loops

Increment & Decrement

The increment and decrement operators in JavaScript will add one (+1) or subtract one (-1).

Increment	Decrement

`var a = 5; a = a + 1;`
`console.log(a);`
	`var a = 5; a = a - 1;`
	`console.log(a);`

```
//Increment

let x=5;
x=x+1;
console.log(x); // 6 (Incrementing the value x by 1)

let y=6;
y=y+4;
console.log(y); // 10 (Incrementing the value of y by 4)

//Decrementing

let x=5;
x=x-1;
console.log(x); // 4 (Deccrementing the value x by 1)
```

```
let y=6;
y=y-4;
console.log(y); // 2 (Deccrementing the value of y by 4)
```

```
let x=10;
x=x+1;
x=x+5;
x=x+10;
console.log(x);
x=x-10;
x=x-5;
console.log(x);
```

Javascript provides another way of incrementing and decrementing the variable i.e

`++`/`--`.

Using ++/-- After the Operand

- When you use the increment/decrement operator after the operand, the value will
  be returned before the operand is increased/decreased.
- In simple terms, Postpone the operation for later instead of it first printing it.
- This is known as Postfix Increment/Decrement.

Increment	Decrement

`var a = 1`
`console.log(a++);`
`console.log(a).`
	`var a = 1;`
	`console.log(a--);`
	`console.log(a);`

Using ++/-- Before the Operand

- When you use the increment/decrement operator after the operand, the value will
  be increased/decreased returned before the operand is returned.
- Preponing the operation.
- This is known as Prefix Increment and decrement.

Increment	Decrement

`var a = 1;`
`console.log(++a);`
`console.log(a).`
	`var a = 1;`
	`console.log(--a);`
	`console.log(a);`

Examples of Prefix and Postfix

	Prefix	Postfix

Increment	`var a = 1;`
	`console.log(++a);`
	`console.log(a).`
		`var a = 1;`
		`console.log(a++);`
		`console.log(a);`

Decrement	`var a = 1;`
	`console.log(--a);`
	`console.log(a).`
		`var a = 1;`
		`console.log(a--);`
		`console.log(a);`

Student Task

Code 1: Predict the output

```
var a = 10;
++a;
var b = 10;
b++;
```

```
console.log(a)
console.log(b);
```

Code 2: Predict the output

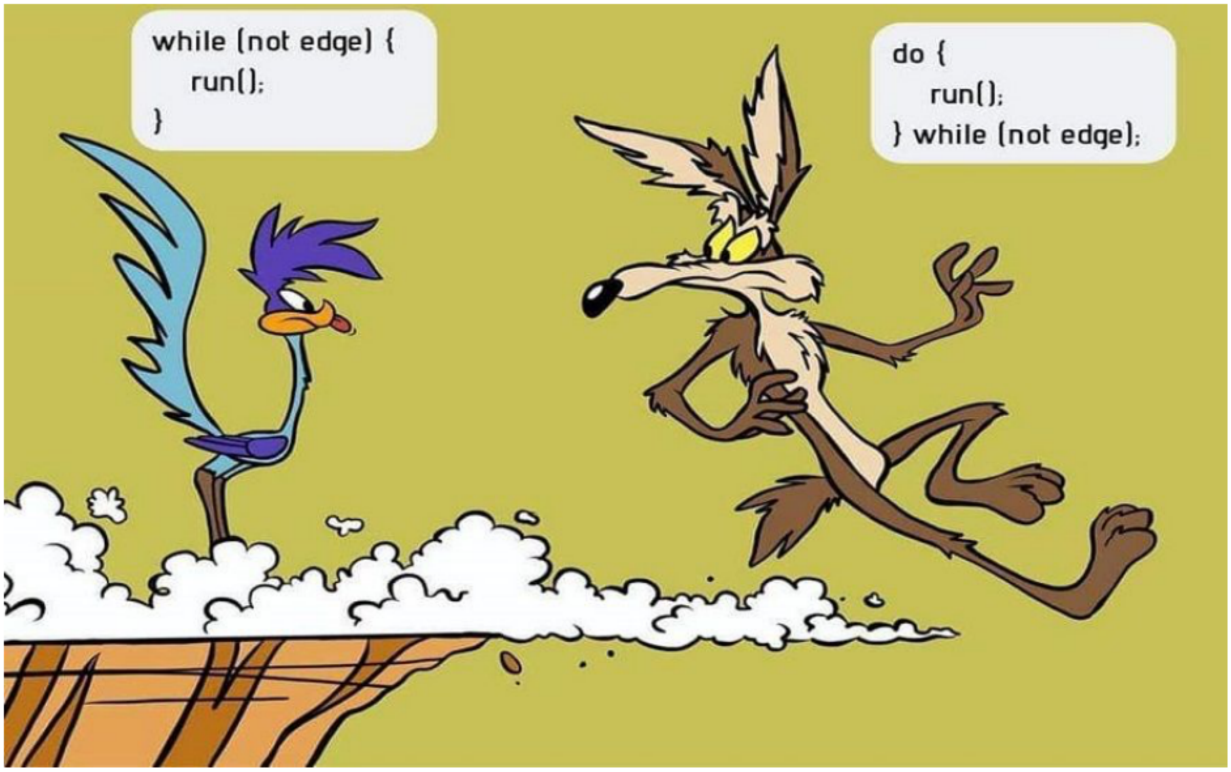
```
var a = 10;
var b = 10;
++a;
b++;
console.log(a);
console.log(b);
```

While loop

student task1- Link

```
var a = 10;

// run a while loop
// wite a condition and print a number from 10 to 20 in seperate

while(){
  //code
  console.log();
  // incerement by 1
}
```

solution- Link



Song Library

On Internet, When we listen to a song. There is an option of listening to the song in the loop, it will play the song again and again when it reaches to end.

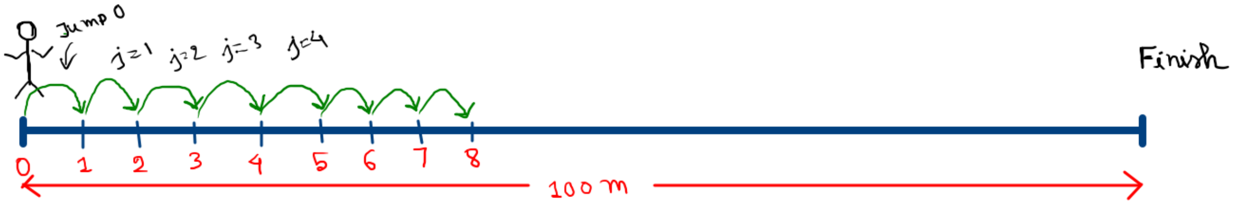
Guests

- There are 10 guests coming to my home, After 2-3 days they decided to leave their home.
- They all have the train on the same day and at the same time.
- I need to drop them at the railway station but I have one bike which can only take one person at a time.
- In this case, I need to drop each guest one by one.
- Taking the First guest to the railway station, dropping them off, and arriving back and following the same procedure again and again till the end.

