Data Science & AI
&
NIC - Param

Python-For Data Science

Functions

Lecture No.- 03

By- Pankaj Sharma Sir

Recap of Previous Lecture

Topic Functions 2

Slide 2

Topics to be Covered

Topic Functions 3

Topic : Functions

```
if condition :
        S1
        S2
        S3
        S4
```

```
if(condition)
    {
        S1;
        S2;
        S3;
        S4;
    }
```

scope

```
if (condition)
        S1;
        S2;
```

→ Till 1st semi colon

⇓

```
if (condition){
        S1;
    }
S2;
```

```
if   a < b :

        s1
        s2

else :
        s3

        s4

s5
```

C syntax

```
if (a< b) {
            s1 ;
            s2 ;
            }

else {
        s3 ;
        s4 ;

     }
s5
```

C Syntax

```
if a<b :
    S1
    S2
elif c<d :
    S3
    S4
elif e<f :
    S5
    S6
S7
```

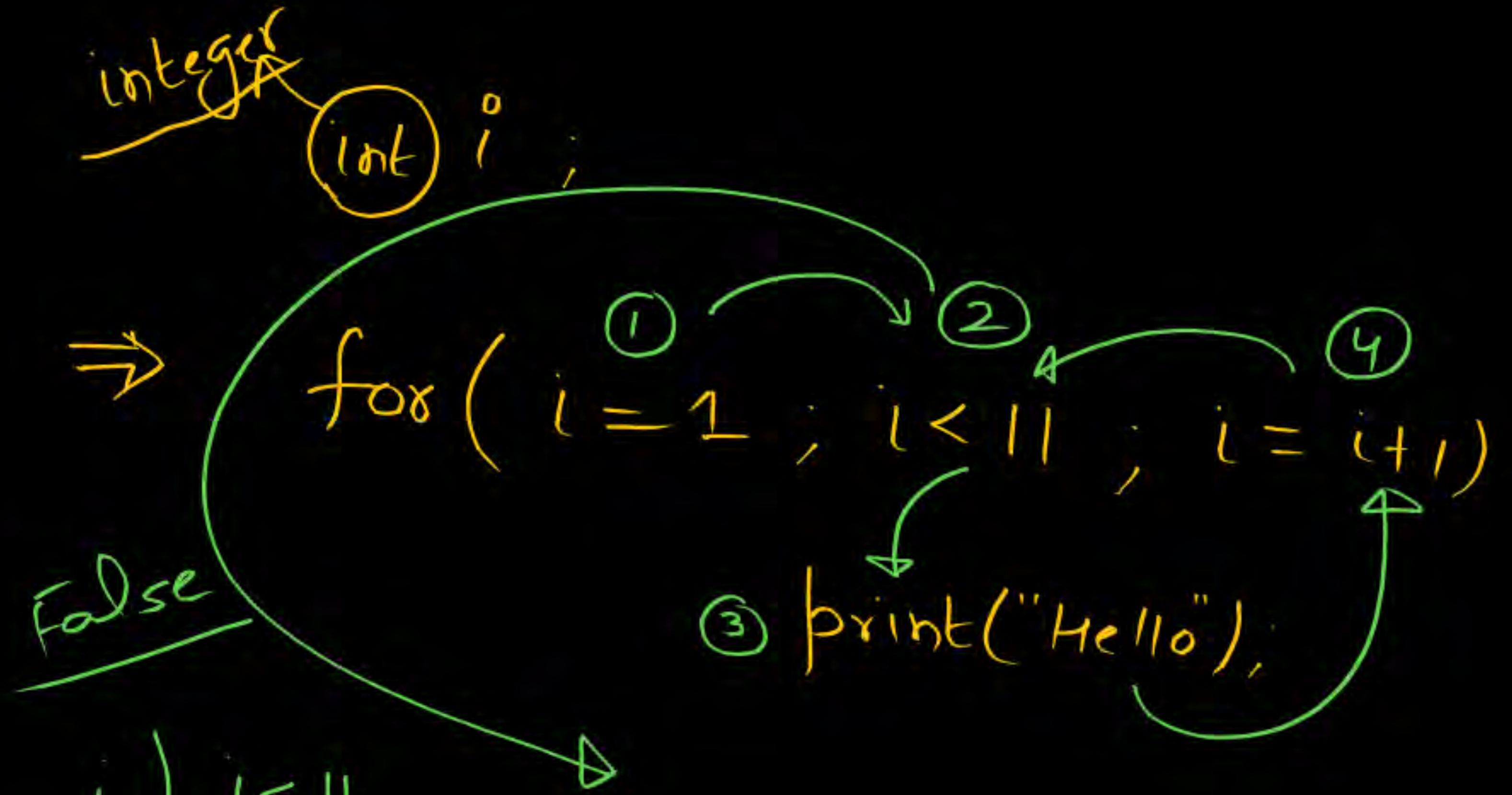
⟹

```
if (a<b){
        S1;
        S2;
    }
else if (c<d){
        S3;
        S4;
    }
else if (e<f){
        S5;
        S6;
    }
    S7;
```

