1) Jump Statements
   1) Pass
   2) Continue
   3) break

2) Nested Loops

3) GCD

4) LCM

Good Code Properties

1) Security

2) Understand and Maintainable
   {
   1) → naming Conventions
   2) → Rules and Standards
   3) → Document and give proper Comments
   → modularity and Portable
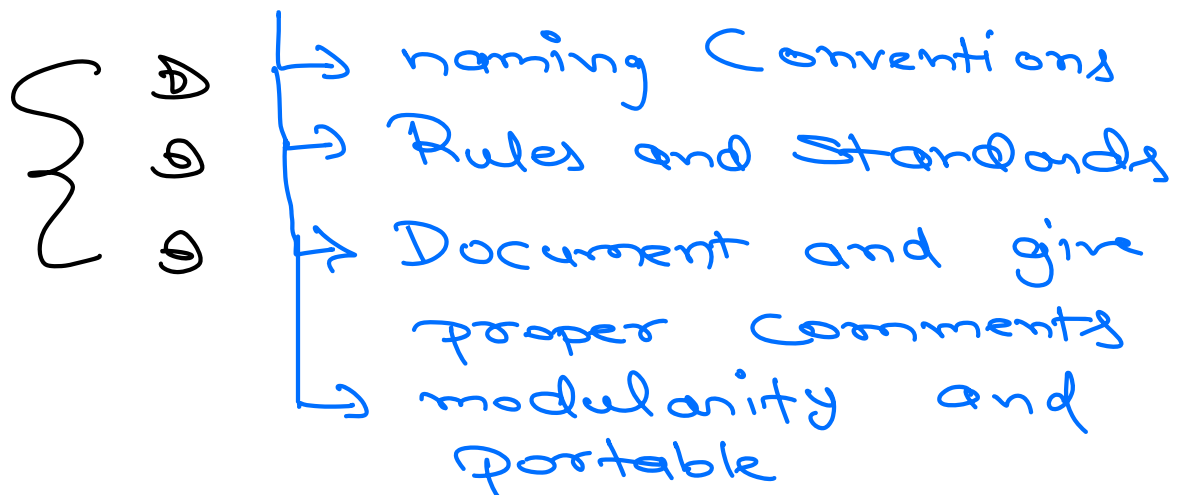
3) Time and Space Efficiency
                    Complexity

- Pass acts as a placeholder
- You can use this to create empty blocks for later revision

Continue

- Skips iteretation

↓

All the code written after continue will be ignored

for i in range (S, E, J) :
_____
_____
_____
(Continue)
_____ Line 5
_____ Line 6

Let's say @ $i = 3$ we visit continue

for i in range (5):
_____
_____

i = 0
i = 1
i = 2
i = 3
i = 4

[Code will enter for loop 5 times]

S = 0
J = 1

Break

for i in range (S, E, J):
_____
_____
_____

break

_____ Line 5
_____ Line 6

→ this code is outside of For Loop

Note: Break exits the loop

Nested Loop

Loop inside Loop

```
for i in range(3):
    for j in range(2):
        print(i, j)
```

i=1

i=0 → j=0
     j=1

(0, 0)   ①
(0, 1)   ②

i=1 → j=0
     j=1

(1, 0)   ③
(1, 1)   ④

i=2 → j=0
     j=1

(2, 0)   ⑤
(2, 1)   ⑥

i x j
3 x 2

for i in range(2):
    for j in range(3):
        print(i, j)

resetted for every i

$i = 0$
$j = 0$ → (0, 0)
$j = 1$ → (0, 1)
$j = 2$ → (0, 2)

$i = 1$
$j = 0$ → (1, 0)
$j = 1$ → (1, 1)
$j = 2$ → (1, 2)

Q) for i in range(3)
    for j in range(i, 3):

i	j

$i = 0$ → $j(0, 3)$
0, 0
0, 1
0, 2 → 3 j's

$i = 1$ → $j(1, 3)$
1
1
2 → 2 j's

$i = 2$ → $j(2, 3)$
2 2 → 1 j's

GCD

16, 24 → 1
→ 2
→ 4
→ 8

min (num1, num2)

for loop on min.