While Loop

- The while loop begins with a condition and it is written similarly to an if statement. The
  inner parenthesis is the condition.
- As long as the condition is true, it will continue to execute the statement(s).
- To stop the loop, the condition must eventually become false.
- A common condition is to have a variable be less than or greater than compared to
  a number.
- Within the statements, that variable will be incremented or decremented depending
  on the condition.
- Each time the loop is executed, the variable will change and eventually become
  larger or less than the number in the condition, stopping the loop.

Let's try to understand the Loop Variables: Marathon Analogy

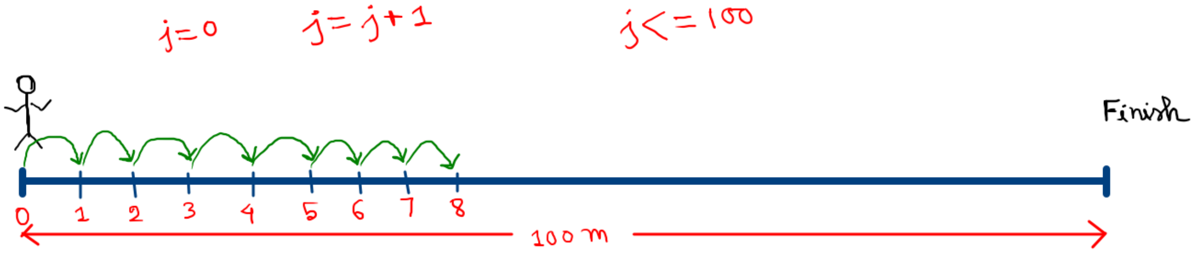


Case 1 : Given a track of 100 m, Hari trains himself for a long Marathon of 100 meter. Hari standing at the 0th position and he needs to cover a 100-meter distance.

Hari will make 1-meter jump at a time.
Let's take Hari's position as position = 0

After 1st jump, Hari will be at position = 1 meter
After 2nd jump, Hari will be at position = 2 meter
....
After the 100th jump, Hari will be at position = 100 meter

Observation



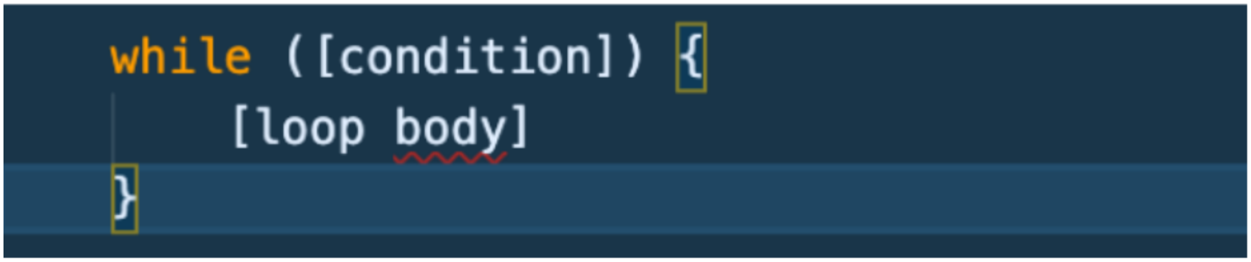
the value of position is going as follows position = 0,1,2,3......100

1. The initial value of position = 0
2. The loop is running till position ≤ 100
3. At every point, Hari is making a jump of 1

For Loops, 3 things are important :

1. **Starting Point:** position = 0
2. **How long jump:** Jump of 1 meter
3. **Till When:** position≤100 meter
   Syntax of While Loop

