LOOPS

Loops:

In JavaScript, the for loop is used for iterating over a block of code a certain number of times, or to iterate over the elements of an array.

For loop:

The **JavaScript for loop** *iterates the elements for the fixed number of times*. It should be used if number of iteration is known. The syntax of for loop is given below.

Syntax:

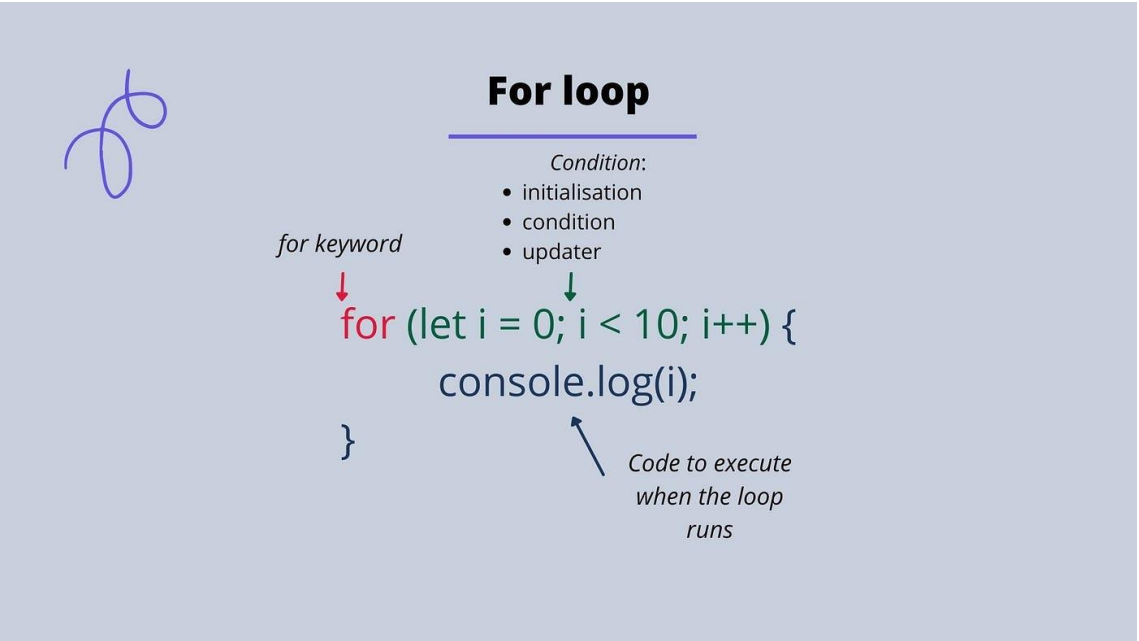
```
for (initialization; condition; increment)
{
   code to be executed
}
```

Example:

```
<script>
for (i=1; i<=5; i++)
{
document.write(i + "<br/>")
}
</script>
```

Output:

1
2
3
4
5



Nested For loop:

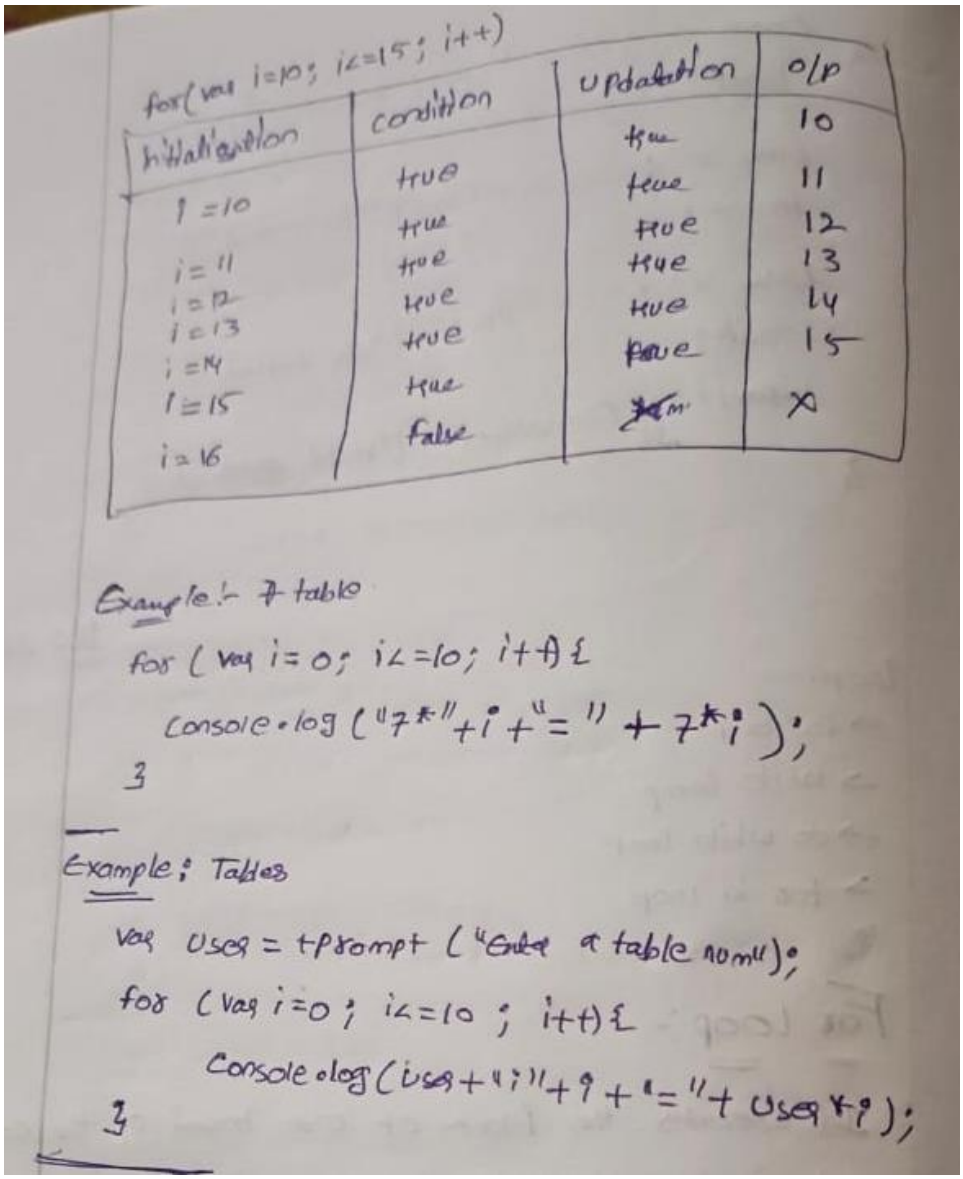
If we have loop inside loop, this is called a nested loop.

