

~~X = 1~~
~~sum = 0~~
 while

X <= 10 :

while loop

for loop

sum += X # sum = sum + X

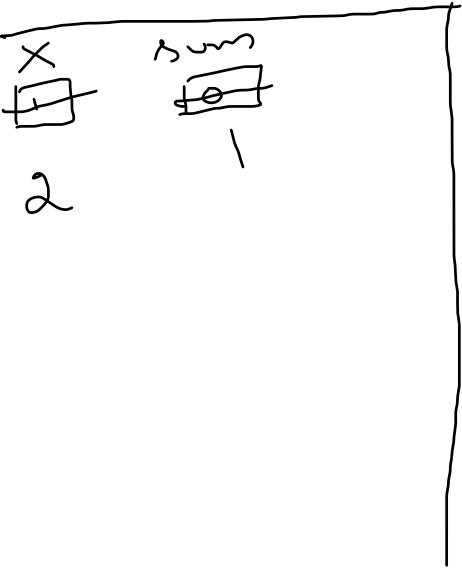
X += 1 # X = X + 1

print("sum = ", sum) = 1 + 1 = 2

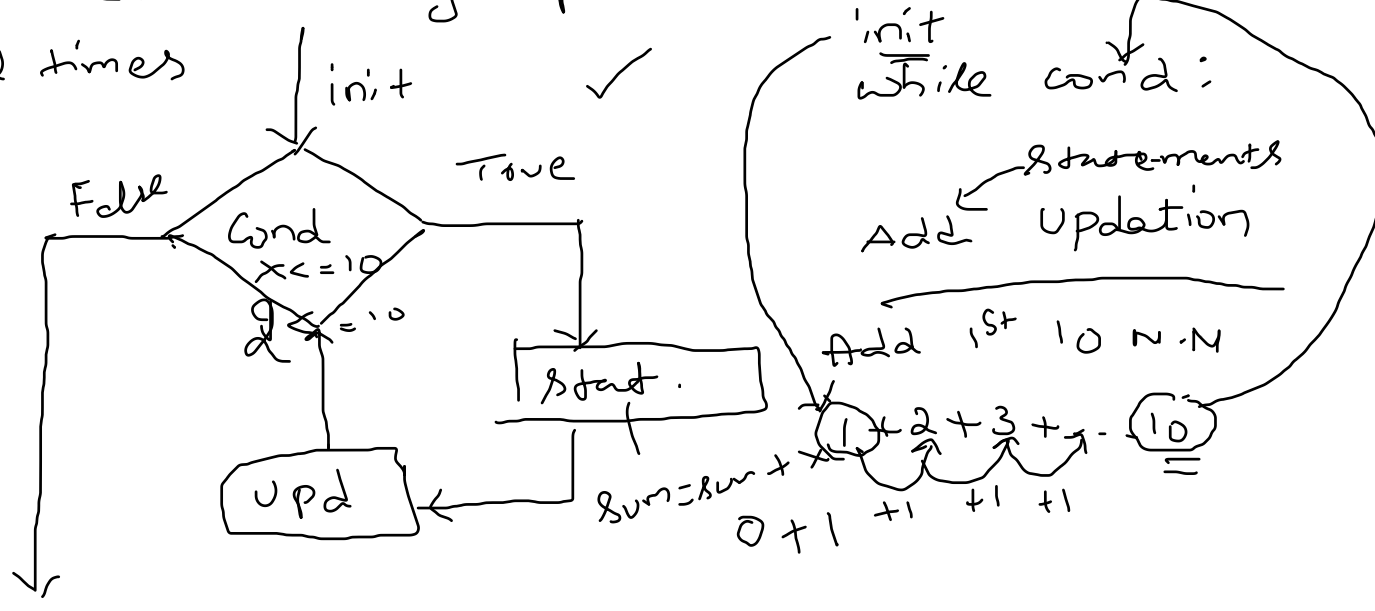
else suite

while loop - used to execute a group of statement(s)

several times



- ① initialization
- ② Condition
- ③ Updation



for loop

useful for iterating over the elements of the sequence

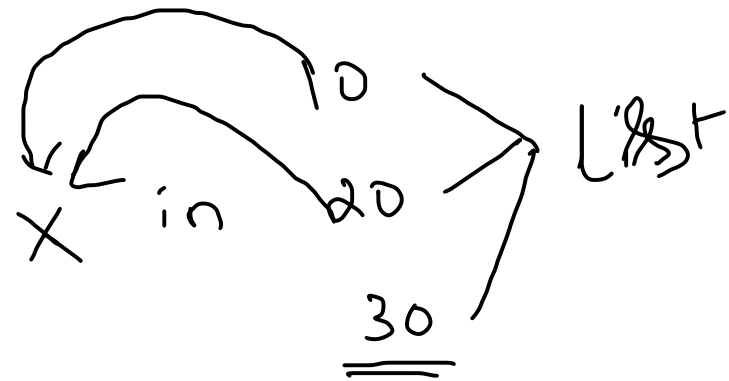
syn for var in sequence:
statements

eg:

lst = [10, 20, 30]

for x in lst:

print(x)



