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**.

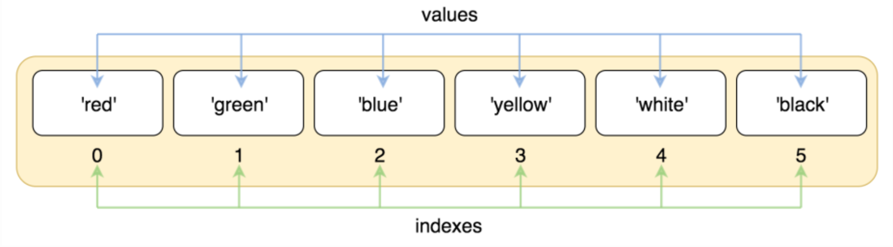
Examples of Creating **Lists** in python:

```
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.

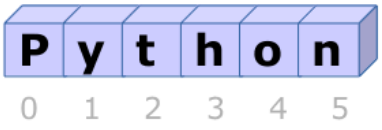
Example:



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

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

That 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 Trinket. Then run the program. **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 Trinket)

```
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 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 trinket, and run the program, and **make sure you know how it works**):

While LOOP
While the variable "friend" is not "end" the program will execute the indented lines of code.

```
Friend_List = []
friend= " "

while friend is not "end":
    friend=input("Enter a friend to add to the list. If done type end:")
    Friend_List.append(friend)

print("Here is a list of your friends!")
print(Friend_List)
Append example from user (repeated)
```

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 like mini programs 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.

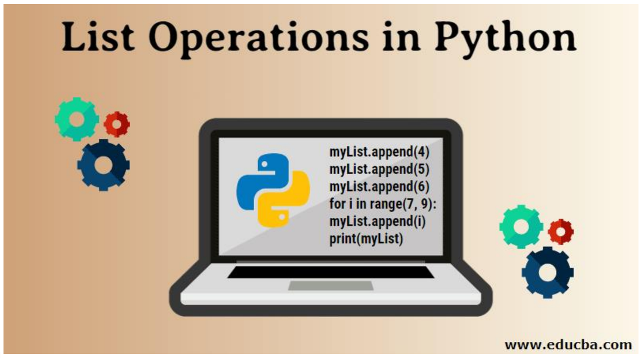
`min(list)`
Returns item from the list with min value.

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

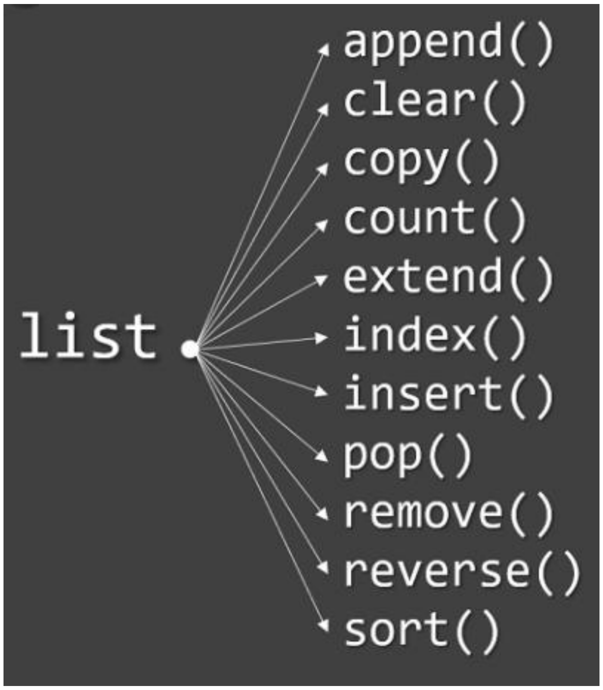
`list.count(element)`
Returns count of how many times element occurs in list

`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)
```


list

append()
clear()
copy()
count()
extend()
index()
insert()
pop()
remove()
reverse()
sort()

`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 one). Each program should use the list method to manipulate a list according to its function. Have fun with it...maybe ask the user for input and interact with them in a fun way using the above functions.

Examples on the next page...**type** them in and then add to them

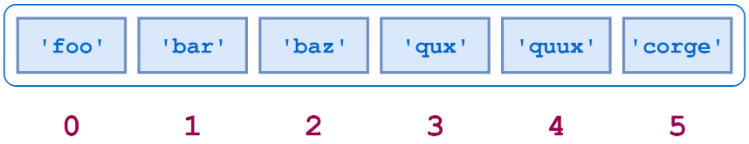
Example#1

```
color_list=["Red", "Blue", "Green", "Black","Blue","White"]
print(color_list)
print(color_list.count('Blue')) # this should return 2. Blue occurs twice
```

Example#2

```
my_list = input("Please enter five numbers separated by a single space only")
print(my_list)
print('the largest number in your list is')
max=max(my_list)
print(max)
```

Exercise#4



Write a program that will:

a) Take a number from a user
b) Check to see if it is even or odd
c) Add the number to either one of two lists:

1. The even list
2. The odd list

When the user types “e” (for exit) the program will print out both lists with an appropriate title for each.

You will have to use the `append` command in python:

```
fruits = ['apple', 'banana', 'cherry']
fruits.append("orange")
```

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 trinket** and run the code.

Now compose your own program that uses a **list** 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**

Exercise#6

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 takes the **first value** of a list and the **second to last** value of an list and puts them into a new list called:

```
gifts_they_want_most []
```

then print the new list.

Assume you **don't** initially know the length of the original list.

Hint: use the `len(list)` function

Make sure the program runs so that it **allows user to create the Christmas list they want** by adding individual words of their choice.

Exercise#6

A Create a program that draws a series of 5 circles next to one another. Each time the program is run each circle will be colored (and filled in) a **randomly** selected color **from a list** of 10 colors you have selected.

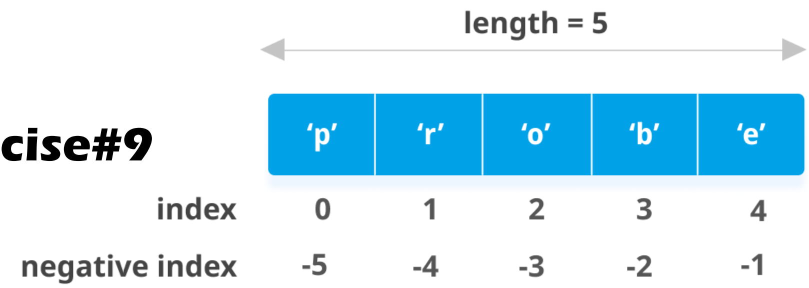
Exercise#7

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 numbers and then remove the last three items using the `pop()` method.

Exercise#8

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#9



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**

Your Task: Use the reverse indexing to print the first, last, and second to last elements in the following list:

```
my_list = ['p','r','o','g','r','a','m','m','I','n','g']
```

Exercise#10

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 that the user creates for you. Each size in the list should be significantly different.

You might have to use a `for loop`:

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

Exercise#11 (for bonus **not mandatory**)

Bonus: figure out how to import a list into Trinket from an external file!