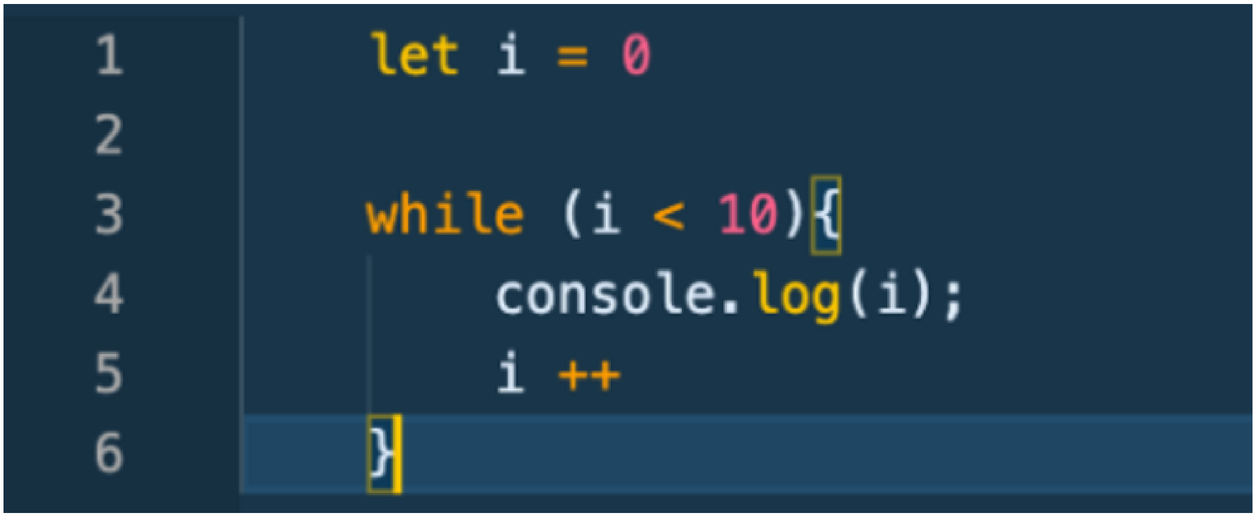
Syntax of While Loop



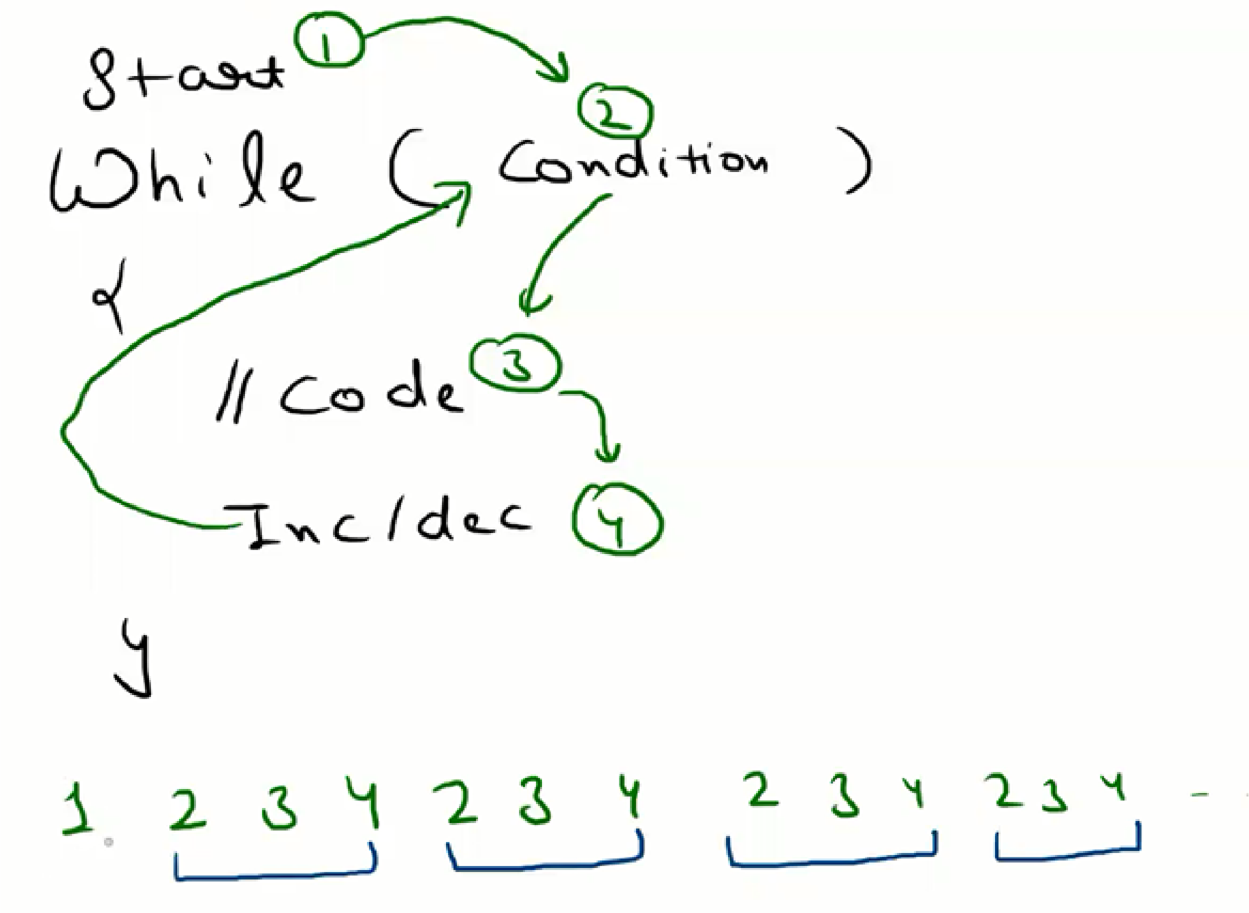
```
while ([condition]) {
    [loop body]
}
```

```
Starting Point
While ( Till When )
{
How long Jump at a time?
}
---------------------------------------------------------------
Initialization
While ( Condition )
{
Increment/Decrement
}
```

```
var position = 0
while(position <= 100)
{
position = position + 1;
}
```

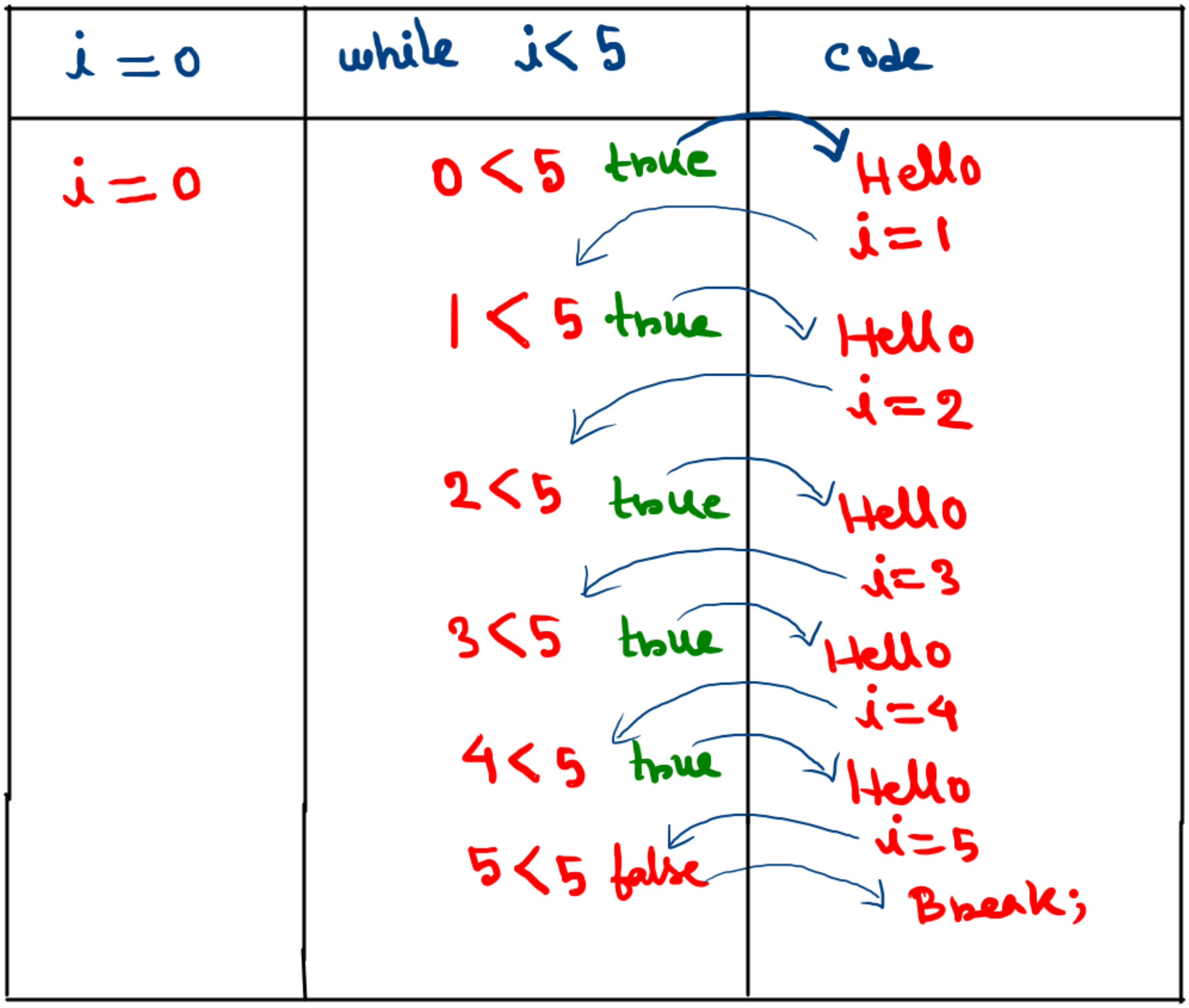


```
    let i = 0

    while (i < 10){
        console.log(i);
        i ++
    }
```



Understanding DRY RUN

```
var i = 0;
while(i<=5)
{
console.log("hello");
i = i + 1;
}
```