Example:

```
<script>
    for(var i=1; i<=5; i++){
        console.log("Table "+ i);
        for(var j=1; j<=10; j++){

            var x=i*j;
            console.log(i+ " * "+j+'='+x);


        }
    }
</script>
```



for(var i=10; i<=15; i++)

Initialization	condition	updation	o/p
i=10	true	true	10
i=11	true	true	11
i=12	true	true	12
i=13	true	true	13
i=14	true	true	14
i=15	true	true	15
i=16	false	~~true~~	X

Example:- 7 table

```
for (var i=0; i<=10; i++){
  console.log("7*"+i+"="+7*i);
}
```

Example: Tables

```
var useq = +prompt("Enter a table num");
for (var i=0; i<=10; i++){
    console.log(useq+"*"+i+"="+useq*i);
}
```

Example:- break the loop

```
for(i=1; i<=20; i++){
    if (i%2==0){
        console.log(i+" is even");
        break;
    }
    else{
        console.log(i);
    }
}
```

Example:- continue for skip the current iteration.

```
for (i=1; i<=2; i++){
    if (i%2==0){
        continue;
        console.log(i);
    }
    else{
        console.log(i)
    }
}
```

Example:- sum of even numbers

```
var count=0;
for (i=1; i<=10; i++){
    if (i%2==0){
        count = count+1;
    }
}
console.log(count);    //30
```

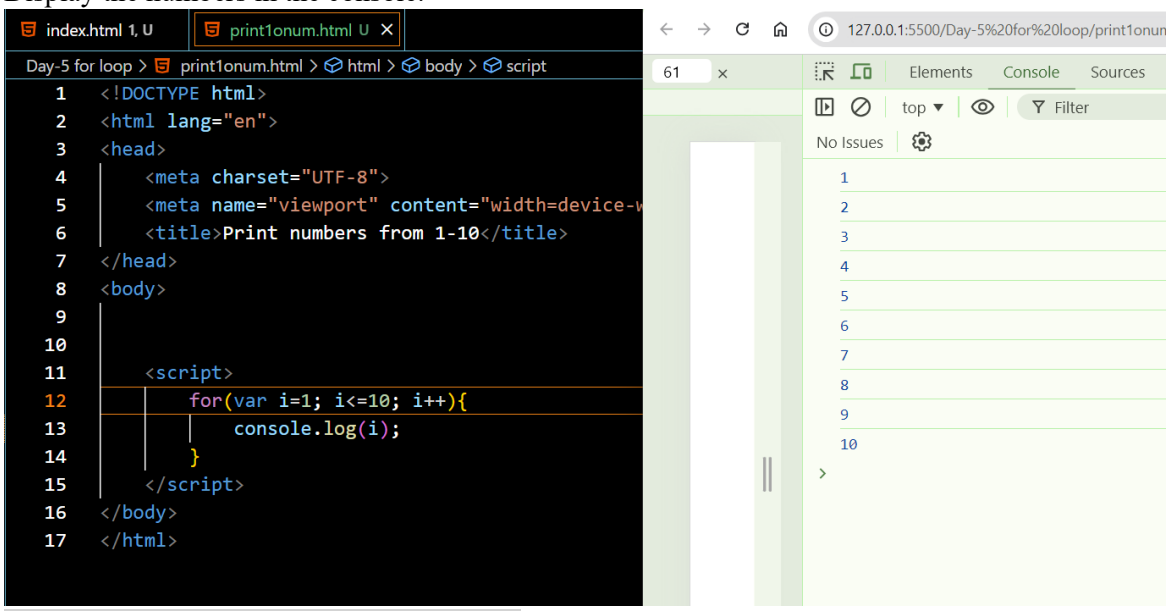
Tasks:

Task 1: Print Numbers from 1 to 10

Instructions:

Use a for loop to print numbers from 1 to 10.

Display the numbers in the console.