start end

```
for  i  in  range(1, 11):
    print("GW")
```

⇒



```
for( i = 1 ; i < 11 ; i = i+1)
    print("Hello");
```

i	1<11		
1	1<11	→ True	→ Hello
2	2<11	→ True	→ Hello
3			
⋮			
10	10<11	→ True	→ Hello
11	11<11	→ False	

→ First semi colon

by default scope

```
for ( Initialization ; condition ; updation)
        {
         code
        }
```

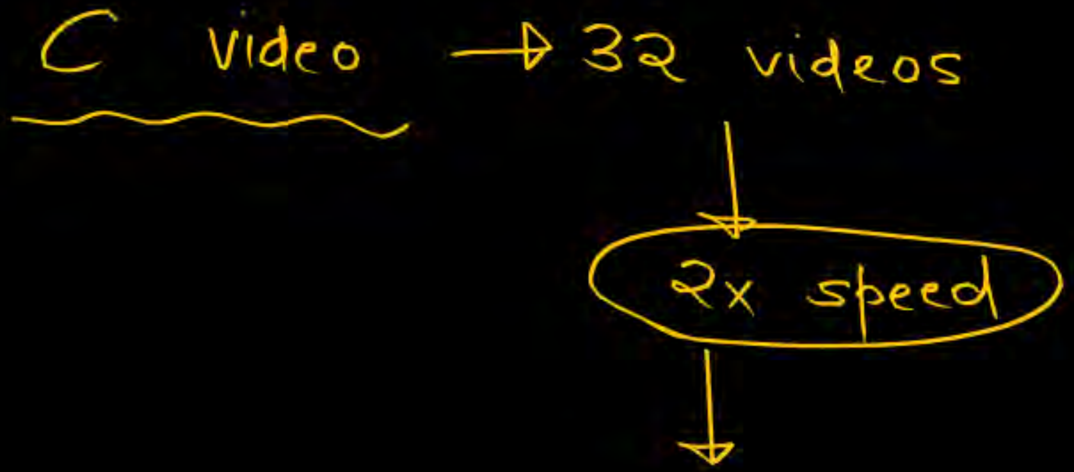
```
for (i=1; i<=11 ; i = i+2)
        print("Hello");
```

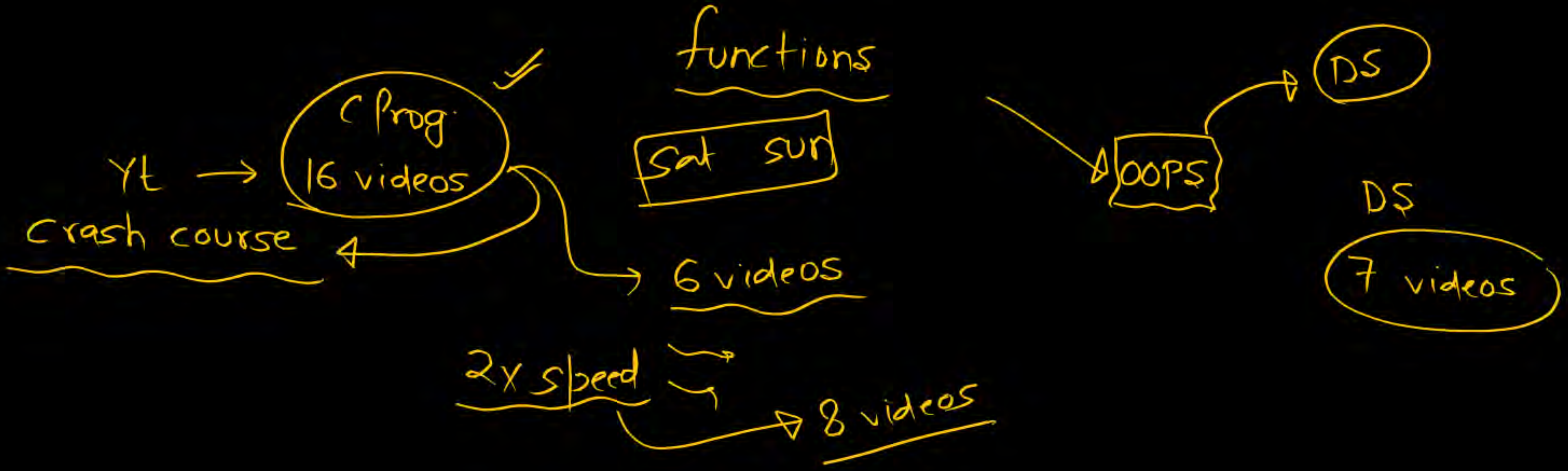
$i = \overset{\checkmark}{1}, \overset{\checkmark}{3}, \overset{\checkmark}{5}, \overset{\checkmark}{7}, \overset{\checkmark}{9}, \overset{\checkmark}{11}$

```
for ( i=1 ; i<11 ; i = i+2)
        print("Hello")
```

```
int Prod ;

Prod= 1
for(i=1; i<=5; i++)
      Prod = Prod * i
```

```
void main(){



}
```


C video
32 videos
2x speed


YT
C Prog.
16 videos
crash course
functions
sat sun
6 videos
2x speed
8 videos
OOPS
DS
DS
7 videos

```
def f(x) :
    return x*x*x
```