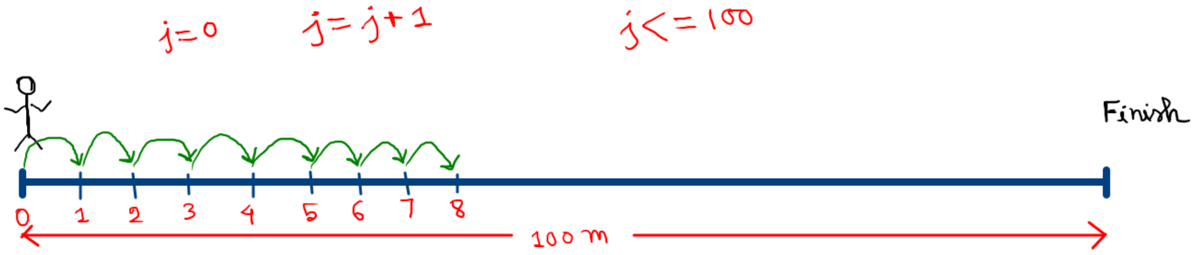


i =0	while i< 5	code
i=0	0<5 true	Hello i=1
	1<5 true	Hello i=2
	2<5 true	Hello i=3
	3<5 true	Hello i=4
	4<5 true	Hello i=5
	5<5 false	Break;

While Loop Examples

Code 1: Loop from 1 to 100 [1-meter jump at a time

```
var position = 0;
while(position <= 100)
{
position = position + 1;
console.log("Current Position ",position);
}
```

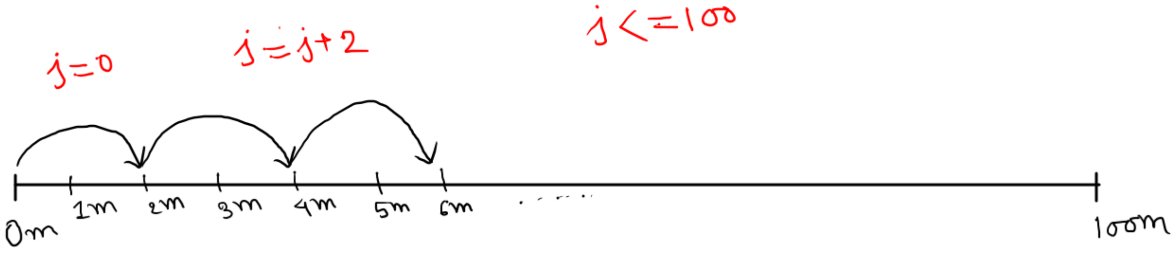


Code 2: Infinite Loop

```
while(true)
{
console.log("Hello Masai");
}
```

Code 3: Loop from 1 to 100 [2-meter jump at a time]

```
var position = 0;
while(position<100){
console.log("Current Position ",position);
position = position + 2;
}
```



Student Task

Code 4: Loop from 1 to 100 [15-meter jump at a time]

```
var position = 0;
while(position<100){
console.log("Current Position",position);
position = position + 15;
}
```

Note : `!=` and `<` behave differently

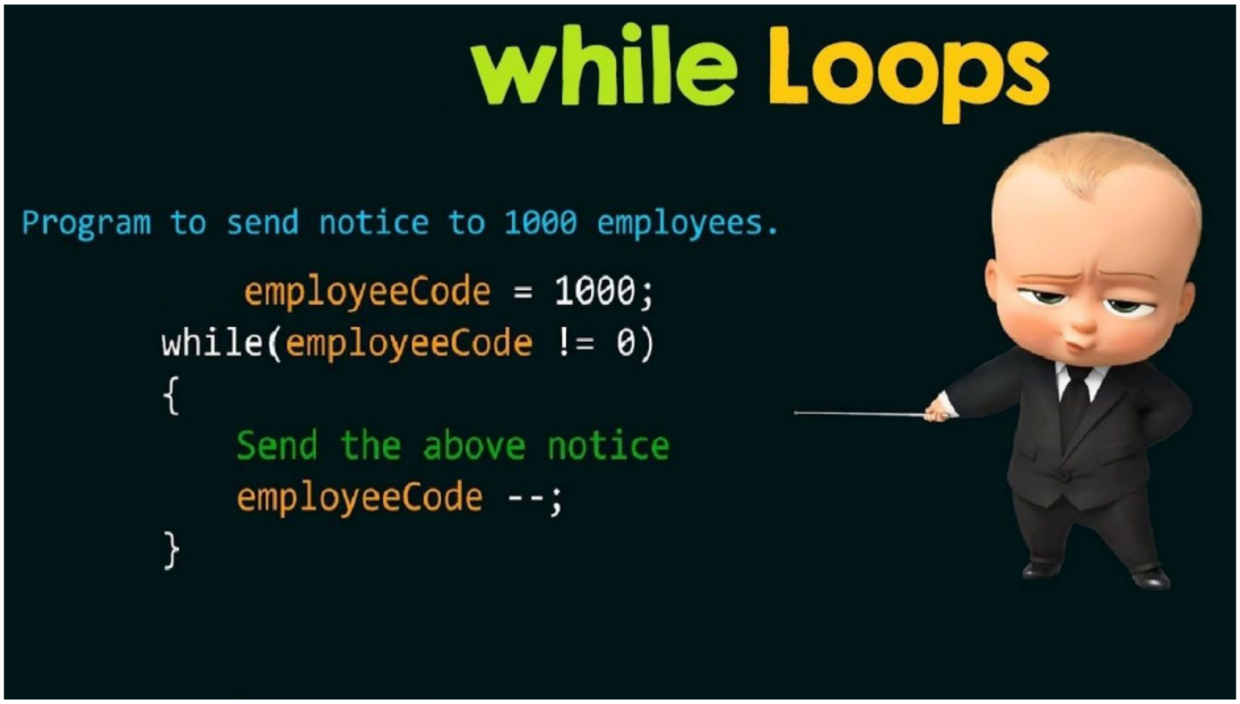
Code 5: Loop from 35 to 100 [3 units jump at a time]

```
var position = 35;
while(position<100){
console.log("Current Position",position);
position = position + 3;
}
```

Code 6: Reverse Loop from 100 to 0 [1 unit jump at a time]

```
var position = 100;
while(position>=0){
console.log("Current Position",position);
position = position - 1;
}
```

Sending Notice to 1000 Employees



while Loops

Program to send notice to 1000 employees.

```
        employeeCode = 1000;
    while(employeeCode != 0)
    {
        Send the above notice
        employeeCode --;
    }
```

Break

Guest Analogy

- There are 10 guests coming to my home, After 2-3 days they decided to leave their home.
- They all have the train on the same day and at the same time.

- I need to drop them at the railway station but I have one bike which can only take
  one person at a time.
- In this case, I need to drop each guest one by one.
- Taking the First guest to the railway station, dropping them off, and arriving back and
  following the same procedure again and again till the end.
  Suppose I took the First Guest and dropped him to the Railway station and come
  back.
- Again I took the Second Guest and followed the same.
- Next, I took the third guest to the Railway station and found that the Train had gone.

So, Will I continue the above procedure or stop it?

Obviously, I will stop it and wait for tomorrow.

