

* for Loop :

```
for (initialization; condition; operation) {
```

/* code

*/



```
352 | for (let num = 1; num <= 5; num++) {  
353 |   console.log(`Level ${num}`);  
354 | }
```

① Initialization happens first then we try to enter the loop. But you can only enter if the condition is True.

② operation $\xrightarrow{\text{condition true}}$
repeats until condition is false.

* num = 1 \neq 2 \neq 3 \neq 4 \neq 5 \neq 6 \leq 5

op: Level 1

Level 2

Level 3

Level 4

Level 5

③ Initialization, let num = 1;

④ check condition before entering,

num \leq 5 \Rightarrow 1 \leq 5 \Rightarrow true

\Rightarrow cl ("Level 1")

num

1

2

3

4

⑤ operation, num++ \Rightarrow num = 2

⑥ num \leq 5 \Rightarrow 2 \leq 5 \Rightarrow true

\Rightarrow cl ("Level 2")

⑦ operation, num++ \Rightarrow num = 3

⑧ num \leq 5 \Rightarrow 3 \leq 5 \Rightarrow true

\Rightarrow cl ("Level 3")

⑨ operation, num++ \Rightarrow num = 4

⑩ num \leq 5 \Rightarrow 4 \leq 5 \Rightarrow true

\Rightarrow cl ("Level 4")

⑪ operation, num++ \Rightarrow num = 5

⑫ num \leq 5 \Rightarrow 5 \leq 5 \Rightarrow true

\Rightarrow cl ("Level 5")

⑬ operation, num++ \Rightarrow num = 6

⑭ num \leq 5 \Rightarrow 6 \leq 5 \Rightarrow false

————— Stop Loop —————

operation

$num = 1 \quad \cancel{+1} \quad \cancel{+1} \quad \cancel{+1}$

~~Init~~ ~~Condition~~ ~~num ≤ 5~~

~~num $\leftarrow num + 1$~~

operation

$num = 5 \quad \cancel{-1} \quad \cancel{-1}$

~~Init~~ ~~Condition~~ ~~num > 0~~

~~num $\leftarrow num - 1$~~

```
for (let num = 1; num  $\leq 5$ ; num++) {
    console.log(`Level ${num}`);
}
```

$num \geq 1$

(a)

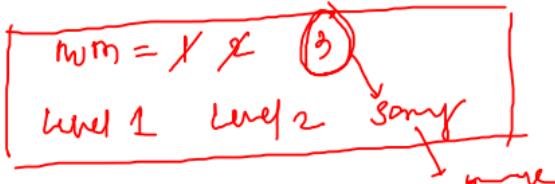
```
for (let num = 5; num > 0; num--) {
    console.log(`Level ${num}`);
}
```

break statement :

→ terminate / come out of the current loop in execution.

```
368 for (let num = 1; num <= 5; num++) {  
369   if (num == 3) {  
370     console.log("Sorry you cannot proceed to next level");  
371     break;  
372   }  
373   console.log(`Level ${num}`);  
374 }  
375 }
```

- ⑥ $\text{num} \leq 5, 3 \leq 5 \Rightarrow \text{true}$
a. $\text{num} == 3, 3 == 3 \Rightarrow \text{true}$
→ "Sorry you"
→ **break**



- ① $\text{num} = 1$
- ② $\text{num} \leq 5, 1 \leq 5 \Rightarrow \text{true}$
 - a. $\text{num} == 3, 1 == 3 \Rightarrow \text{false}$
 - b. $\text{cl}(\text{"Level 1"})$
- ③ operation, $\text{num}++ \Rightarrow \text{num} = 2$
- ④ $\text{num} \leq 5, 2 \leq 5 \Rightarrow \text{true}$
 - a. $\text{num} == 3, 2 == 3 \Rightarrow \text{false}$
 - b. $\text{cl}(\text{"Level 2"})$
- ⑤ $\text{num}++ \Rightarrow \text{num} = 3$

Continue statement :

→ move to the next iteration of current loop by ignoring all the code below it. (skip)

```
379 for (let num = 1; num <= 5; num++) {  
380   if (num == 3) {  
381     console.log("You can skip this level");  
382     continue;  
383   }  
384   console.log(`Level ${num}`);  
385 }  
386 console.log("outside for loop");
```