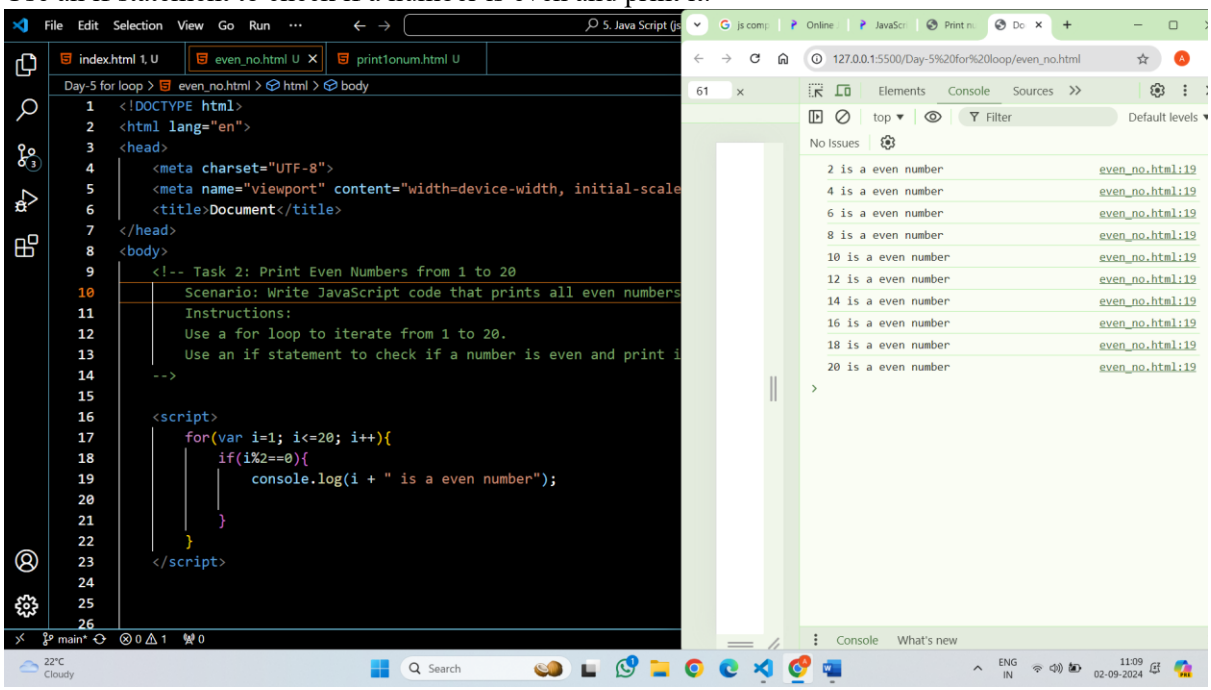
Task 2: Print Even Numbers from 1 to 20

Scenario: Write JavaScript code that prints all even numbers from 1 to 20.

Instructions:

Use a for loop to iterate from 1 to 20.

Use an if statement to check if a number is even and print it.



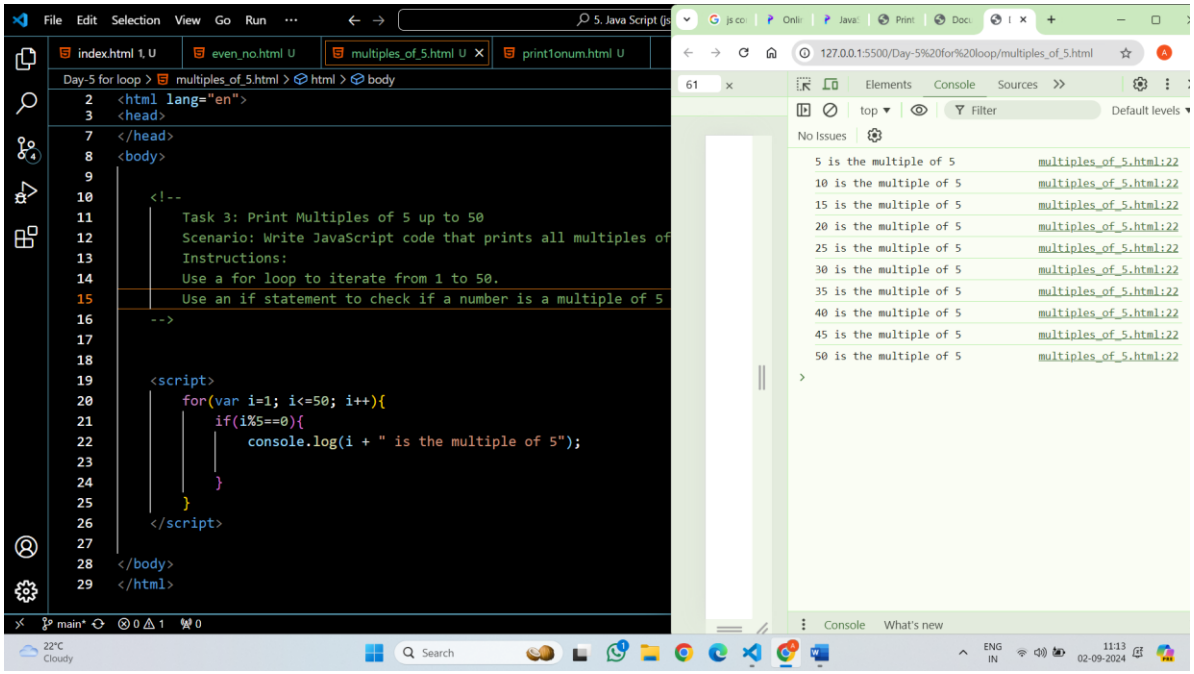
Task 3: Print Multiples of 5 up to 50

Scenario: Write JavaScript code that prints all multiples of 5 up to 50.