Code 7: Loop from 0 to 10 (using break)

```
var guest=0;
while(guest<=10)
{
console.log("Guest",guest);
if(guest == 3)
{
break;
}
guest++;
}
```

Student Task

Code 8: Predict the output

```
var count=0;
while(true)
{
console.log("Hello");
count++;
++count;
if(count>5)
{
break;
}
count--;
}
```

Continue

Guest Analogy

- There are 10 guests coming to my home, After 2-3 days they decided to leave their home.
- They all have the train on the same day and at the same time.
- I need to drop them at the railway station but I have one bike which can only take one person at a time.
- In this case, I need to drop each guest one by one.
- Taking the First guest to the railway station, dropping them off, and arriving back and

following the same procedure again and again till the end.

- Suppose I took the First Guest and dropped him to the Railway station and come
back.
- Again I took the Second Guest and followed the same.
- Suppose the third guest is Sick, In that case, I will skip him.
- and I will continue with the fourth guest and follow the same procedure.

Code 11: Loop from 0 to 10 (using continue)

```
var guest=0;
while(guest<=10)
{
if(guest == 3)
{
guest++;
continue;
}
console.log("Guest",guest);
guest++;
}
```

Code 12: Find a Sum of 1 to 10

```
**Problem** :
// Sum of 1 to 10
// 1 + 2 + 3....... + 10
var i = 1;
var sum = 0;
while(i<=10)
{
sum = sum+i;
i++;
}
console.log(sum);
```

for loop

Student task2: Link

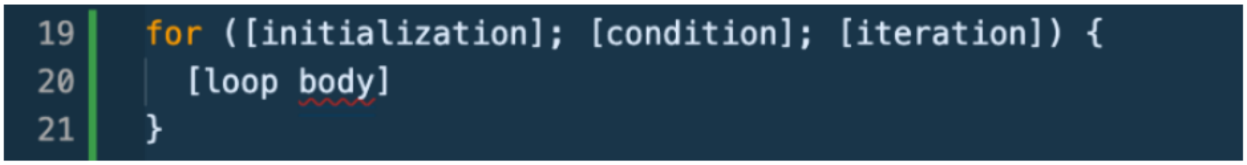
```
// write a for loop
// Initialize it with 1
// calculate the product of number till 5.
// print the product of number between 1 to 5.

var product = 1;
for(){

}
console.log()
```

solution: link

- Let’s say you want to display Masala Dosa one time then you will print the console.log("Masala Dosa") one time.
- Similarly, If it is two then you will write console.log twice
- Let’s say you want to display it 100 times. Without some sort of loop in your code,
  we would probably have to write the same line of code 100 times.

A for-loop can help us to do so by running the same code repeatedly under certain
conditions.

Syntax



```
for ([initialization]; [condition]; [iteration]) {
  [loop body]
}
```

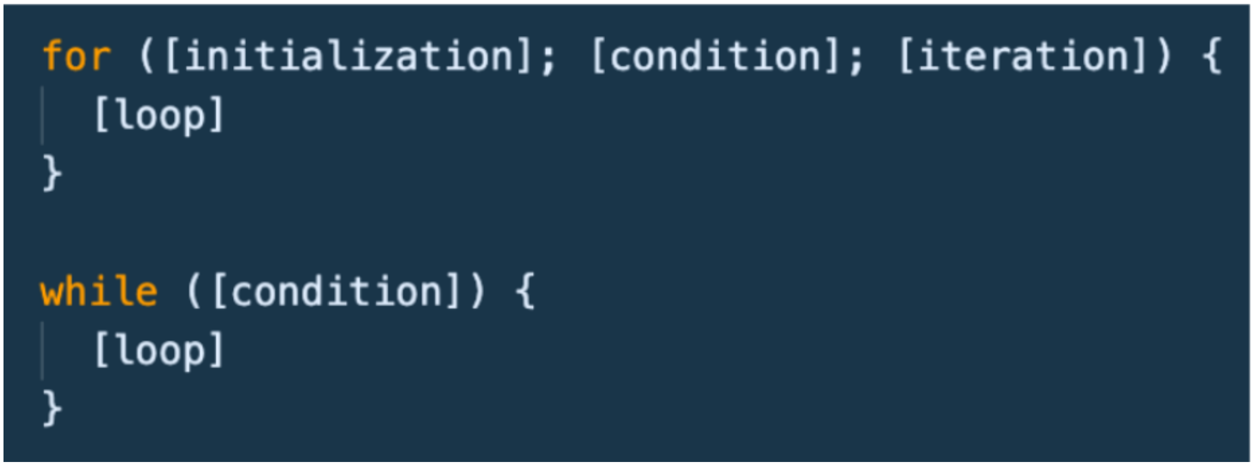
1. **Initialization:** Decides the starting point of a loop

2. **Condition:** Condition is checked before the execution of every iteration. If it evaluates to true, the loop’s statement is executed. If it evaluates to false, the loop stops.
3. **Iteration:** Iteration is used to affect your counter. It can be increment/decrement.
4. **Loop Body:** The loop body repeats the code as long as the condition part is TRUE.

Three Ways of Writing For Loop

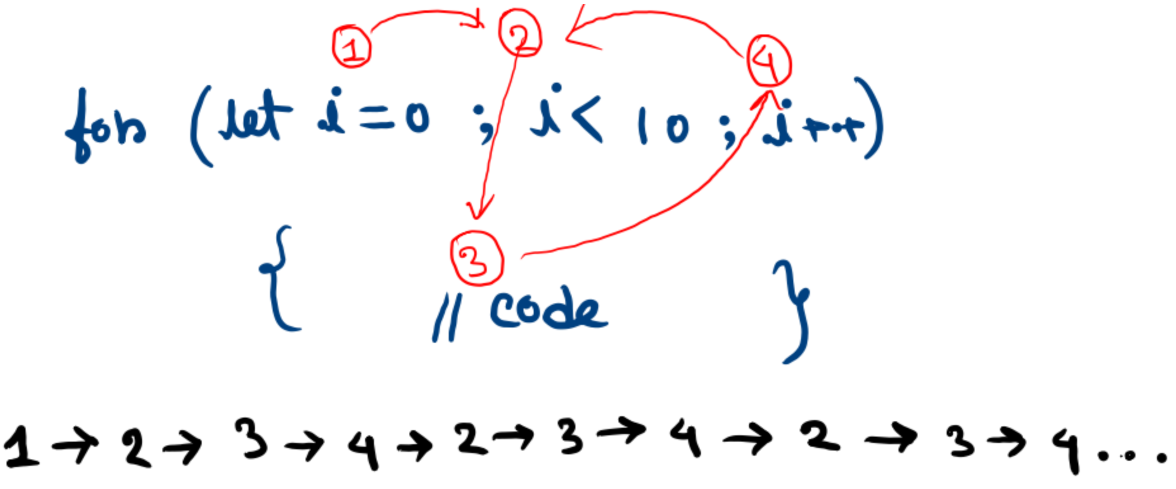
1. `for (initialize; condition; increment);`
2. `for (initialize; condition; increment) single statement;`
3. `for (initialize; condition; increment) { multiple; statements;}`