- ① num = 1
- ② 1 <= 5 ✓
 - a. 1 == 3 X
 - b. "Level 1"
- ③ num++ \Rightarrow num = 2
- ④ 2 <= 5 ✓
 - a. 2 == 3 X
 - b. "Level 2"
- ⑤ num++ \Rightarrow num = 3
- ⑥ num <= 5, 3 <= 5 ✓
 - a. 3 == 3 ✓
 - "Skip Level..."
 - Continue
- ⑦ num++ \Rightarrow 4 <= 5
 - a. 4 == 3 X
 - b. "Level 4"
- ⑧ num++ \Rightarrow 5 <= 5
 - a. 5 == 3 X
 - b. "Level 5"
- ⑨ num++ \Rightarrow 6 <= 5 X

★ Sum of natural numbers :

ip: $N = 5$

op: $1+2+3+4+5 = 15$

ip: $N = 10$

op: $1+2+3+4+5+6+7+8+9+10$
 $= 55$

$nwm = 1 \rightarrow \text{Sum} + = 1 \quad \begin{matrix} 0 \\ +1 \end{matrix}$
 $2 \rightarrow \text{Sum} + = 2 \quad \begin{matrix} +2 \end{matrix}$
 $3 \rightarrow \text{Sum} + = 3 \quad \begin{matrix} +3 \end{matrix}$
 $4 \rightarrow \text{Sum} + = 4 \quad \begin{matrix} +4 \\ +5 \end{matrix}$
 $5 \rightarrow \text{Sum} + = 5 \quad \underline{\begin{matrix} +5 \\ 15 \end{matrix}}$
 $6 \Rightarrow \text{stop}$

let sum = 0;

for(let nwm = 1; nwm <= N; nwm++) {

 sum += nwm; // sum = sum + nwm;

}

① $nwm = 1$

sum = $\cancel{0} \neq \cancel{1} \cancel{2} \cancel{3} \cancel{4} 15$

② $1 <= 5,$

sum += 1

③ $2 <= 5$

sum += 2

⑥ $5 <= 5$

④ $3 <= 5$

sum += 3

⑤ $4 <= 5$

⑦ $6 <= 5 \times$

⑥ $5 <= 5$
 sum += 4

ff formula:

Sum of 1st N natural numbers

$$= \frac{N * (N+1)}{2}$$

Ex: $N = 10$
 $\Rightarrow \frac{10 * 11}{2}$
 $\Rightarrow 55$

* always $N = \text{Number}(N)$
Ex: $N = '10'$

$$\begin{aligned} & \frac{N * (N+1)}{2} \quad S + N \Rightarrow S + N \\ \Rightarrow & \frac{'10' * ('10' + 1)}{2} \quad '10' + '1' \Rightarrow '101' \\ \Rightarrow & \frac{'10' * '101'}{2} \\ \Rightarrow & \frac{10 * 101}{2} \Rightarrow 505 \end{aligned}$$

* Even sum :

q: $N = 12$

op: $2 + 4 + 6 + 8 + 10 + 12 + \cancel{14}$
= 42

operation

$nwm = 2 \times 2 + 2 \times 2 + 2 \times 2$
init
condition

let sum = 0;

for(let nwm=2; nwm <= N; nwm+=2) {

sum += nwm;

}

cl(sum);

- | | | |
|---|--------------------------------|----|
| ① | $nwm = 2 \Rightarrow sum += 2$ | 0 |
| ② | $nwm = 4 \Rightarrow sum += 4$ | +2 |
| ③ | $nwm = 6 \Rightarrow sum += 6$ | +4 |
| ④ | $nwm = 8 \Rightarrow sum += 8$ | +6 |
| | ⋮ | ⋮ |
| | ⋮ | ⋮ |

⋮
⋮
⋮
⋮
⋮

~~# formula~~

* do not apply formula for β_3 , instead apply for β_2
eg 24

$$\frac{n * (n + 2)}{4}$$

$$\begin{array}{r} 3 \\ \times 16 \\ \hline 48 \end{array}$$

$$\begin{array}{r} 6 \\ \times 26 \\ \hline 156 \end{array}$$

$$= 42$$

~~$-2 + 4 + 6 + 8 + 10 + 12$~~ $\left(2 + 4 + 6 + 8 + 10 + 12 + \dots + 20 + 22 + 24 \right)$

Eg: 13

~~tg: 25~~

$$\begin{array}{r} 13 * 15 \\ \hline 4 \end{array}$$

$$\begin{array}{r} 25 * 87 \\ \hline 4 \end{array}$$

wrong

* the formula fails for odd numbers.

