Return: It completely comes out of the function.

Eg: With return and with input

```
def prime (r1,  r2):
    for i in range(r1,r2+1,1):
        for j in range(2,i,1):
            if(i%j==0):
                break
        else:
            return i
```

- Here in the above eg it returns only the 1st prime number i.e.,2
- S, here we use the concept called list[]
- The above eg can be written as,

Eg:

```
def prime(r1,r2):
    for i in range(r1, r2+1, 1):
        for j in range(2,i,1):
            if(i%j==0):
                break
        else:
            list.append(i)
    return list
```

O/P: [2,3,5,7]

Lambda Function: It's a small, anonymous function

- Defined using the 'lambda' keyword instead of 'def'.
- Can take any no. of arguments but must contain only one expression.
- Expression is automatically returned (No need to use return keyword).
- Syntax: lambda arguments:expression.
- Eg: s=lambda a,b:a+b

- O/P: s(2,2) =>4

26/9/25

DATA STRUCTURE IN PYTHON:

- Python DS are the ways of organising and sorting data so that they can be accessed and modified efficiently.
- Python provides both built-in DS and allows us to implement user-defined DS.

Built-in DS:

- **List:[]**
- **Tuple()**
- **Set:{}**
- **Dict:{key:value}**

=>List: It is a heterogenous DS which is ordered, mutable and allow duplicates.

Eg: val= input(“enter val: “).split()

O/P: enter val: yash 20 3.4

val: [‘yash’, ‘20’,’3.4’]

*We want numbers in int format use:

Num=list(map(int,input(“enter num”).split()))

O/P: enter num: 5 6 7
=> [5,6,7]

*We can represent list in 2 ways:

1.list(): It’s used when we do convertions like set to list.

2.[]: It’s commonly used to create a list.

Methods in list:

- To add a value: list.append()

  Eg: details.append("bangalore")

- To update or replace a value: list[index]=value

  Eg: details[2]=False

- To delete a value: list.pop()

  Eg: details.pop()

- To append the entire list: list.append([])

  Eg: details.append([1,2,3])

- To add list of items separately: create separate list and concat

  Eg: details=details+l2

- To add to a particular position without removing the element

  Eg: details.insert(2,"908765432")

- To remove a particular element

  Eg:details.pop(2)

- details.pop([2,3,4]): If we try to pop list of items it throws ‘Typeerror’
- To get no. of elements

  Eg: len(details)

- To see the num no. of occurance
  Eg:details.count(2)
- To delete all elements at once
  Eg:details.clear()
- To remove the complete list
  Eg: del details

  Eg: create even,odd,prime numbers list from 1-20

```
l2=[]
l3=[]
l4=[]
for i in range(1,21,1):
    if i%2==0:
        l2.append(i)
```

```
    else:
        l3.append(i)
    for j in range(2,i,1):
        if(i%j==0):
            break
    else:
        l4.append(i)
print(l2,l3,l4)
```

O/P:

[2, 4, 6, 8, 10, 12, 14, 16, 18, 20] [1, 3, 5, 7, 9, 11, 13, 15, 17, 19] [1, 2, 3, 5, 7, 11, 13, 17, 19]