Check 'if 'i' divides both

$\frac{2|6}{3}$

$\frac{2|8}{\frac{2|2}{1}}$

GCD
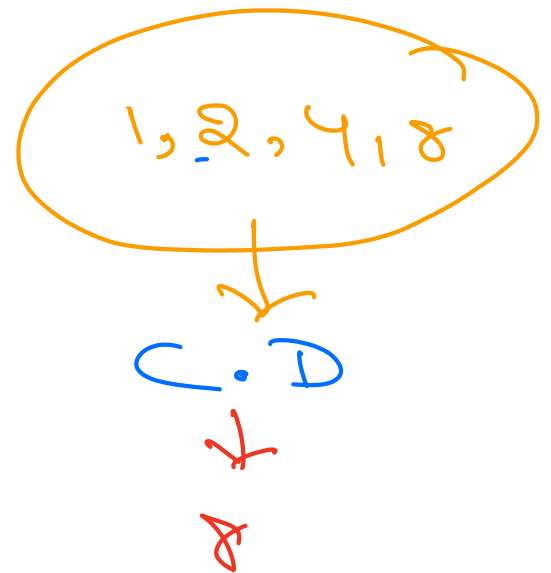
$$n = 16$$

$$m = 24$$

Largest
Is ^ number that divides both
16 and 24

$$16 \% 8 \rightarrow 0$$

$$24 \% 8 \rightarrow 0$$

1, 2, 4, 8

G.D

8

LCM

Least Common Multiple

$n = 6$
$m = 8$ ← max (8)

$32 \% 6 \, \cancel{x}$

$\boxed{24 \% 6} \, ?$
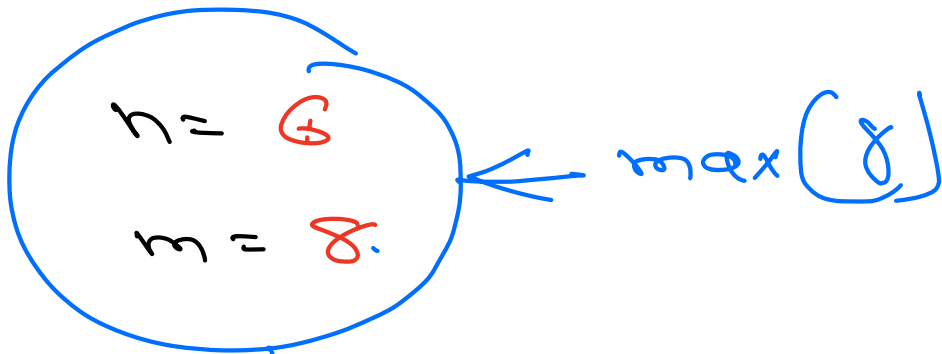
$i \% n == 0$
and
$i \% m == 0$

$24 \% 6 == 0$
$24 \% 8 == 0$

$[\boxed{24}, 48]$

48

$CM \,]$

$i = 8$

$= 16$

$= 24$

$i \% m == 0$ break
$i \% n == 0$

1 0 1

↑

digits = 3
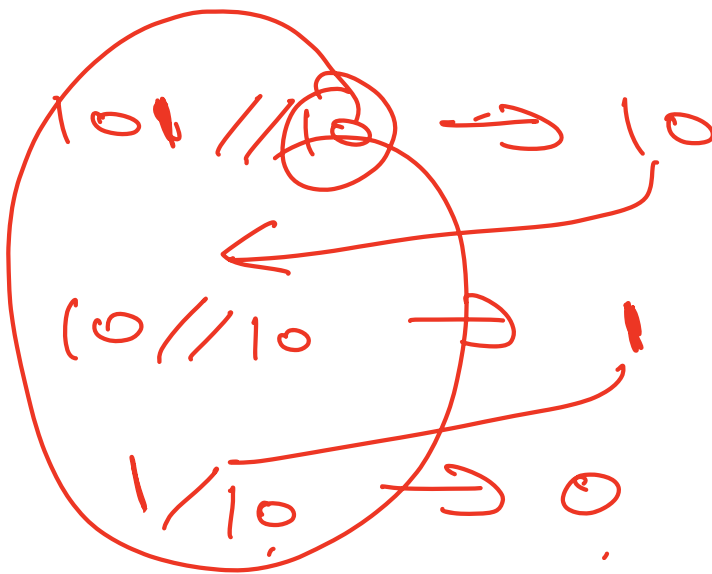
Count = 0

Count += 1

Count = 2

Count = 3

10 1 // 10 → 10

(0 // 10 → 1

1 / 10 → 0

while 2

1 0 0 1 0 1 0

1 0 0 0 1