In [8]:
```
#USING THE SAMPLE FUNCTION
#SYNTAX IS sample(list_variable,group)
from random import *
l=["A","B","C","D","E","F","G","H"]
c=sample(l,2)
print(c)
```

```
['D', 'C']
```

In [2]:
```
#using choice function
#syntax is choice(list_variable)
from random import *
l=["A","B","C","D","E","F","G","H"]
c=choice(l)
print(c)
```

```
G
```

In [9]:
```
#using shuffle
#syntax shuffle(list_variable)
from random import *
l=["A","B","C","D","E","F","G","H"]
c=shuffle(l)
for p in range(len(l)):
    print(l[p],"its your turn")
print(c)
```

```
A its your turn
B its your turn
F its your turn
G its your turn
E its your turn
H its your turn
C its your turn
D its your turn
None
```

In [10]:
```
#split
s="hey! go home"
#splits the strings using commas by default
print(s.split())
```

```
['hey!', 'go', 'home']
```

In [59]:
```
#join
```

```
#syntax "joinchar".join(string_variable
#syntax "join_variable".join(string_variable
name="david"
#spr is the join variable
#name is the string_variable
spr="$"
d = spr.join(name)
print(d)
```

```
d$a$v$i$d
```

In [2]:
```
from string import punctuation
s = input("enter phrase")
for c in punctuation:
    s = s.replace(c,"")
print(s)
```

```
enter phraseflavian;s
flavians
```

In [3]:
```
l=[1,2,3,4,5,6,7,8]
for i in range(0,len(l)):
    if l[i]%2==0:
        l.append(l)
    del l[i]
    print(l)
```

```
[2, 3, 4, 5, 6, 7, 8]
[2, 4, 5, 6, 7, 8]
[2, 4, 6, 7, 8]
[2, 4, 6, 8]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-3-e513a9ddb369> in <module>
      1 l=[1,2,3,4,5,6,7,8]
      2 for i in range(0,len(l)):
----> 3     if l[i]%2==0:
      4         l.append(l)
      5     del l[i]

IndexError: list index out of range
```

In [4]:
```
#split
def split_list():#defining the function
```

```
        s ="flavian is a student"
        v = s.split()
        print(len(v))
    def m():
        print(v)
        for i in v:
            print(i)
    split_list()#calling
    m()
```

```
4
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-4-c702810be56f> in <module>
      9         print(i)
     10 split_list()#calling
---> 11 m()

<ipython-input-4-c702810be56f> in m()
      5     print(len(v))
      6 def m():
----> 7     print(v)
      8     for i in v:
      9         print(i)

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

In [5]:
```
#join
j ="flavian"
#syntax is "seperator".join(variable)
s ="-"
j=s.join(j)
print(j)
```

```
f-l-a-v-i-a-n
```

In [6]:
```
#list comprhension
[x**2 for x in range(1,11) if x%2==0]
#[exprssion for var in range(condition)if statements]
```

Out[6]:
```
[4, 16, 36, 64, 100]
```

In [7]:
```
l=[1,2,3,4,5,6,7,8]
for j in range(len(l)):
    l[j]=l[j]*2
```

```
        print(l[j])
```

```
2
4
6
8
10
12
14
16
```

In [6]:
```
def hello():#defining function
    print("hello")
hello()#call the function

    #syntax def name ():
```

In [16]:
```
def fun_name(l):#formal parameters
    for i in range (len(l)):
        print(i)

fun_name(l) #actual parameters
l=[1,2,3,4,5,6]
```

```
0
1
2
3
4
5
```

In [18]:
```
l=[1,2,3,4,5,6]
for i in range (len(l)):
        print(i)
```

```
0
1
2
3
4
5
```

In [ ]:
```
g=5#global variable
```

```
def fun_name():
    v=5    #local variable
    l=[1,2,3,4,5,6]
    for i in range (len(l)):
        print(i)
fun_name()

def var_access(g):#formal parameter
    print(g)
var_access(g) #actual parameter
```

In [23]:
```
def add(d,c,s):#formal parameters
    s=d+c
    print(s)
 add(5,4,s)#actual parameters
```

```
9
```

In [2]:
```
def get_unique():
    dup_arr=[10,20,30,40,50,60,70,80,90,100]
    unique_arr=[];
    for i in range(0,len(dup_arr)):
        for j in range(i+1,len(dup_arr)):
            if dup_arr[i]==dup_arr[j]:
                unique_arr[i]=dup_arr[j]
                print(unique_arr[i])
get_unique()
```

In [3]:
```
file2 = open("MyFile2.txt","w")
file2.write("hello flavian anselmo")
file2.close()
```

In [ ]: