

While Loop

Increment & Decrement

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

Aa Title	Increment	Decrement
Untitled	<pre>var a = 5; a = a + 1; console.log(a);</pre>	<pre>var a = 5; a = a - 1; console.log(a);</pre>

- 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.

Aa Title	Increment(<code>++</code>)	Decrement(<code>--</code>)
Untitled	<pre>var a = 1 console.log(a++); console.log(a);</pre>	<pre>var a = 1; console.log(a--); console.log(a);</pre>

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 knowns as Prefix Increment & Decrement.

Aa Title	≡ Increment(++)	≡ Decrement(--)
<u>Untitled</u>	<code>var a = 1; console.log(++a); console.log(a);</code>	<code>var a = 1; console.log(++a); console.log(a);</code>

Examples of Prefix and Postfix

Aa Title	≡ Prefix	≡ Postfix
<u>Untitled</u>	<code>var a = 10; var c = a++; console.log(a); // 11 console.log(c); // 10</code>	<code>var a = 10; var c = ++a; console.log(a); console.log(c);</code>

Student Task

Code 1 : Predict the output

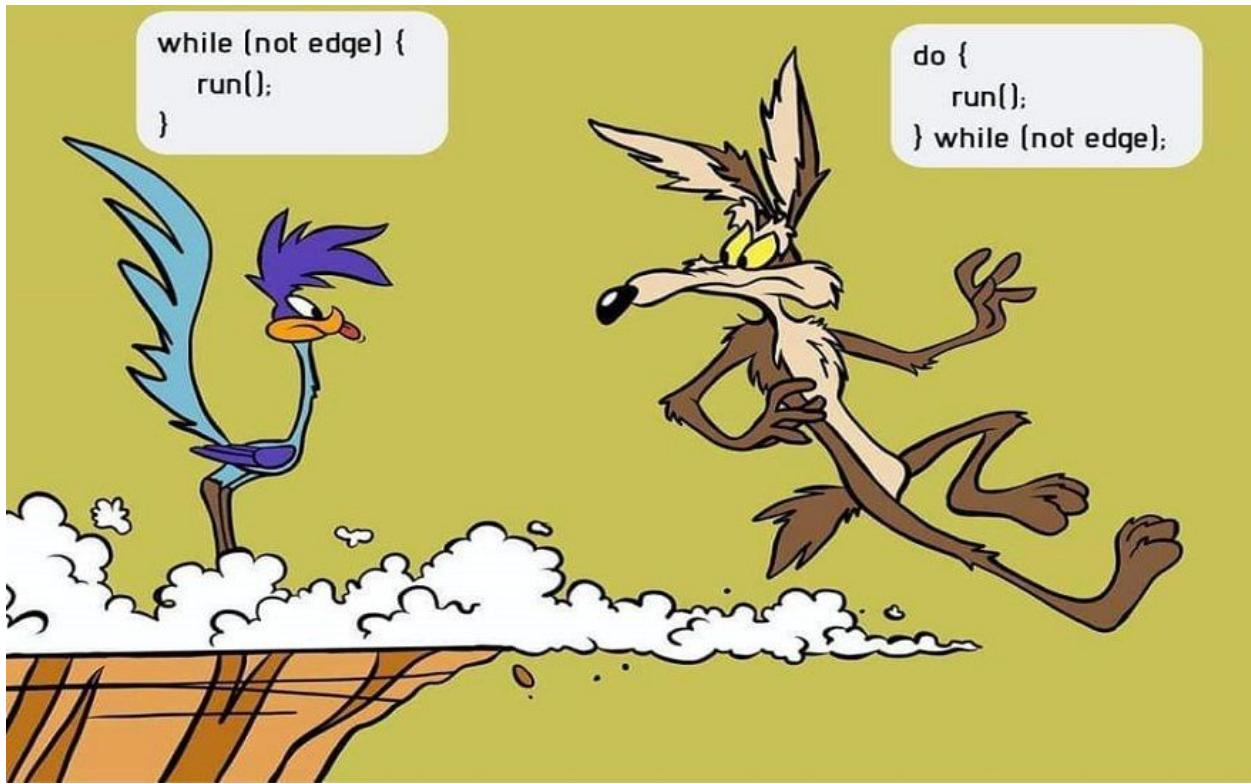
```
var a = 10;  
++a;  
var b = 10;  
b++;  
console.log(a)  
console.log(b);
```

