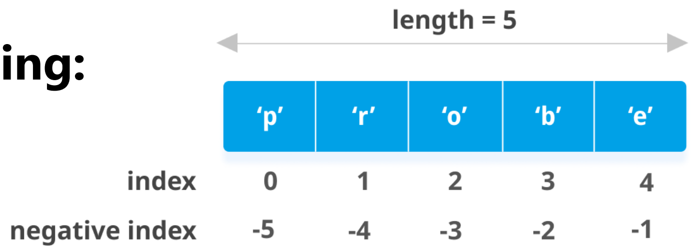
The code below uses a list of numbers that directs python turtle to draw a cool image. **Copy and paste the code below into Replit** and run the code.

**Now compose your own program** that uses a **list** and a **For Loop**, to create a cool image of your own.  Use your imagination and have fun!

```
import turtle as t
t.speed(7)
my_list = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120]
t.color('dark green')

for value in my_list:
  t.forward(value)
  t.backward(value*2)
  t.forward(value)
  t.right(90)
  t.forward(8)
  t.left(90)

t.color("brown")
t.begin_fill()
t.forward(5)
t.right(90)
t.forward(30)
t.right(90)
t.forward(10)
t.right(90)
t.forward(30)
t.right(90)
t.forward(5)
t.end_fill()
t.penup()
t.hideturtle()
t.goto(-20,40)
t.color("blue")
t.write("What A Cool Tree!", None, "center", "16pt bold")
```

This is a **For LOOP** that will **repeat** the code in blue for each **value** in **my_list**

## Negative Indexing:



If you look at the graphic above you will see that each box in the list actually has **2** available addresses (indexes) you can use.

You can retrieve the **last** element of the list using : `listA[-1])`
`listA[-2])` is the **second to last** element and so on…

**Example:**

`listA=[1,2,3,4,5,7,8,14]`
`print(listA[-1])` this would print the value **14**
`print(listA[-4])` this would print the value **5**

## Exercise#6

**Your Task**: Use the **reverse** indexing to print in following elements in the list shown below:

a) T
b) E
c) R
d) Z
e) W

```
my_list = ['w','a','l','z','l','i','s','g','r','e','a','t']
```

# Exercise#7

A psychologist recently did an experiment and has just determined that when someone lists the Christmas gifts they would like to get, the ones they *really* want the most are:

- the **first** gift in the list and
- **second to last** gift they listed.

**For Example:**

Jim says he wants a: **car**, phone, candle, gym membership, **a watch**, and a book.

```
gifts=['car', 'phone', 'candle', 'gym membership', 'a watch', 'book']
```

Jim *really* wants **a Car** and **a watch** the most.

Write a program that:

a) **Allows user** to create the Christmas list by *repeatedly* adding individual words to a list (see exercise#2).

b) Grabs the **first value** and the **second to *last*** value of the list the user has created and puts them into a new list called: `gifts_they_want_most []`

c) then print the new list.
You **MUST** use **negative indexing**

## Exercise#8

Create a program that: draws a series of 5 circles next to one another (using python turtle) using a '**For Loop'**. Each of the 5 circles will be colored in with a **randomly** selected color **from a list** of 10 colors you have created.

```
Color_list = [ ‘yellow’, ‘gold’, ‘orange’, ‘red’, ’black’, ‘violet’,
‘magenta’, ‘purple’, ‘navy’, ‘blue’, ‘cyan’]
```

**Important Hints**:

a) you will need to use a **for loop** like the one in **exercise #5**, but you need to use the **`for i in range(5):`** to get it to repeat **5 times**.
Try the code below as an example:

```
for i in range(5):
    print(i)
```

b) you also will need to **import the random library** and use the `random.choice(list)` function to select each color randomly each time the loop runs. Try the following code as an example:

```
import random
mylist = ["apple", "banana", "cherry"]
print(random.choice(mylist))
```

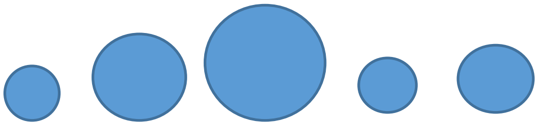
## Exercise#9

Watch the `pop()` method and `sort()` method videos on the course page. Create a program to that uses the `sort()` method to sort a **user created** list of integers and then remove the last three items using the `pop()` method.

## Exercise#10

Your Computer Science teacher wants to make a sorted list of best grades to least grades from both classes. Get a user to enter two separate lists of grades. Combine the lists using the `extend()` method and then `sort()` the list from **greatest to least** values.

## Exercise#11



A create a program that draws a series of 5 circles neatly spaced out in a line. Each time a circle is drawn you must take the **size** of the circle from a list of integers that the **user has already created** for you. Tell the user that each integer in the list should be significantly different and in the range of 20-300.

- Look back at **Exercise#4** to see how to get a **user generated** list of integers.
- You will have to use `for loop` to grab each size in the **user generated** list:

**Example**:

```
for x in circle_size:
  t.circle(x)
  t.forward(30)  # only an example (don’t use this exactly)
```

## Exercise#12

Create a program that:

a) gets the python turtle to move in a circle **slowly.** *Draw a circle with a the penup()command*

b) move the turtle to another **random** location and have the turtle move in another circle path of similar size *(or the same size)*

c) each time you move locations increase the **speed** of the turtle so each successive circular path is done more quickly. *(look up how to change the turtle’s speed)*.

d) You are required to use a **list** of speeds (integers) and a **for loop** to complete the task above.

## Bonus Exercise:

(not mandatory) but for extra marks

Add to the code to exercise#11 so that every time a user clicks on turtle they get a point and their points are recorded (on the turtle screen). The game is to see how many clicks you can get before the turtle completes 10 circles