Instructions:

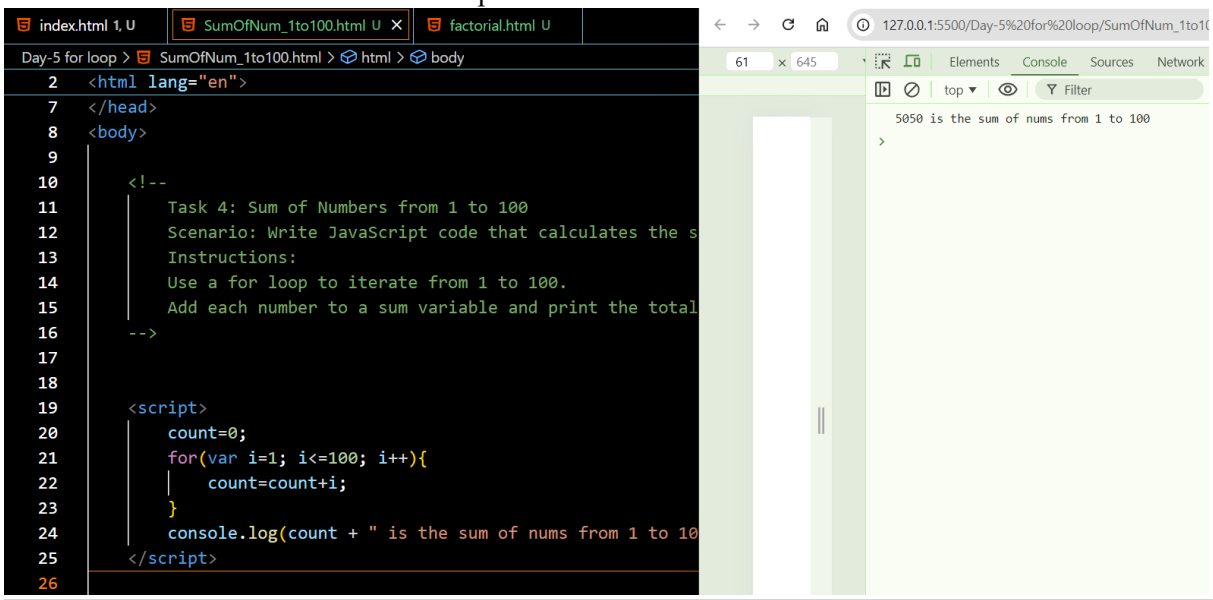
Use a for loop to iterate from 1 to 50.

Use an if statement to check if a number is a multiple of 5 and print it.



Task 4: Sum of Numbers from 1 to 100

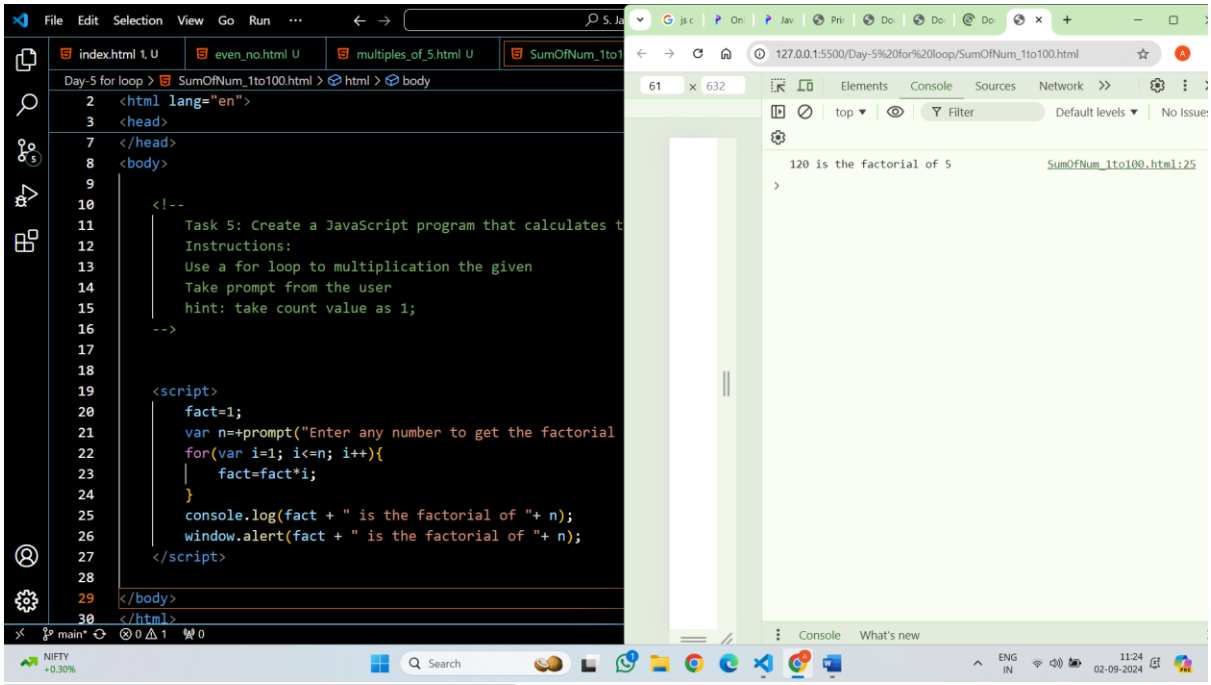
Scenario: Write JavaScript code that calculates the sum of numbers from 1 to 100.
Instructions:
Use a for loop to iterate from 1 to 100.
Add each number to a sum variable and print the total sum.



Task 5: Create a JavaScript program that calculates the factorial of a given number using a for loop.

