Devin

Will it take your Job ?

First let us get a Job

2. a] req very clear + You know what to do

⇓

AI — Data entry
basic analytics
no domain knowldge

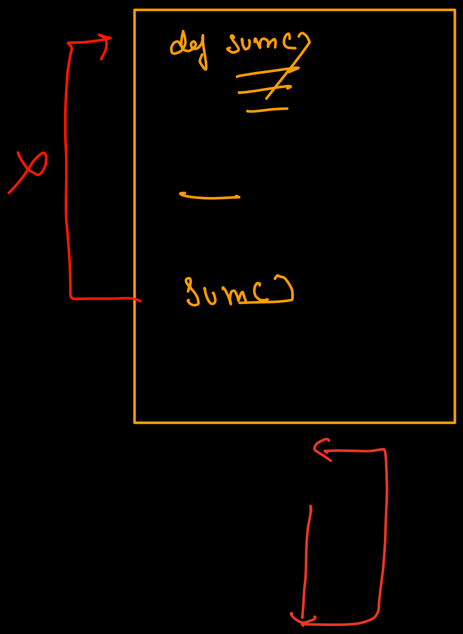
★★ b) Proper coding + req not clear
+ Complex Logic
+ System design

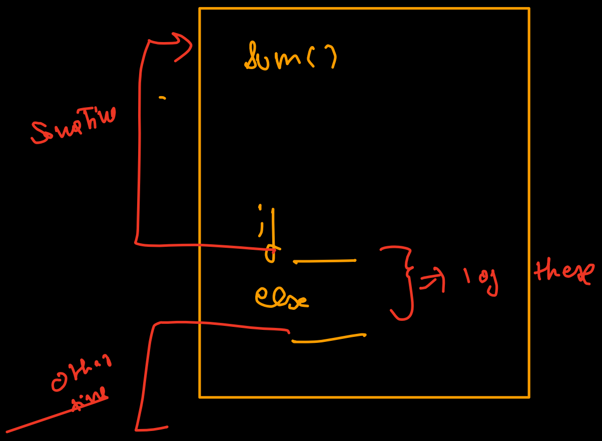
⇓

hype
⇒ AI software dev
{ tbs }
sql

+ Boilerplate
specific requirem

+ SDE3 / EM
★★





Based on some cond^n, we are doing something

Agenda for today

1. Loops / Iteration
   - → While
   - → For
2. Range
3. Some Questions

Need of loop? ⇒ To do repeated steps or task which are exactly same more than one time.

Stud 1 = 53
stud 2 = 55
stud3 = 59

________ stud1 * 100
________ stud 2 * 100
________ stud 3 * 100 }

The issue is that our code unnecessarily becomes long & complex.

⇒ Simple
⇒ As small as possible }

We use loops :-

1] We have to run a cell multiple times

Python have 2 loops

1. While
2. For

do...while
not in python

While Loop

Syntax :-

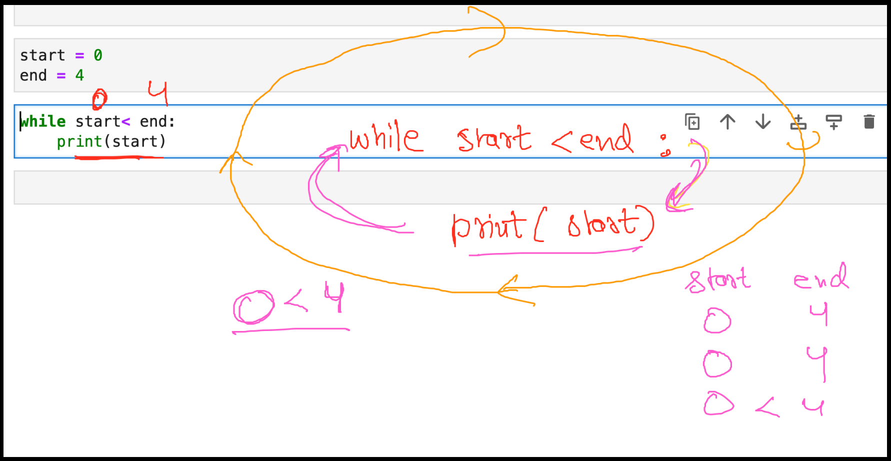
while condition :

logic/
What to do
~~Cond^n check~~

non-indented.

it doesn't end up a ∞ loop

So, make sure to double check & update our condition.



```
start = 0
end = 4
```

```
while start< end:
    print(start)
```

0 4

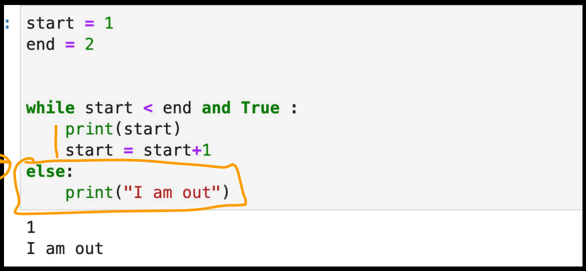
while start < end :

print(start)

0 < 4

start	end
0	4
0	4
0 <	4

Another form of while loop:



```
start = 1
end = 2


while start < end and True :
    print(start)
    start = start+1
else:
    print("I am out")
```

```
1
I am out
```