Student Task

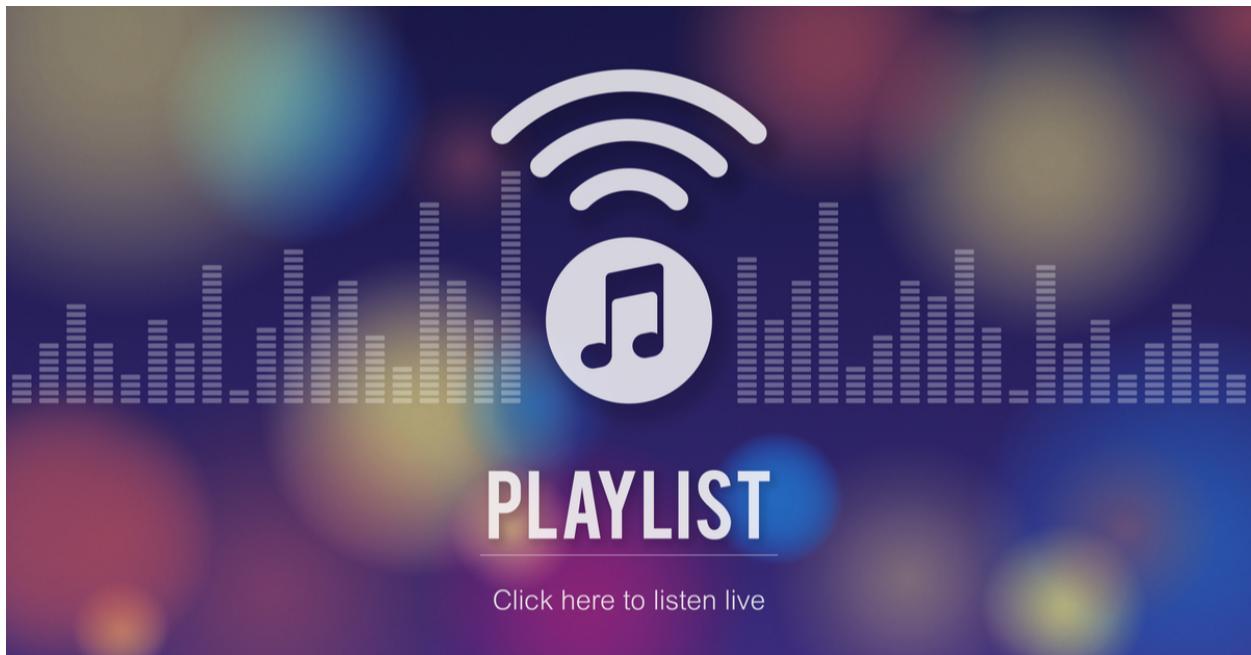
Code 2 : Predict the output

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

Loop



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

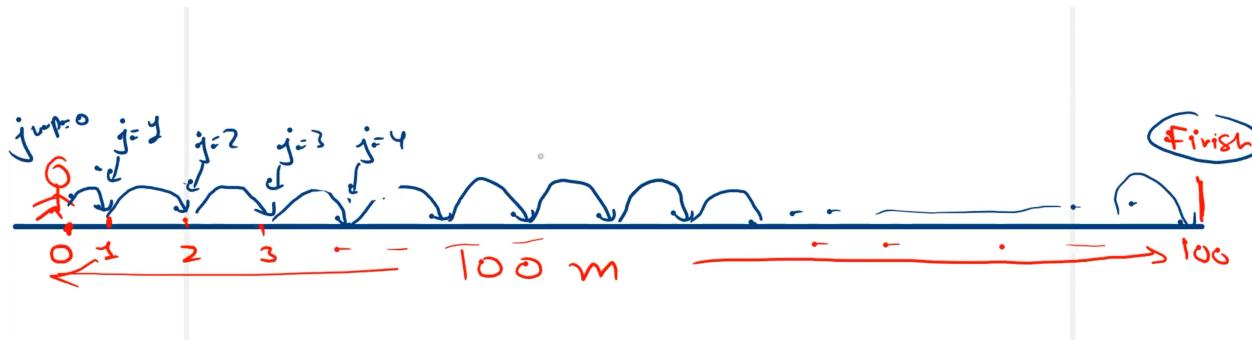


- There are 10 guests coming to my home, After 2-3 days they decided to leave their home.
- They all have the train on the same day and at the same time.
- I need to drop them at the railway station but I have one bike which can only take one person at a time.
- In this case, I need to drop each guest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

While Loop

- The while loop begins with a condition and it is written similar 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 train himself for a long Marathon of 100 meter. Hari standing at 0th position and he needs to cover 100 meter distance.

Hari will make 1 meter jump at a time.