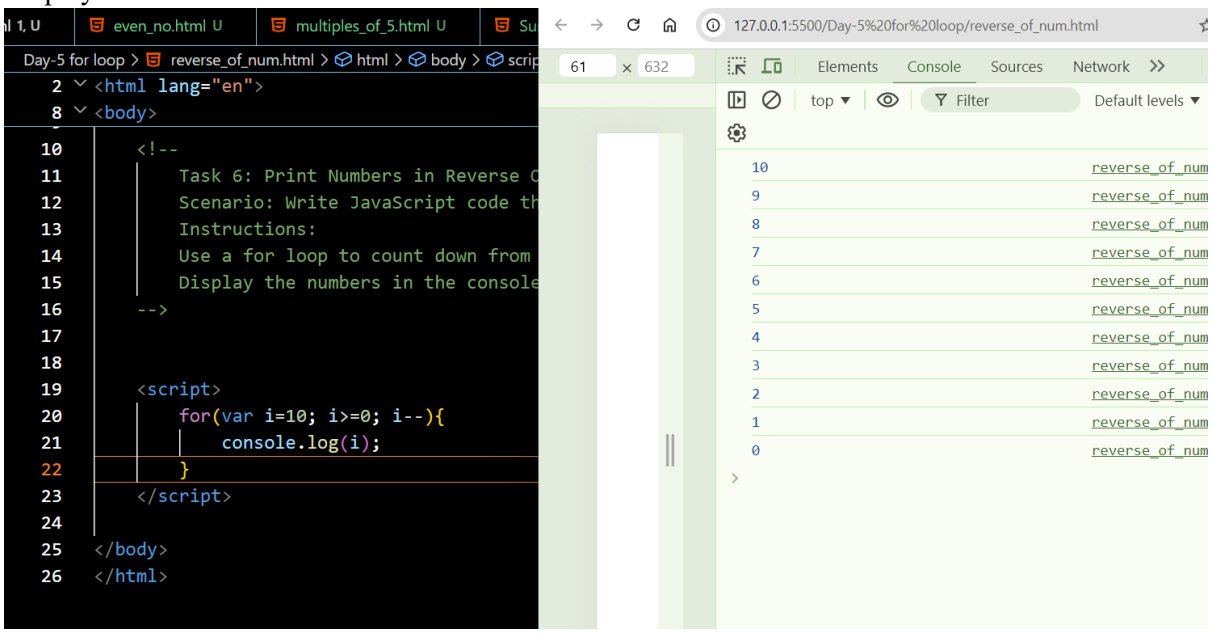
Instructions:

Use a for loop to multiplication the given
Take prompt from the user
hint: take count value as 1;



Task 6: Print Numbers in Reverse Order

Scenario: Write JavaScript code that prints numbers from 10 to 1 in reverse order.
Instructions:
Use a for loop to count down from 10 to 1.
Display the numbers in the console.



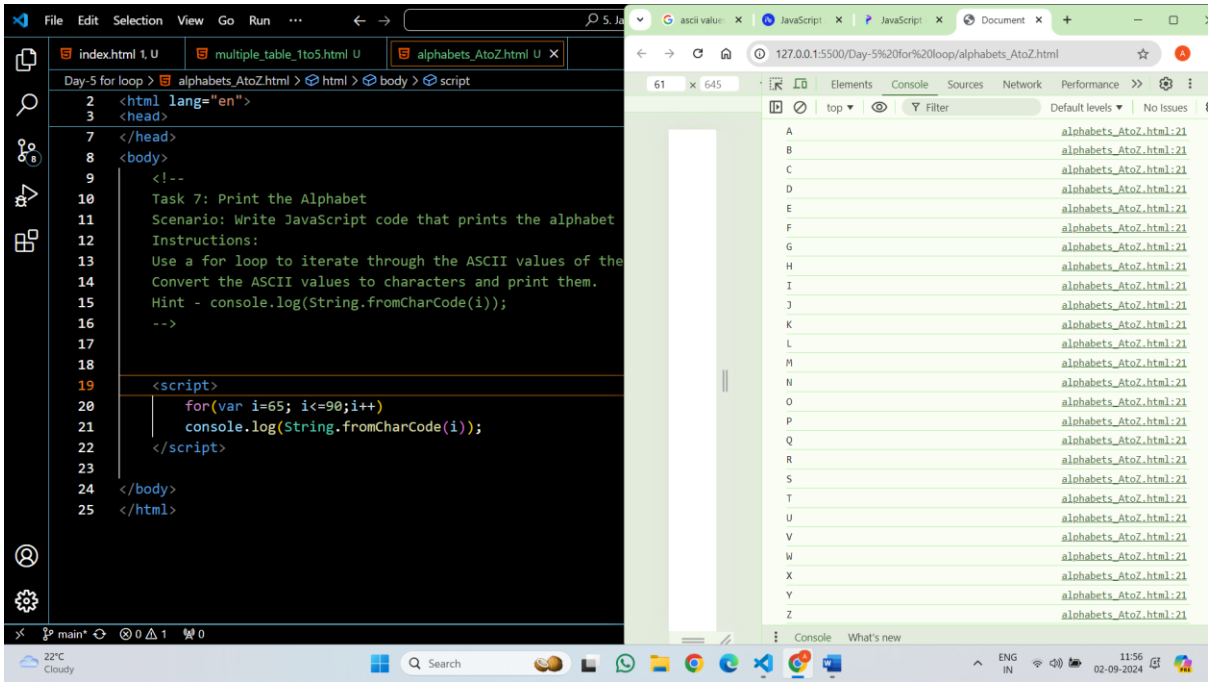
(optional)