Comparing For Loop Vs While Loop



```
for ([initialization]; [condition]; [iteration]) {
  [loop]
}

while ([condition]) {
  [loop]
}
```

The sequence of Execution of For Loop

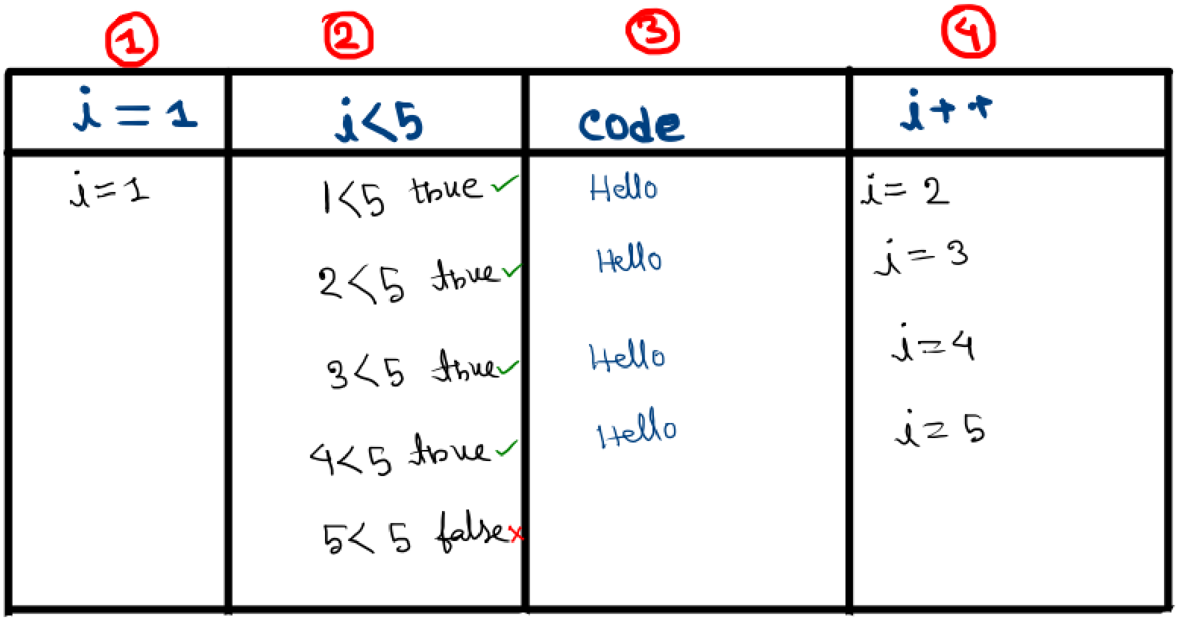


1. Initialization → Condition → Loop Body → Iteration → Condition → Loop Body -
   >
   Iteration and so one
2. Initialization is denoted as 1, Condition denoted as 2, Loop Body is denoted as
   3,
   Iteration denoted as 4.
3. Sequence of Execution will be : 1 → 2 → 3 → 4 → 2→ 3 → 4 → 2 → 3 → 4 and
   so
   on

Examples of For Loop with Dry Run

Example 1: Print Hello 5 times.

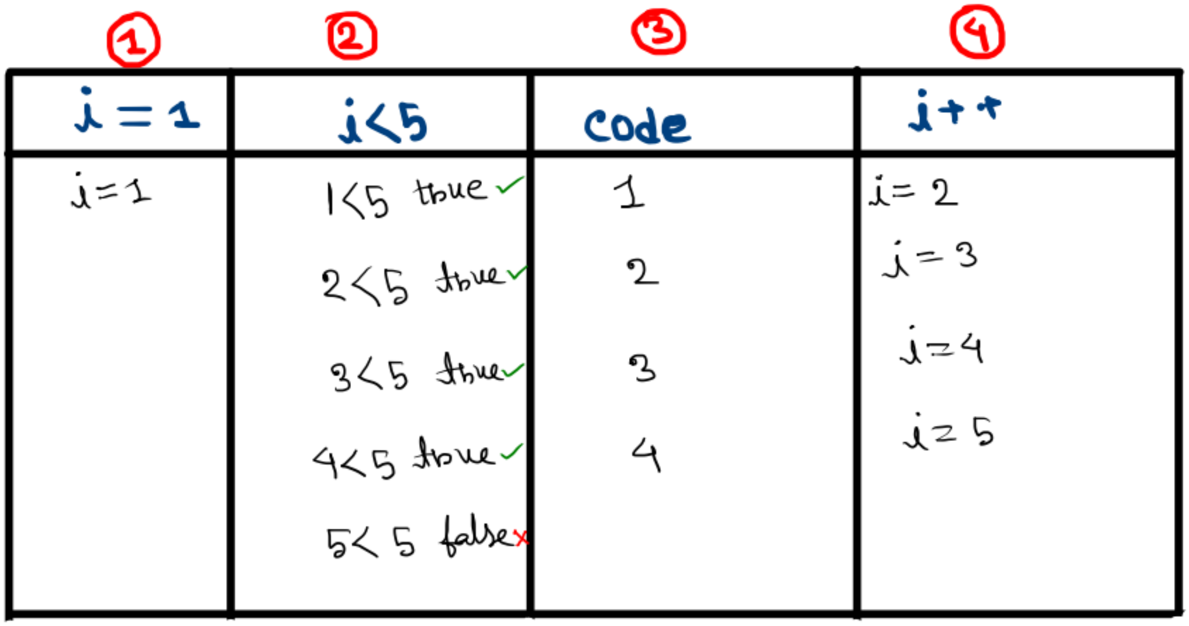
```
for(var i=0; i<5; i++)
{
console.log("Hello");
}
```



①	②	③	④
i = 1	i<5	Code	i++
i=1	1<5 true✓	Hello	i= 2
	2<5 true✓	Hello	i=3
	3<5 true✓	Hello	i=4
	4<5 true✓	Hello	i=5
	5< 5 false✗		