O/P
10
20
30

is 0, 1, -1

range (10, 0, -1)
 ↑ ↓ ↓
 10 9 8 ... 1
 include end. step param

if else

~~enclave~~

else suite

while
while ...
else:
 start

for
for ...
else:
 start

Indentation spaces	Suite St. with same spaces
----------------------------	----------------------------------

else suite will be executed irrespective of one loop statement getting sum or not

even or odd

using while

x = 41

Unconditional st

① break - while or for to
of the loop.

come out

o/p

41 is odd

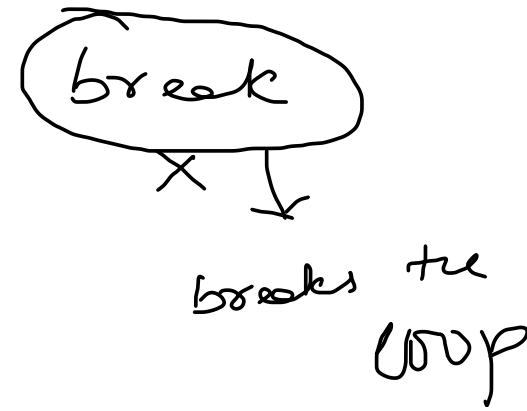
while

1 ✓
2 -
3
4
5
6
7
8
9

if (x == 5)
5 == 5
break

② Continue + used to go back to the beginning of the loop

Continue will execute the next repetition.



~~x~~ = 1
while x <= 10

if ~~x~~ == 5:
Continue

- 1 -
- 2 -
- 3 -
- 4 -
- 5 -
- 6
- 7
- 8
- 9
- 10

SKIP



③ pass — if or any loop, which does nothing

if $x == 5$:

pass →

④ assert — check for a particular condition is
met or not

syn

assert expression: message

input()

→ used to take the input from
console

gt

assert $x > 0$, 'wrong input'

Control

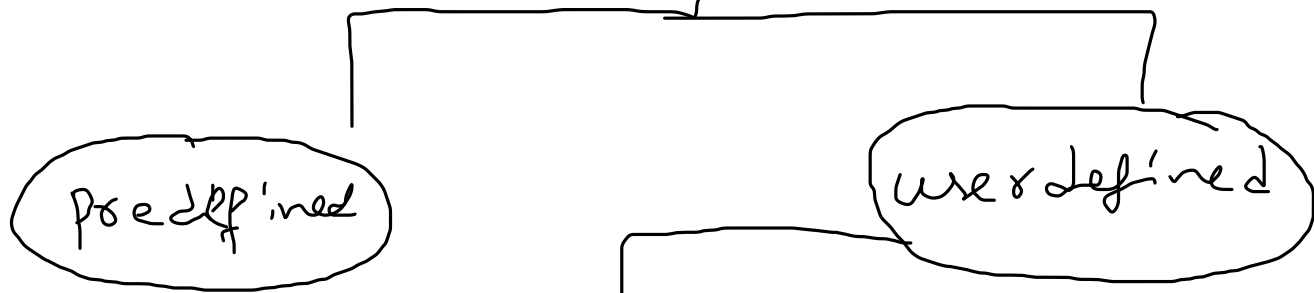
Python

API

Application programming Docu

R 2 u D

Functions



Syntax

```
def functionname (P1, P2, ..., Pn)
    """ docstring """
    Statement Suite
```

input
print
type
id

sum

```
def sum():
    a = 5
    b = 2
    c = a + b
    print(c)
```

① sum()

↓

7

```
def sum(a, b):
    c = a + b
    print(c)
```

sum(5, 2)

↓

7

Virtual

or logical

return understood. Stat

int float

type ()
id

spyder

- function is
- ① Component a program ✓
 - ② Performs a predefined task ✓
 - ③ It may take data ✓
 - ④ It may return data ✓
 - ⑤ They must be called ✓

```
def sum(a, b):
    c = a + b
    return c
```

x = sum(5, 2)

print(x)

x = 7

CS+

-

True

True

TRUE

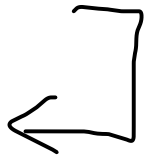
False

true

```
X = int(input("Enter any no"))
```

```
assert (X > 0,  
        S2 > 0), 'wrong input'
```

```
Print(x)
```



Enter any no

52

Assertion Error, wrong input