Task 7: Print the Alphabet

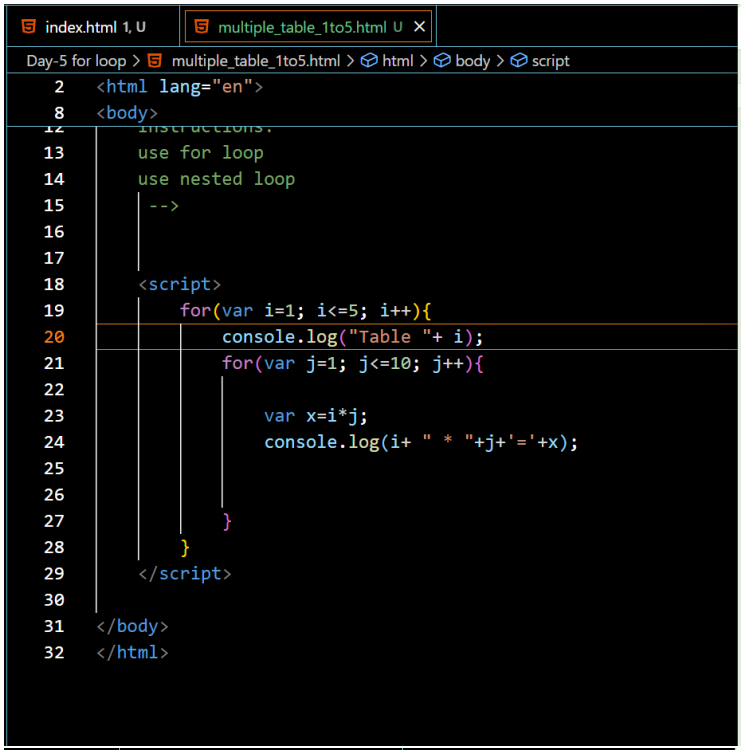
Scenario: Write JavaScript code that prints the alphabet from A to Z.
Instructions:
Use a for loop to iterate through the ASCII values of the letters A to Z.
Convert the ASCII values to characters and print them.
Hint - console.log(String.fromCharCode(i));

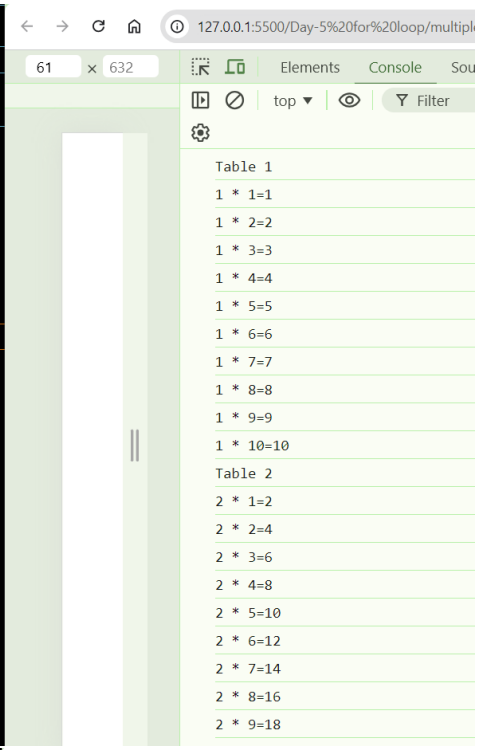
ASCII TABLE

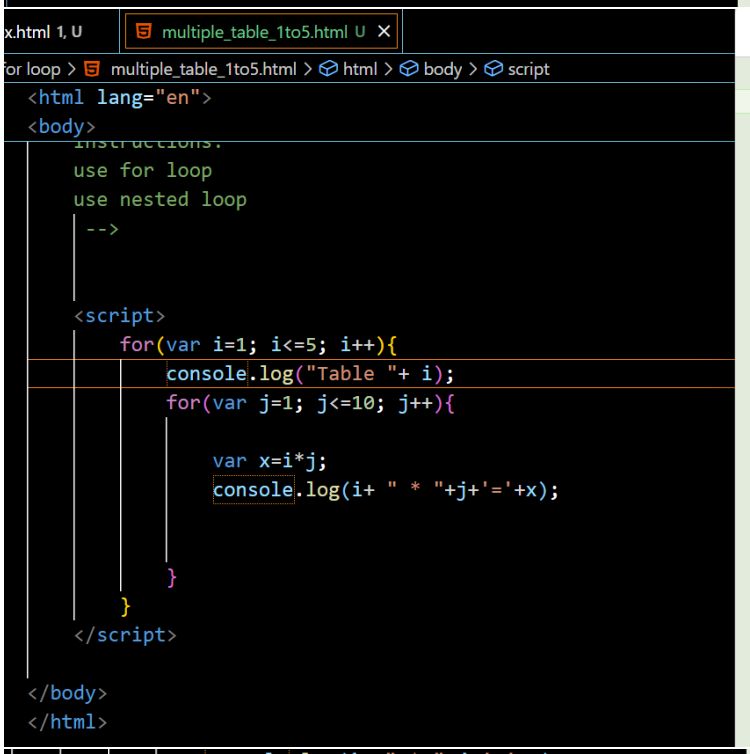
Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