Example 2: Print Values from 1 to 5

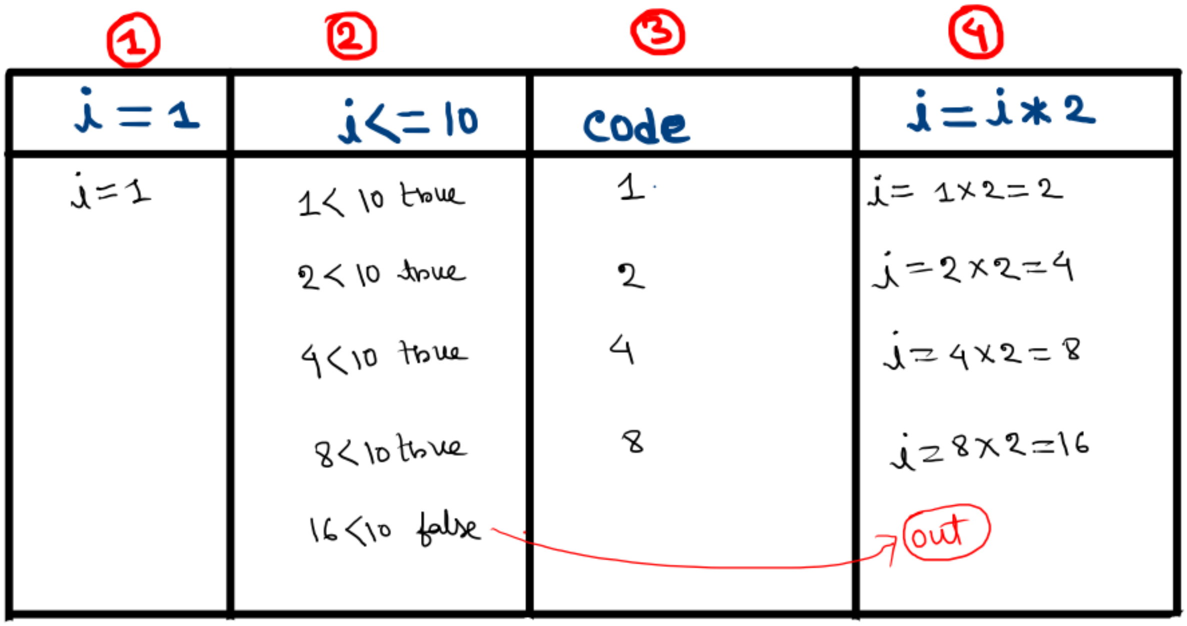
```
// Ist way : Output on new line
for(var i = 1; i<=5; i++){
console.log(i);
}
// IInd way: Output on a single line
var bag="";
for(var i = 1; i<=5; i++){
bag = bag + i + " ";
}
console.log(bag);
```



①	②	③	④
i = 1	i<5	code	i++
i=1	1<5 true ✓	1	i= 2
	2<5 true ✓	2	i=3
	3<5 true ✓	3	i=4
	4<5 true ✓	4	i= 5
	5< 5 false ✗		

Example 3: Print multiple of 2 values from 1 to 10

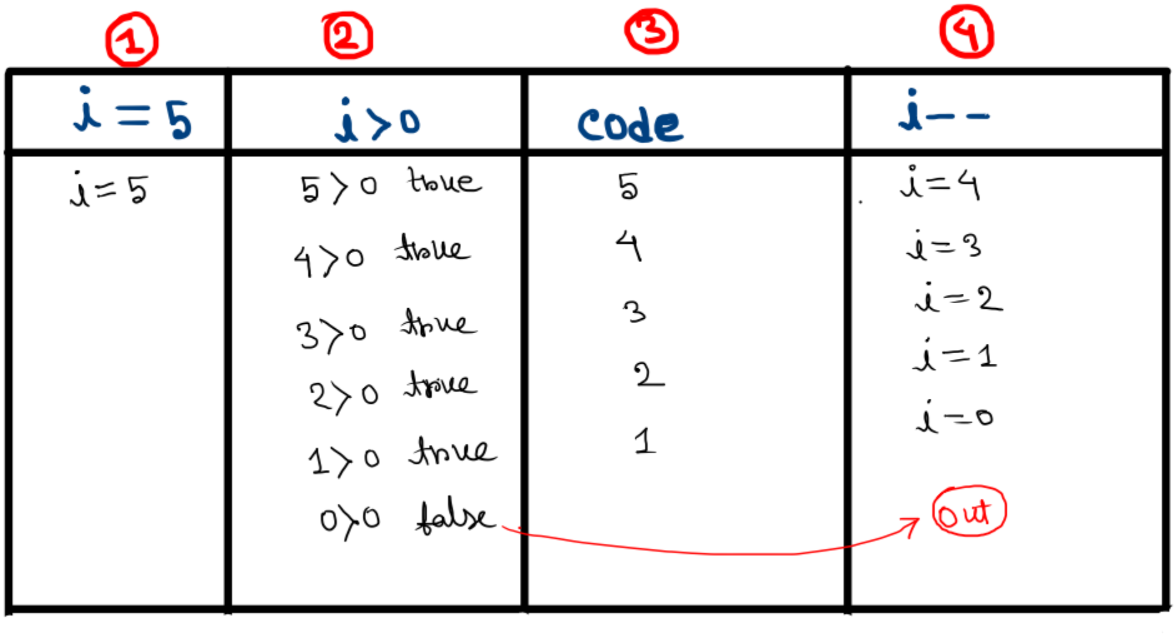
```
// Ist way : Output on new line
for(var i = 1; i<=10; i=i*2){
console.log(i);
}
// IInd way : Output on single line
var bag="";
for(var i = 1; i<=10; i=i*2){
bag = bag + i + " ";
}
console.log(bag);
```



1	2	3	4
i = 1	i<= 10	code	i=i*2
i=1	1< 10 true	1.	i= 1×2=2
	2< 10 true	2	i=2×2=4
	4<10 true	4	i= 4×2=8
	8<10 true	8	i=8×2=16
	16<10 false		out

Example 4: Reverse Loop from 5 to 1

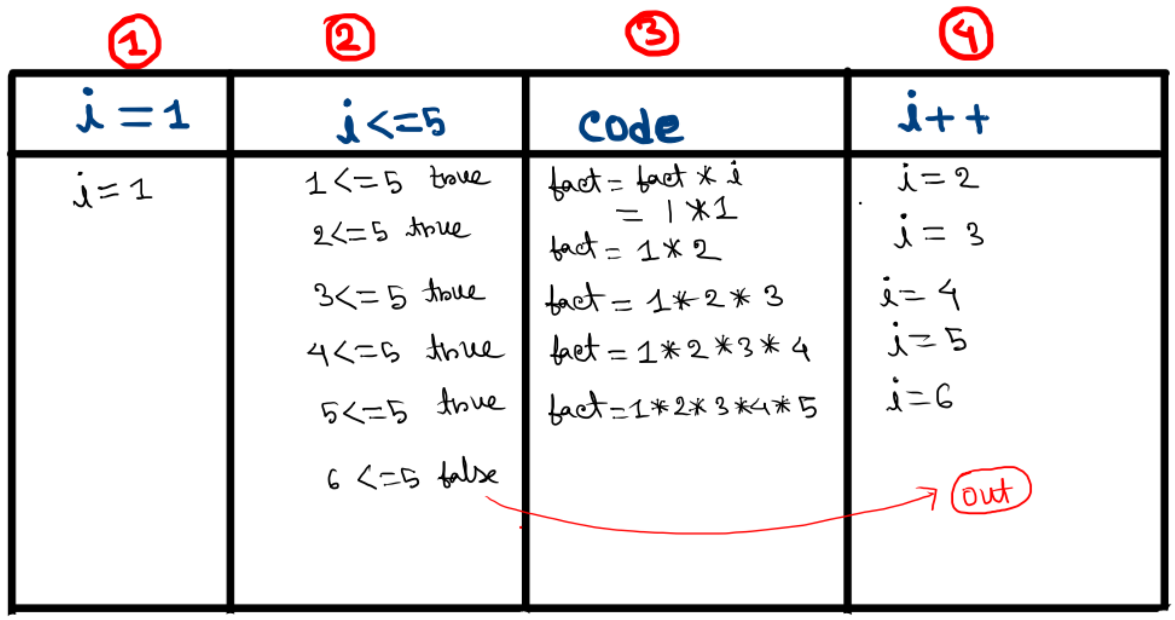
```
var bag = "";
for(var i = 5; i>0; i--){
bag = bag + i + " ";
}
console.log(bag);
```