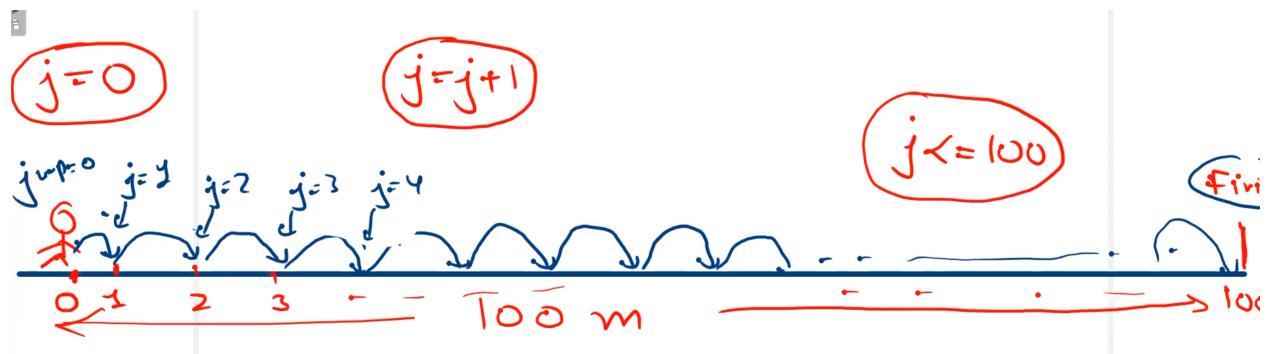
Lets take Hari position as position = 0

After 1st jump, Hari will be at position = 1 meter

After 2nd jump, Hari will be at position = 2 meter

.....
After 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 \leq 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 \leq 100 meter**

Syntax of While Loop

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

```
Starting Point  
While ( Till When )  
{  
    How long Jump at a time?
```

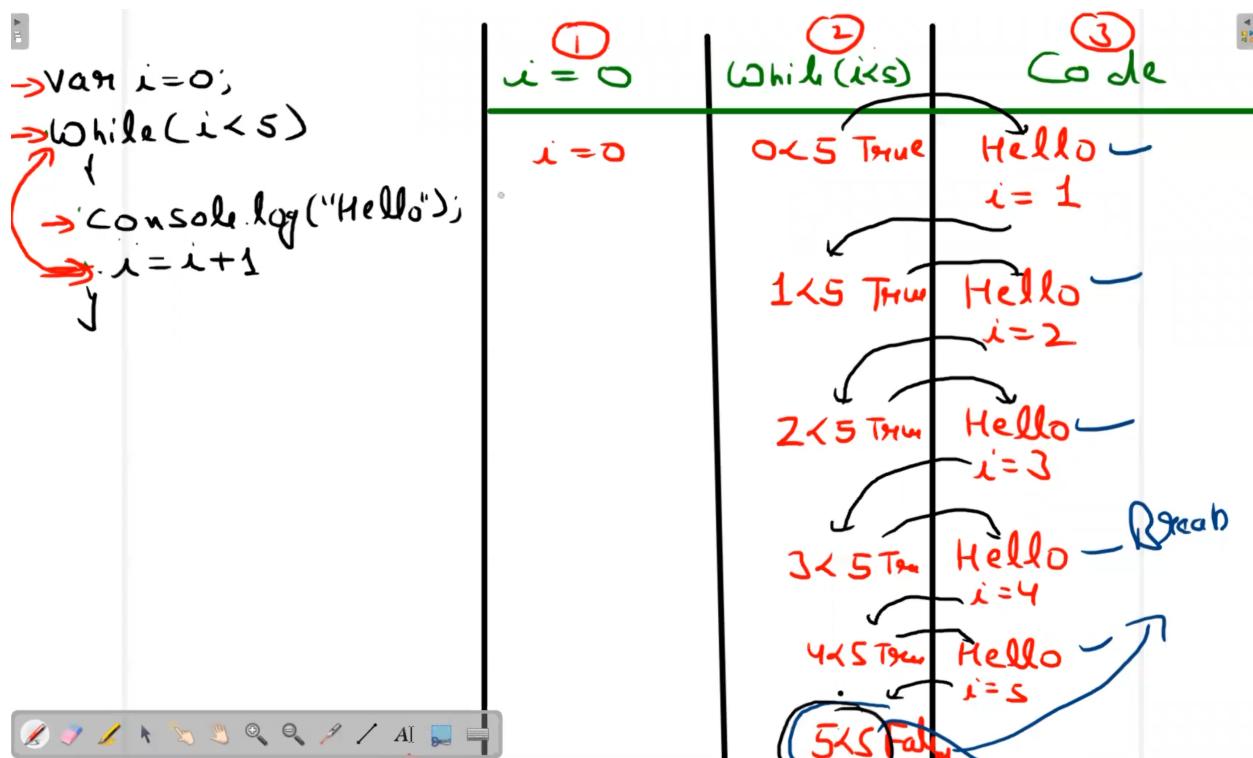
```
}
```

```
Initialization
While ( Condition )
{
    Increment/Decrement
}
```