$$x \longrightarrow \underset{\sim}{f} \longrightarrow x*x*x$$

Anonymous function

$$s = \text{lambda}\ \overset{3}{x} : \overset{3\times3\times3}{x*x*x}$$

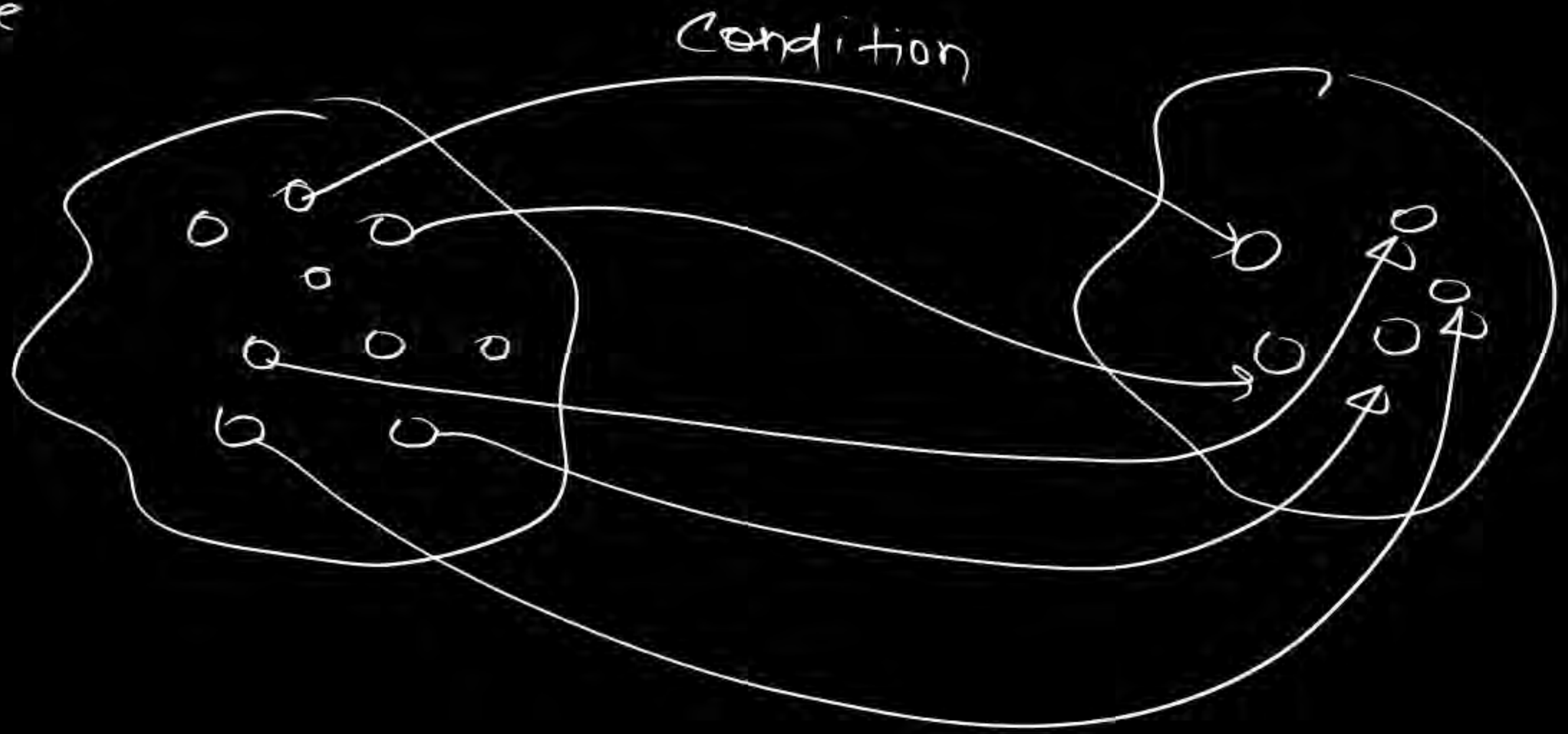
$$\text{print}(\underset{27}{s(3)})$$

f (function1, function2, value1, value2)

1.) filter ⟶

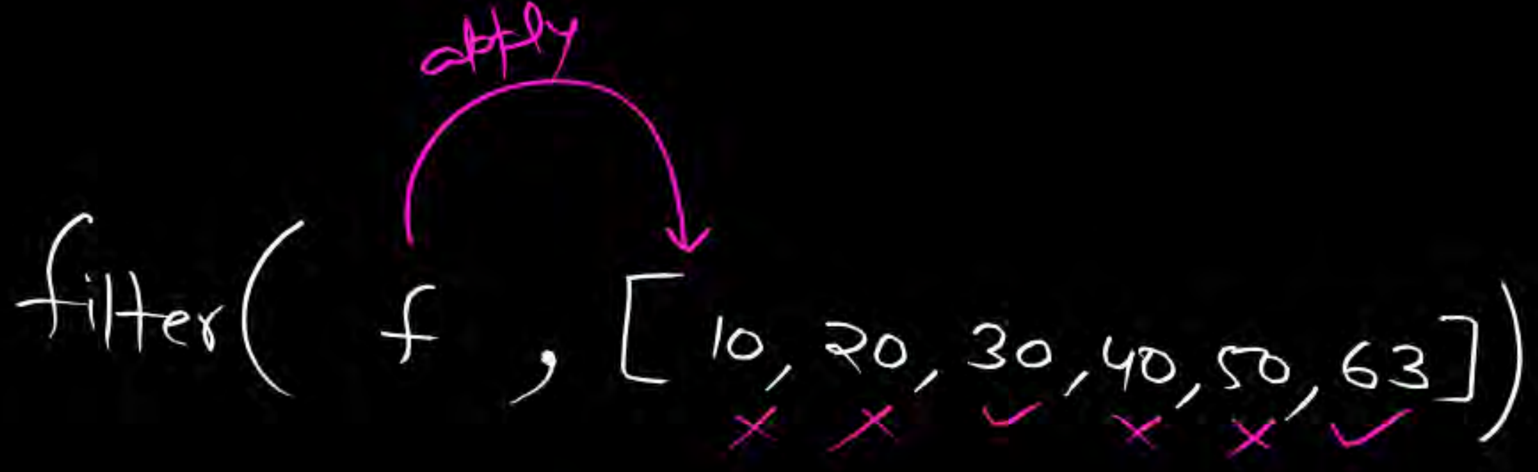
2.) map

3.) reduce



filter (function, sequence)

```
def f(x):
    if x%3 == 0:
        return True
    else
        return False
```



filter(f , [10, 20, 30, 40, 50, 63])