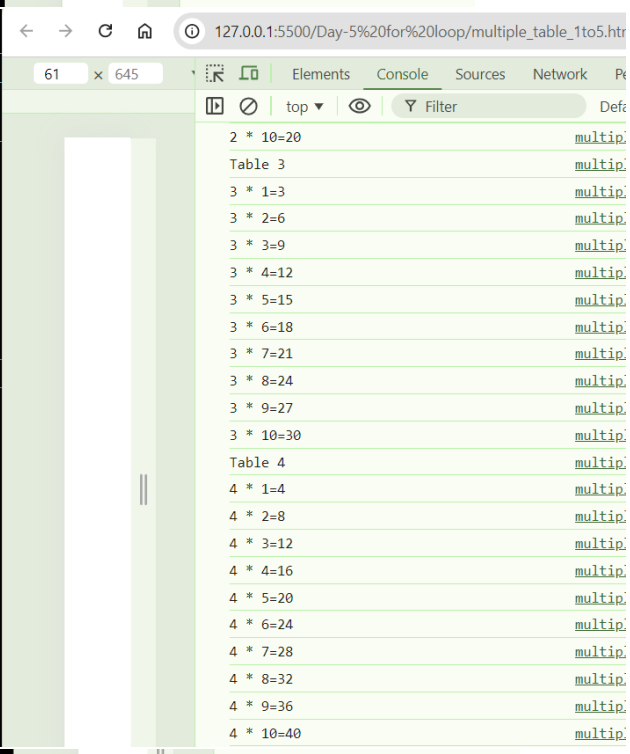


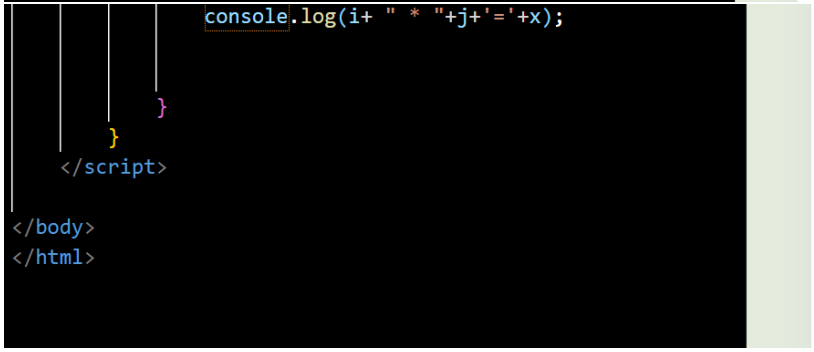
Task 8:Write a JavaScript script that uses nested loops to print a multiplication table for numbers 1 through 5.
Instructions:
use for loop
use nested loop


index.html 1, U
multiple_table_1to5.html U
Day-5 for loop > multiple_table_1to5.html > html > body > script
<html lang="en">
<body>
use for loop
use nested loop
-->
<script>
for(var i=1; i<=5; i++){
console.log("Table "+ i);
for(var j=1; j<=10; j++){
var x=i*j;
console.log(i+ " * "+j+'='+x);
}
}
</script>
</body>
</html>


127.0.0.1:5500/Day-5%20for%20loop/multipl
61 x 632
Elements
Console
top
Filter
Table 1
1 * 1=1
1 * 2=2
1 * 3=3
1 * 4=4
1 * 5=5
1 * 6=6
1 * 7=7
1 * 8=8
1 * 9=9
1 * 10=10
Table 2
2 * 1=2
2 * 2=4
2 * 3=6
2 * 4=8
2 * 5=10
2 * 6=12
2 * 7=14
2 * 8=16
2 * 9=18


multiple_table_1to5.html U
multiple_table_1to5.html > html > body > script
<html lang="en">
<body>
use for loop
use nested loop
-->
<script>
for(var i=1; i<=5; i++){
console.log("Table "+ i);
for(var j=1; j<=10; j++){
var x=i*j;
console.log(i+ " * "+j+'='+x);
}
}
</script>
</body>
</html>


127.0.0.1:5500/Day-5%20for%20loop/multiple_table_1to5.htr
61 x 645
Elements
Console
Sources
Network
top
Filter
2 * 10=20
Table 3
3 * 1=3
3 * 2=6
3 * 3=9
3 * 4=12
3 * 5=15
3 * 6=18
3 * 7=21
3 * 8=24
3 * 9=27
3 * 10=30
Table 4
4 * 1=4
4 * 2=8
4 * 3=12
4 * 4=16
4 * 5=20
4 * 6=24
4 * 7=28
4 * 8=32
4 * 9=36
4 * 10=40


console.log(i+ " * "+j+'='+x);
}
}
</script>
</body>
</html>

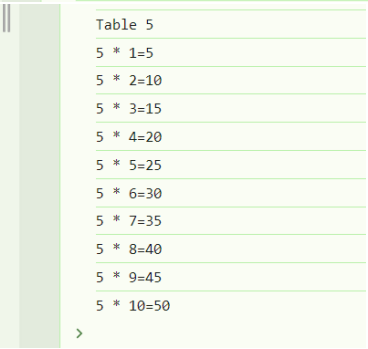


Table 5
5 * 1=5
5 * 2=10
5 * 3=15
5 * 4=20
5 * 5=25
5 * 6=30
5 * 7=35
5 * 8=40
5 * 9=45
5 * 10=50

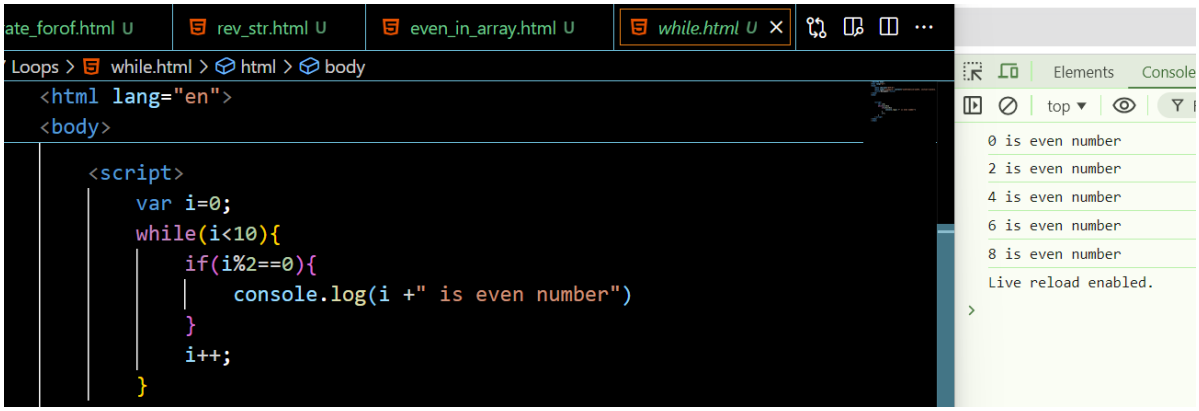
While Loop:

A **while** loop repeats a block of code while a specified condition is true.
The condition is evaluated before each iteration. If it returns true, the loop continues; otherwise, it stops.