1	2	3	4
i=5	i>0	Code	i--
i=5	5>0 true	5	i=4
	4>0 true	4	i=3
	3>0 true	3	i=2
	2>0 true	2	i=1
	1>0 true	1	i=0
	0>0 false		out

Example 5: Factorial

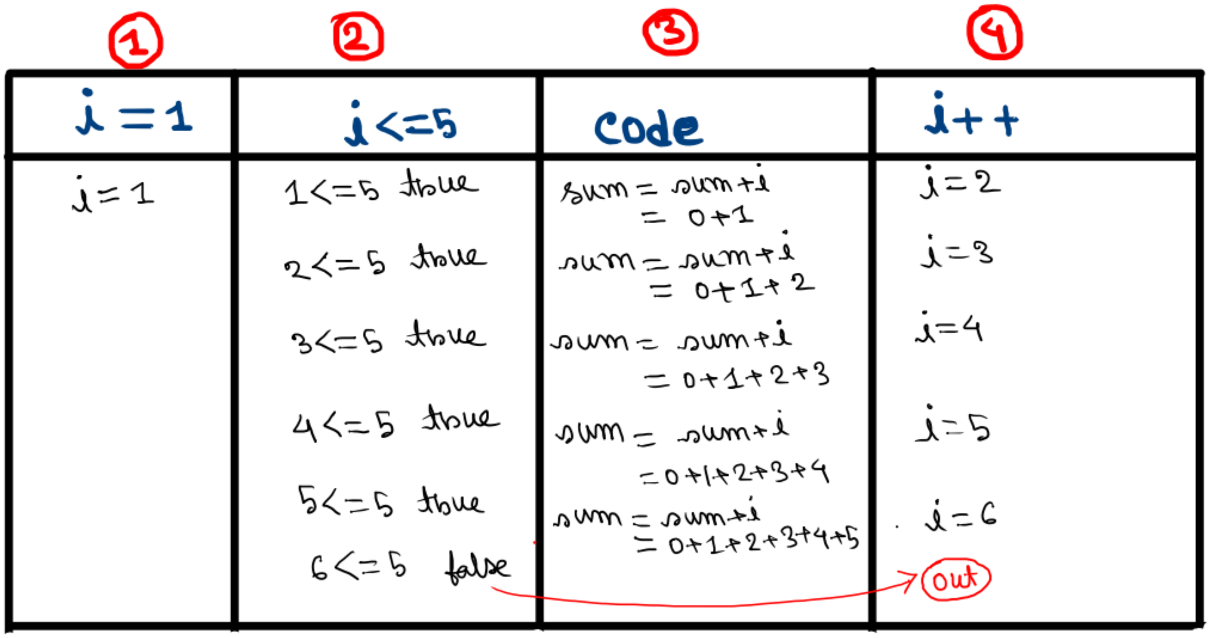
```
var fact = 1;
for(var i=1; i<=5;i++)
{
fact = fact * i;
console.log(fact);
}
```



①	②	③	④
i = 1	i<=5	Code	i++
i=1	1<=5 true	fact = fact * i = 1*1	i=2
	2<=5 true	fact = 1*2	i = 3
	3<=5 true	fact = 1*2*3	i= 4
	4<=5 true	fact = 1*2*3*4	i=5
	5<=5 true	fact=1*2*3*4*5	i=6
	6 <=5 false		out

Example 6: Find Sum 1 to N

```
var N = 5;
var sum = 0;
for(var i = 1; i<=N; i++){
sum = sum + i;
}
console.log(sum);
```



1	2	3	4
i=1	i<=5	Code	i++
i=1	1<=5 true	sum = sum+i = 0+1	i=2
	2<=5 true	sum = sum+i = 0+1+2	i=3
	3<=5 true	sum = sum+i = 0+1+2+3	i=4
	4<=5 true	sum = sum+i =0+1+2+3+4	i=5
	5<=5 true	sum = sum+i = 0+1+2+3+4+5	i=6
	6<=5 false		out

Break

Guest Analogy

- There were 10 guests who came to my home, After 2-3 days they decided to leave.
- They all have the train on the same day and at the same time.
- I need to drop them at the railway station but I have one bike which can only take
  one person at a time.
- In this case, I need to drop each guest one by one.
- Taking the First guest to the railway station, dropping them off, and arriving back and
  following the same procedure again and again till the end.
- Suppose I took the First Guest and dropped him to the Railway station and come
  back.

- Again I took the Second Guest and followed the same.
- Now, I took the third guest to the Railway station and found that the Train had gone.

So, Will I continue the above procedure or stop it?

Obviously, I will stop it and wait for tomorrow

Code 7 : Loop from 1 to 10 (using break). Using console.log before break statement

```
for(var guest=1; guest<=10; guest++)
{
console.log("guest ",guest,"got the train");
if(guest == 3){
break;
}
}
```

Code 8 : Loop from 1 to 10 (using break). Using console.log after break statement

```
for(var guest=1; guest<=10; guest++)
{
if(guest == 3){
break;
}
console.log("guest ",guest,"got the train");
}
```

Continue

Guest Analogy

- There are 10 guests coming to my home, After 2-3 days they decided to leave their
  home.
- They all have the train on the same day and at the same time.
- I need to drop them at the railway station but I have one bike which can only take
  one person at a time.
- In this case, I need to drop each guest one by one.
- Taking the First guest to the railway station, dropping them off, and arriving back and
  following the same procedure again and again till the end.
- Suppose I took the First Guest and dropped him to the Railway station and come
  back.
- Again I took the Second Guest and followed the same.
- Suppose the third guest is Sick, In that case, I will skip him.
- and I will continue with the fourth guest and follow the same procedure.

Code 9: Loop from 1 to 10 (using Continue). Using console.log before continue statement

```
for(var guest=1; guest<=10; guest++)
{
console.log("guest ",guest,"got the train");
if(guest == 3){
continue;
}
}
```

Code 10: Loop from 1 to 10 (using Continue). Using console.log after continue statement

```
for(var guest=1; guest<=10; guest++)
{
if(guest == 3){
continue;
}
console.log("guest ",guest,"got the train");
}
```

Code 11: Predict the output.

```
var count = 1;
for(var i = 1; i<10; i++)
{
count++;
if(i==5){
continue;
}
}
console.log(count);
```

Code 12: Predict the output.

```
var count = 1;
for(var i = 1; i<10; i++)
{
if(i==5){
continue;
}
count++;
}
console.log(count);
```