× × ✓ × × ✓

10 / 3==0 × 10 ×

20%3==0 × 20 ×

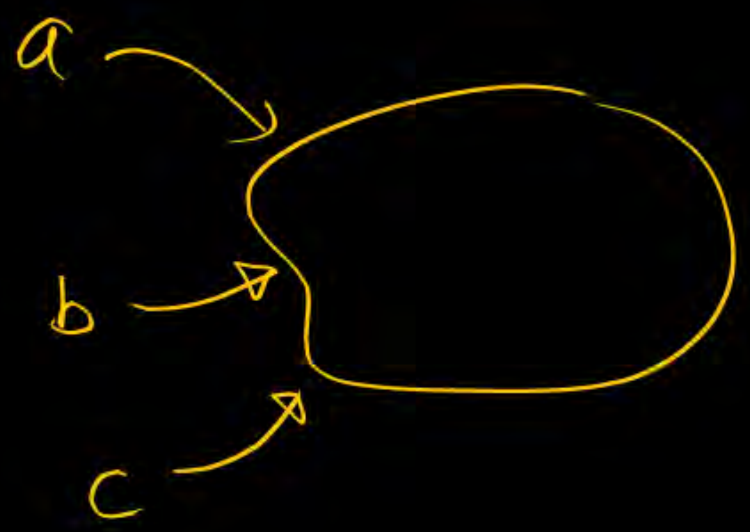
30 / 3==0 ✓ 30 ✓

map

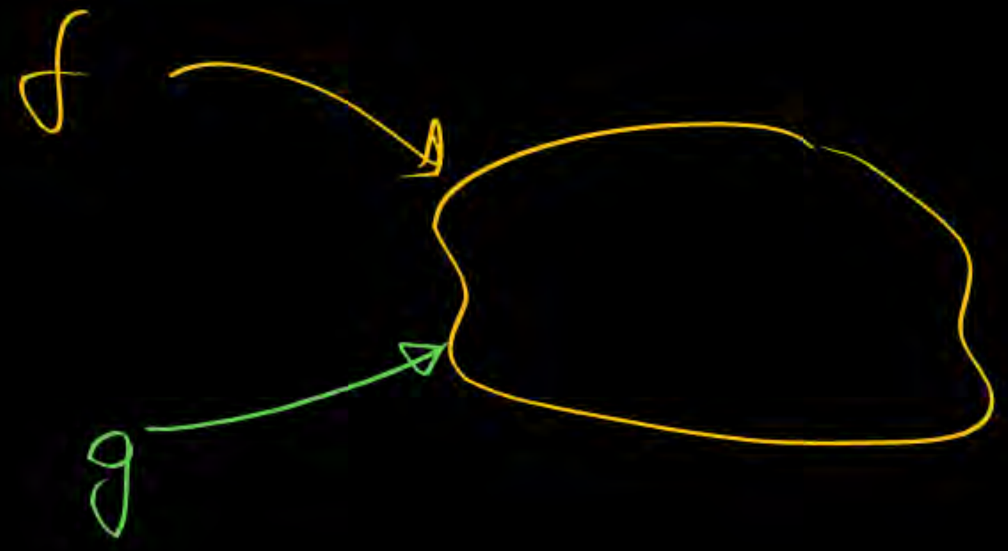
$$\begin{tikzcd}
{[1, 2, 3, 4, 5]} \arrow[r, Rightarrow] \arrow[d, "map"] & 5 \text{ elem} \\
{[1, 8, 27, 64, 125]} \arrow[r, Rightarrow] & 5 \text{ element}
\end{tikzcd}$$