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

1	let i = 0
2	
3	while (i < 10){
4	console.log(i);
5	i ++
6	}

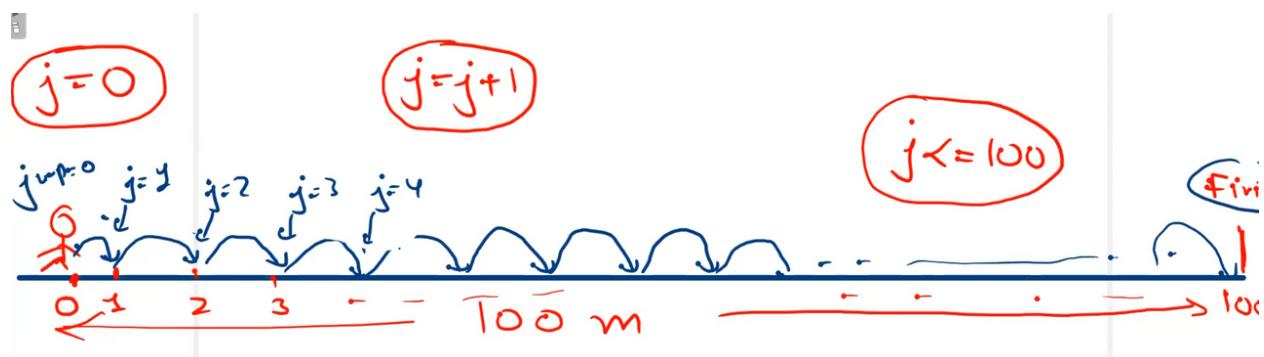
Understanding DRY RUN



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

While Loop Examples

Code 3 : 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);
}

```

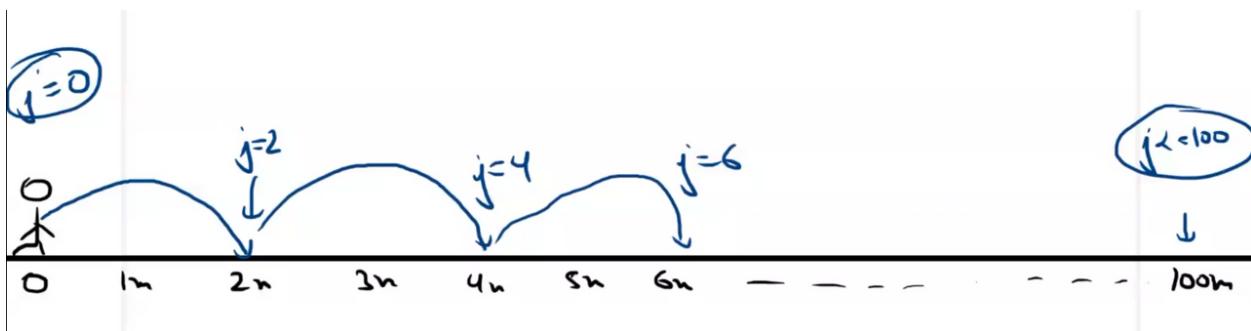
Code 4 : Infinite Loop

```

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

```

Code 5 : 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 6 : Loop from 1 to 100 [15 meter jump at a time]

// PIC of image track

```

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

```

Note : != and < behave differently

Student Task

Code 7 : Loop from 35 to 100 [3 units jump at a time]

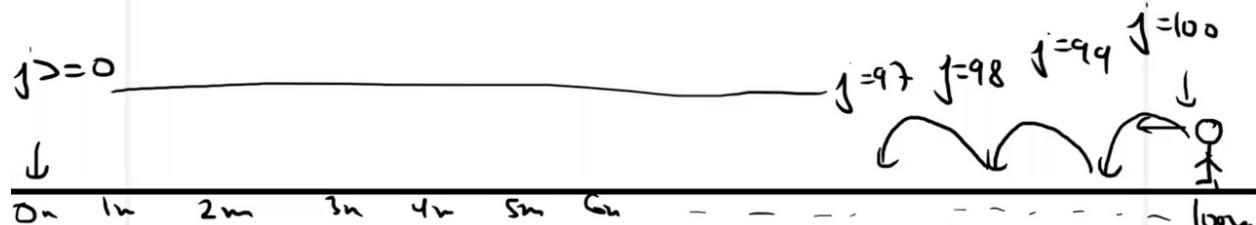
```

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

```

Code 8 : Reverse Loop from 100 to 0 [1 units jump at a time]

Reverse loop



Start : $j=100$
 end : $j>=0$
 jump : $\text{jump} = \text{jump} - 1$

```
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 and arrive back and follow 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 follow the same.
- Now, Next I took the third guest to Railway station and found that Train has gone.

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

Obviously, I will stop it and wait for tomorrow.

Code 9 : 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 10 : 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 and arrive back and follow 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 follow 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)
{
  console.log("Guest",guest);

  if(guest == 3)
  {
    continue;
  }

  guest++;
}
```

Code 12 : Find 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);
```