When my loop all iteration are not finished, then else block won't get executed

Break statement

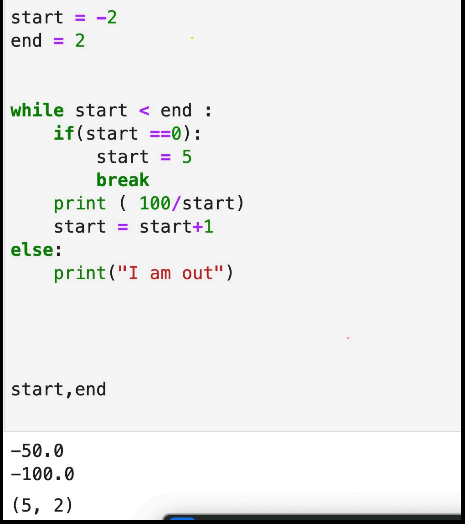
Ideally, loop only closes itself when the condition becomes false

⇓

else gets activated

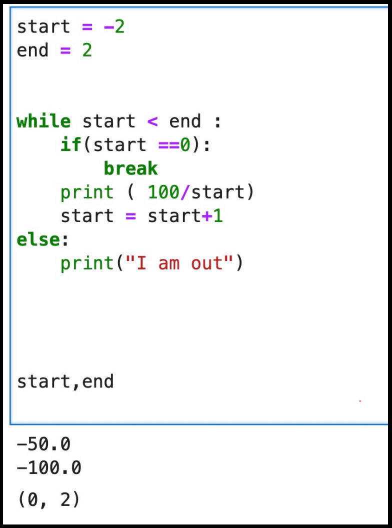
But But But. we can also end our loop before it has all iteration exhausted

For that we have a keyword **break**



```
start = -2
end = 2


while start < end :
    if(start ==0):
        start = 5
        break
    print ( 100/start)
    start = start+1
else:
    print("I am out")




start,end
```

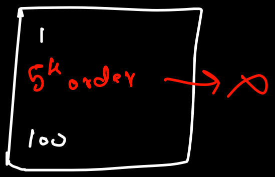
```
-50.0
-100.0

(5, 2)
```



```
start = -2
end = 2


while start < end :
    if(start ==0):
        break
    print ( 100/start)
    start = start+1
else:
    print("I am out")




start,end
```

```
-50.0
-100.0

(0, 2)
```

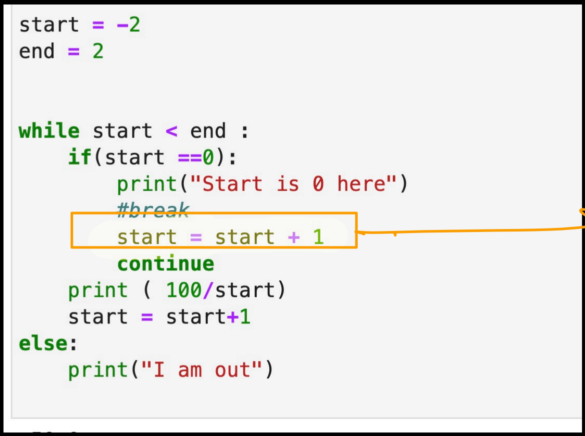
Continue Statement :



```
for (  .    ):
    if notpaid(i):
        break
```

In this case we might just want to avoid this particular order but process all other order afterwords.

Enter "Continue":



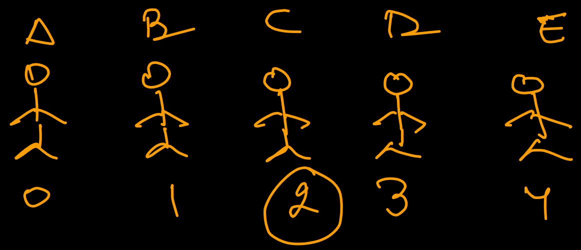
```
start = -2
end = 2


while start < end :
    if(start ==0):
        print("Start is 0 here")
        #break
        start = start + 1
        continue
    print ( 100/start)
    start = start+1
else:
    print("I am out")
```

very very important

For loop

Like while, it is another loop.

Helps to run statements iteratively on collection/sequence



posn

2
3

Name/Student

l = [A, B, C, D, E]
0 1 2 3 4

l[0] A

while i < 5:
____ l[i] ⇒ marks

for stud. in l:
____ stud.

For loop handles increment after a single iteration gets completed

Syntax :-

```
for some_var in collection:
    #loop body
```

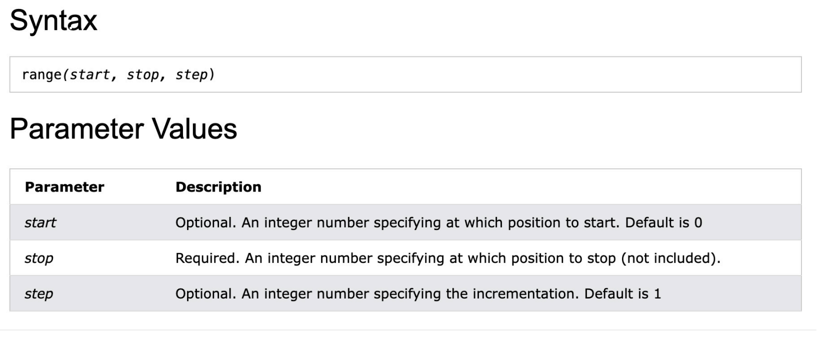
← normal code flow

Range f^n

range ⇒ inbuilt f^n in python

returns us a range object

it is used to generate a sequence of numbers in form of a list



Syntax

```
range(start, stop, step)
```

Parameter Values

Parameter	Description
start	Optional. An integer number specifying at which position to start. Default is 0
stop	Required. An integer number specifying at which position to stop (not included).
step	Optional. An integer number specifying the incrementation. Default is 1