Syntax

In [1]:
```
print("Welcome to C-DAC")
```

```
Welcome to C-DAC
```

In [4]:
```
if 4>2:
    print("4 greater than 2")
```

```
4 greater than 2
```

In [5]:
```
print("Welcome to C-DAC") #print welcome to c-dac
```

```
Welcome to C-DAC
```

In [6]:
```
'''cdac mein apka swagat hai'''
```

Out[6]:
```
'cdac mein apka swagat hai'
```

Variables

In [7]:
```
1
```

Out[7]:
```
1
```

In [8]:
```
student = "Himani"
```

In [9]:
```
student
```

Out[9]:
```
'Himani'
```

In [10]:
```
1student="himani"
```

```
  Input In [10]
    1student="himani"
     ^
SyntaxError: invalid syntax
```

In [11]:
```
s1tudent="Himani"
```

In [12]:
```
@studnt="himani"
```

```
  Input In [12]
    @studnt="himani"
            ^
SyntaxError: invalid syntax
```

In [13]:
```
num_of_schools=10
```

In [14]:
```
numSchool=10
```

In [15]:
```
Schoolnumber=10
```

Scope of Variables

In [16]:
```
icecream = "Vanilla"    #global variable
def foods():
    vegetable = "Potato"    #local variable
    fruit = "Lichi"         #local variable
    print(vegetable + " is a local variable value.")
    print(icecream + " is a global variable value.")
    print(fruit + " is a local variable value.")

foods()
```

```
Potato is a local variable value.
Vanilla is a global variable value.
Lichi is a local variable value.
```

In [17]:
```
icecream = "Vanilla"    #global variable
def foods():
    vegetable = "Potato"    #local variable
    fruit = "Lichi"         #local variable
    print(vegetable + " is a local variable value.")

foods()
print(icecream + " is a global variable value.")
print(fruit + " is a local variable value.")
```

```
Potato is a local variable value.
Vanilla is a global variable value.

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Input In [17], in <cell line: 9>()
      7 foods()
      8 print(icecream + " is a global variable value.")
----> 9 print(fruit + " is a local variable value.")

NameError: name 'fruit' is not defined
```

In [18]:
```
a = "double quotes"
```

In [19]:
```
type(a)
```

Out[19]:
```
str
```

In [20]:
```
b = 1
type(b)
```

Out[20]:
```
int
```

In [21]:
```
c = 1.4
```

In [22]:
```
type(c)
```

Out[22]:
```
float
```

In [23]:
```
q = 1 + 2j
type(q)
```

Out[23]:
```
complex
```

In [24]:
```
len(a)
```

Out[24]: 13

In [25]:
```
a[0]
```

Out[25]: 'd'

In [26]:
```
a[-1]
```

Out[26]: 's'

In [28]:
```
a[2:4]
```

Out[28]: 'ub'

In [34]:
```
a[1:6]
```

Out[34]: 'ouble'

In [37]:
```
b=a.upper()
```

In [38]:
```
b
```

Out[38]: 'DOUBLE QUOTES'

In [39]:
```
a.replace("d","b")
```

Out[39]: 'bouble quotes'

In [40]:
```
a[1].replace("o","a")
```

Out[40]: 'a'

In [41]:
```
L = [1,"a","string",1+2]
```

In [42]:
```
L
```

Out[42]: [1, 'a', 'string', 3]

In [43]:
```
L.append(9)
```

In [44]:
```
L
```

Out[44]:
```
[1, 'a', 'string', 3, 9]
```

In [45]:
```
L.pop()
```

Out[45]:
```
9
```

In [46]:
```
L
```

Out[46]:
```
[1, 'a', 'string', 3]
```

In [47]:
```
L[1]
```

Out[47]:
```
'a'
```

In [48]:
```
pizza_type = ["Small", "Medium", "Large", "Extra Large"]
#              [0]      [1]        [2]      [3]
print(pizza_type[2])
print(pizza_type[0])
```

```
Large
Small
```

In [49]:
```
pizza_type = ["Small", "Medium", "Large", "Extra Large"]
#               [-4]    [-3]     [-2]      [-1]
print(pizza_type[-1])
print(pizza_type[-3])
```

```
Extra Large
Medium
```

In [50]:
```
animals = ["cat", "dog", "bat", "mouse", "pig", "horse", "donkey", "goat", "c
print(animals[3:7]) #using positive indexes
print(animals[-7:-2]) #using negative indexes
```

```
['mouse', 'pig', 'horse', 'donkey']
['bat', 'mouse', 'pig', 'horse', 'donkey']
```

In [51]:
```
#Example print alternate values

animals = ["cat", "dog", "bat", "mouse", "pig", "horse", "donkey", "goat", "c
print(animals[::2]) #using positive indexes
print(animals[-8:-1:2]) #using negative indexes
```

```
['cat', 'bat', 'pig', 'donkey', 'cow']
['dog', 'mouse', 'horse', 'goat']
```

In [52]:
```
#Example: printing every 3rd consecutive within a given range

animals = ["cat", "dog", "bat", "mouse", "pig", "horse", "donkey", "goat", "c
print(animals[1:8:3])
```

```
['dog', 'pig', 'goat']
```

In [53]:
```
# insert extra large in the list at plce second

pizza_type = ["small", "large", "medium"]
#           [0]        [1]      [2]

pizza_type.insert(1, "extra large")   #inserts item at index 1

print(pizza_type)
```

```
['small', 'extra large', 'large', 'medium']
```

In [54]:
```
#add a list to a list
colors = ["voilet", "indigo", "blue"]
rainbow = ["green", "yellow", "orange", "red"]
colors.extend(rainbow)
print(colors)
```

```
['voilet', 'indigo', 'blue', 'green', 'yellow', 'orange', 'red']
```

In [55]:
```
#concatenate two lists:

colors = ["voilet", "indigo", "blue", "green"]
colors2 = ["yellow", "orange", "red"]
print(colors + colors2)
```

```
['voilet', 'indigo', 'blue', 'green', 'yellow', 'orange', 'red']
```

In [56]:
```
# remove item

colors = ["voilet", "indigo", "blue", "green", "yellow"]
colors.remove("blue")
print(colors)
```

```
['voilet', 'indigo', 'green', 'yellow']
```

In [57]:
```
#del item

colors = ["voilet", "indigo", "blue", "green", "yellow"]
del colors
print(colors)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Input In [57], in <cell line: 5>()
      3 colors = ["voilet", "indigo", "blue", "green", "yellow"]
      4 del colors
----> 5 print(colors)

NameError: name 'colors' is not defined
```

In [58]:
```
#clear list

colors = ["voilet", "indigo", "blue", "green", "yellow"]
colors.clear()
print(colors)
```

```
[]
```

In [59]:
```
# change items in the list

names = ["raghav", "vishal", "neha", "ritika", "sourabh"]
names[2] = "pragiti"
print(names)
```

```
['raghav', 'vishal', 'pragiti', 'ritika', 'sourabh']
```