$\beta_2 \leftrightarrow \beta_3$ are same

$24 \leftarrow 25$ are same

even \leftrightarrow even +/ and same

* When n is odd,

$$h = n - 1; \quad (n - 1);$$

* Check prime :

→ divisible by 1 and itself

e.g.: $n=5$ e.g.: $n=13$
op: true op: true

Q: 49 is prime?

~~[2, 48]~~

$$\begin{array}{ll} 49 \div 2 == 0 \times & 49 \div 5 == 0 \times \\ 49 \div 3 == 0 \times & 49 \div 6 == 0 \times \\ 49 \div 4 == 0 \times & 49 \div 7 == 0 \checkmark \end{array}$$

⇒ I found a number (7) that divides 49, hence it is not a prime.

⇒ 8-48 checks are ignored

Q: check 13 is prime or not?

~~2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13~~
[2, 12] → Is there any number that divides 13?

$$\begin{array}{ll} 13 \div 2 == 0 \times & 13 \div 7 == 0 \times \\ 13 \div 3 == 0 \times & 13 \div 8 == 0 \times \\ 13 \div 4 == 0 \times & 13 \div 9 == 0 \times \\ 13 \div 5 == 0 \times & 13 \div 10 == 0 \times \\ 13 \div 6 == 0 \times & 13 \div 11 == 0 \times \\ 13 \div 12 == 0 \times \end{array}$$

⇒ I have tried and failed to find a num which divides 13, hence it is a prime.

→ Convert to code;

$$N \Rightarrow [2, N-1]$$

$$B \Rightarrow [2, 12] \quad 49 \Rightarrow [2, 48]$$

$nwm < n$
(a)
for (let nwm = 2; nwm <= n-1; nwm++) {

if ($n \% nwm == 0$) {
 console.log("Not a prime number");
 break;
}

}

else {

~~console.log("Prime"); break;~~

}

* you are looking at ^{only} one check
 $n \% nwm == 0$
and deciding prime/not.
do not decide inside for loop.

49

① $nwm = 2,$

if (~~$49 \% 2 == 0$~~) {

~~3~~

else {

 prime ↗

3

⇒ But 49 is not prime

* Flag variable Technique :

Currently assume that
N is prime number.

* Please do not take
any decisions in for loop.

let flag = true;

```
for (int num = 2; num <= n-1; num++) {
```

```
    if (num % num == 0) {
```

```
        flag = false;
```

```
        break;
```

```
}
```

```
}
```

```
if (flag == true) → "prime" } take decision after
```

```
else → "not a prime"
```

} try to
prove that
n is not
a prime

} take decision after
the entire process

* HCF of two numbers : (highest common factor / greatest common divisor)
(HCF/GCD)

e.g.: 75, 90

→ find a number which divides both 75, 90

properties of such number :

① 75% num = 0 and 90% num = 0

② num $\leq \min(75, 90)$

e.g.: 75% 80 will never be = 0

75% num != 0 & num > 75

③ num is largest possible

find HCF(75, 90) ?

1, 2, 3, 4, 5, 6, 7, 8, 9,

....., 50, 51, 52, 53,

....., 60, 61, 62, ..

....., 73, 74, 75

$$\checkmark 75 \% (1) = 0, 90 \% (1) = 0$$

$$\checkmark 75 \% (3) = 0, 90 \% (3) = 0$$

$$\checkmark 75 \% (5) = 0, 90 \% (5) = 0$$

$$\boxed{75 \% (15) = 0, 90 \% (15) = 0}$$

HCF

HCF(75, 90) ?

let ans = 1;

for(let num = 1; num <= min(75, 90); num++) {

if (75 % num == 0 && 90 % num == 0) {

ans = num;

}

}

cl(ans);

H/W

print 1 to 10

factorial with loop

print sum 10

power of number
(using loop)

for(let num = min(75, 90); num >= 1; num--) {

if (75 % num == 0 && 90 % num == 0) {

console.log(num);

break;

;

;

* the first common
factor from back
will be highest