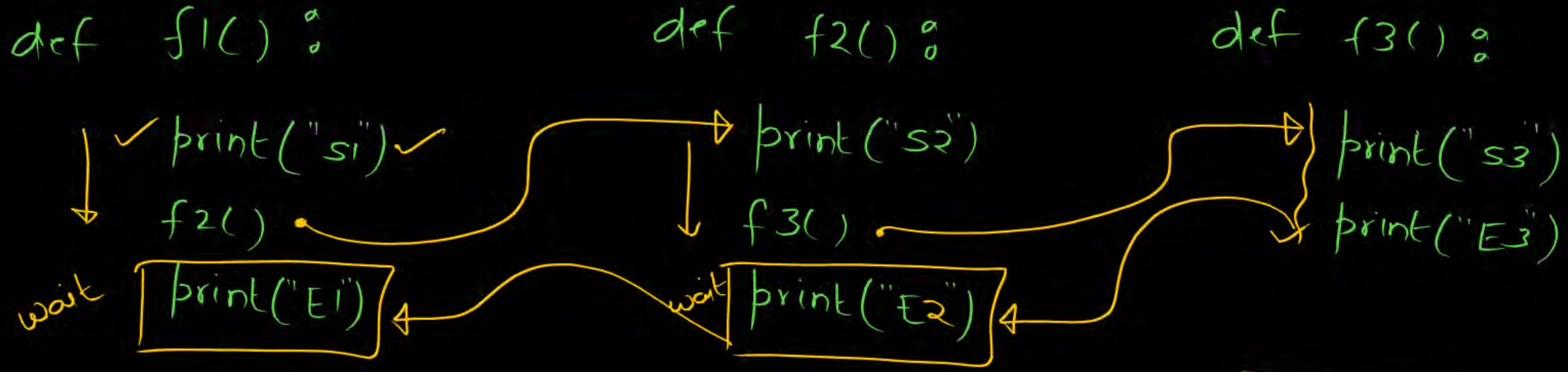
```
def f(x):
    return x*x*x

out = list(map(f, [1,2,3,4,5]))
print(out)
```

Python ⇒ object



```
def f(x):

    return x*x*x

f(10) => 1000

g = f

g(10) => 1000 ✓
g(2) => 8 ✓
```





```
def f1():
    print("s1")
    f2()
    print("E1")
```

```
def f2():
    print("s2")
    f3()
    print("E2")
```

```
def f3():
    print("s3")
    print("E3")
```

```
f1()
```

```
s1
S2
S3
E3
E2
E1
```

Outer func.
inner func. ✓

OOPS

Assume

YT C Programming

CC → 16 videos ✓

Next ⇒ 4 days

Recursion ⇒ video upload

Maths → focus ✓

t.me/PWpankajsirP

In [1]:
```
s=lambda x : x*x*x
```

In [2]:
```
print(s(4))
```

```
64
```

In [3]:
```
def f(a,b):
    return a*b
print(f(10,20))
```

```
200
```

In [4]:
```
#inputs====>2 inputs
f=lambda a,b : a*b
print(f(10,20))
```

```
200
```

In [5]:
```
f(10,30)
```

Out[5]:
```
300
```

In [6]:
```
# lambda arguments : expression
t=lambda x,y :   x if x<y else y #to find minimum among 2 numbers
print(t(10,34))
```

```
10
```

In [8]:
```
def f(x):
    if x%3==0:
        return True
    else:
        return False
output=list(filter(f,[10,20,30,40,50,63]))
```

In [9]:
```
print(output)
```

```
[30, 63]
```

In [10]:
```
out=list(filter(lambda x :x%3==0 ,[10,20,30,40,50,63]))
```

In [11]:
```
out
```

Out[11]:
```
[30, 63]
```

In [12]:
```
def f(x):
    return x*x*x
out=list(map(f,[1,2,3,4,5]))
print(out)
```

```
[1, 8, 27, 64, 125]
```

In [13]:
```
out=list(map(lambda x:x*x*x,[1,2,3,4,5]))
```

In [14]:
```
out
```

Loading [MathJax]/extensions/Safe.js

```
125]
```

In [15]:

```
a=[1,2,3,4,5]
b=[11,12,13,14,15]
c=list(map(lambda x,y:x+y ,a,b))#1st ===>1,11  ==>1+11=12
#2nd time 2,12 ==>14
```

In [16]:

```
c
```

Out[16]:

```
[12, 14, 16, 18, 20]
```

In [19]:

```
def f(x):
    return x*x*x
g=f
```

In [20]:

```
f(2)
```

Out[20]:

```
8
```

In [21]:

```
g(2)
```

Out[21]:

```
8
```

In [22]:

```
g(10)
```

Out[22]:

```
1000
```

In [23]:

```
def f1():
    print("start1")
    f2()
    print("end1")
def f2():
    print("start2")
    f3()
    print("end2")
def f3():
    print("Start3")
    print("End3")
f1()
```

```
start1
start2
Start3
End3
end2
end1
```

In [ ]:

Loading [MathJax]/extensions/Safe.js

THANK - YOU