

Javascript

Types of For loop

`for(i=0; i<n; i++)`
`for....of`
`for await....of`
`for....in`



@startwithmani



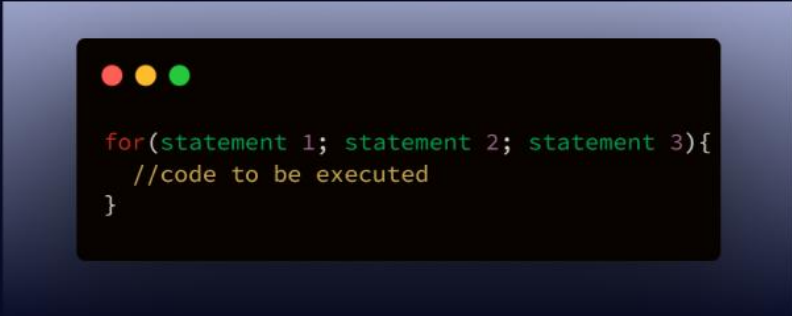
@M.serisha Kothapalli

←swipe

Traditional for loop

The simplest traditional type of *for loop* which increments a variable as its iteration method. The variable acts as a counter for every “n”th element within an object.

Syntax




```
for(statement 1; statement 2; statement 3){  
    //code to be executed  
}
```

Statement 1 is executed only once before the execution of the code.

Statement 2 is the condition for executing the code block.

Statement 3 is executed every time after the code block has been executed.

 @startwithmani

 @M.serisha Kothapalli





```
for(let a = 0; a < 10; a++){  
  console.log(a)  
}  
  
// 0 1 2 3 4 5 6 7 8 9
```

- The above program first **initializes** a with 0. Then it checks for the **condition** if a is less than 10. If the condition returns true, then it executes the block of statements. Then it **increments** a by 1 (third statement).
- Now a = 1 ..checks for the condition if a < 10 executes block of code. a increments by 1 again
- Now a = 2 ..checks for the condition if a < 10 executes block of code. a increments by 1.....This continues a=3 , a=4.....so on
- The **loop will end**, if the **condition** a<10 returns **false**
- So when a=10 it checks for condition if a<10 it returns false. So the loop ends



@startwithmani



@M.serisha Kothapalli

←swipe

for of loop

The *for of loop* iterates over values of iterable like String, Map, Set, Array etc...


Syntax

```
for(variable of iterable){  
  //code to be executed  
}
```

Variable The value of iterable will be assigned to this variable

Iterable Any object which has looping properties



 @startwithmani

 @M.serisha Kothapalli

←swipe

```
for(let a of 'mani'){  
  console.log(a)  // m a n i  
}  
  
for(let a of [1,2,3,4]){  
  console.log(a)  // 1 2 3 4  
}
```

Variable : From above example a iterates over the iterable and it will be assigned with new value.


 @startwithmani
 @M.serisha Kothapalli

←swipe

for in loop

The *for in loop* iterates through the properties of an Object

Syntax



```
for(key in Object){  
  //code to be executed  
}
```

key For each iteration each key of the object will be assigned.




```
let obj = {'x':1,'y':2}
for(key in obj){
  console.log(key,obj[key])
  //x,1    y,2
}
```

From above example key will be assigned **x** for **first** iteration and **y** for **second** iteration

To get the key of the object for each iteration: **key**

To get the value of the object for each iteration: **obj[key]**

 @startwithmani


 @M.serisha Kothapalli



for await ...of loop

The *for await ..of* statement creates a loop iterating over async iterable objects as well as on sync iterables, String, Array, Array-like objects, Map, Set etc...

Syntax




```
for await (variable of iterable) {  
  //code to be executed  
}
```

Variable For each iteration different property will be assigned to this variable

Iterable Any object which has looping properties

 @startwithmani

 @M.serisha Kothapalli

←swipe


```
async function* asyncGenerator() {  
  yield 'a'  
  yield 'b'  
}  
  
(async () => {  
  for await (const num of asyncGenerator()) {  
    console.log(num);  
    // a b  
  }  
})();
```

Note: The **generator** function is denoted by *****. You can either use `function* asyncGenerator() {...}` or `function *asyncGenerator(){...}` to create them.

 @startwithmani

 @M.serisha Kothapalli

←swipe

Did you find it helpful??



Like this post!



Share with your friends



Save it for later



Follow for more!



@startwithmani



@M.serisha Kothapalli