Syntax:

```
while (condition) {
        // code to be executed
}
```

Example:



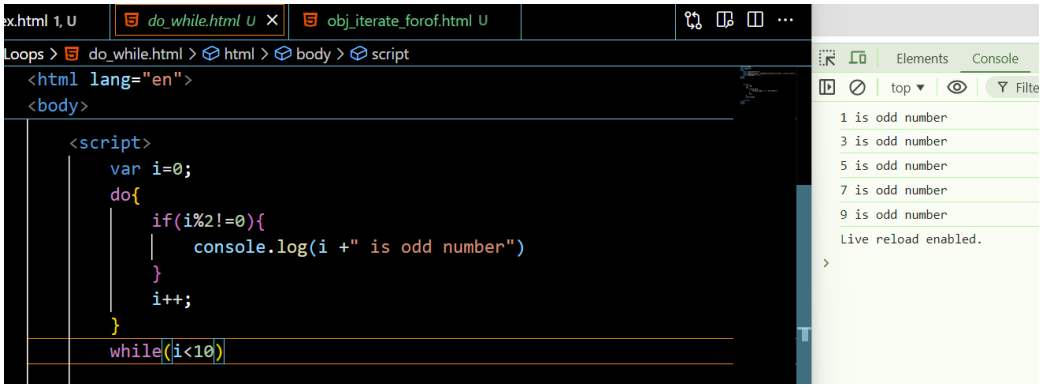
Do While:

Similar to the **while** loop, but it always executes its block of code at least once, even if the condition evaluates to false.
The block of code is executed first, then the condition is evaluated. If true, the loop continues; if false, it stops.

Syntax:

```
do {
// code to be executed
}
while (condition);
```

Example:



For in loop:

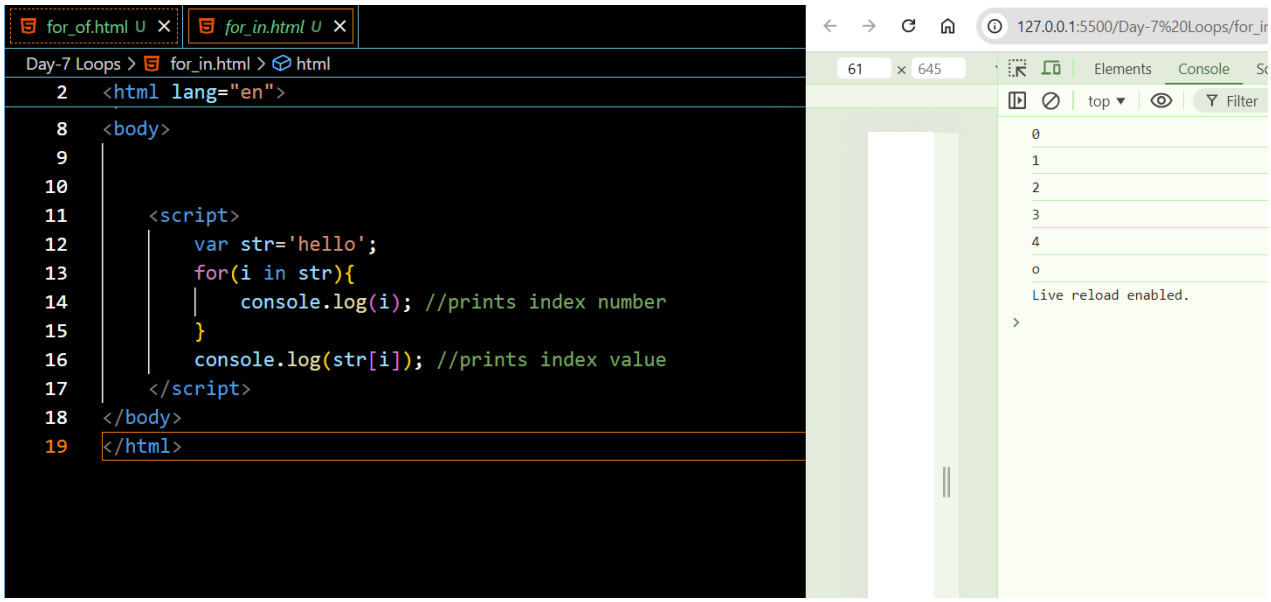
Used to iterate over the properties of an object,array, string. It iterates over enumerable properties of an object, in an arbitrary order.

We can access index and values.

Syntax:

```
for (ref in strname){
      console.log(ref);//indexes
    }
```

Example:



1) **Iterates over Properties**:

- The for...in loop iterates over all enumerable properties of an object.

2) **Order Not Guaranteed**:

- The order of iteration is not guaranteed. It's generally the order in which properties were defined, but this can vary.

3) **Use with Objects**:

- Typically used for objects, not arrays, because it iterates over property names (keys) rather than values.

For of Loop:

Introduced in ES6, it iterates over iterable objects such as arrays, strings, maps, sets, etc.

Syntax:

```
 for (ref of strname){
      console.log(ref);//values
    }
==========================
for (variable of iterable) {
      // code to be executed
}
```

- It provides a more concise syntax compared to the traditional **for** loop for iterating over arrays and other iterable objects.

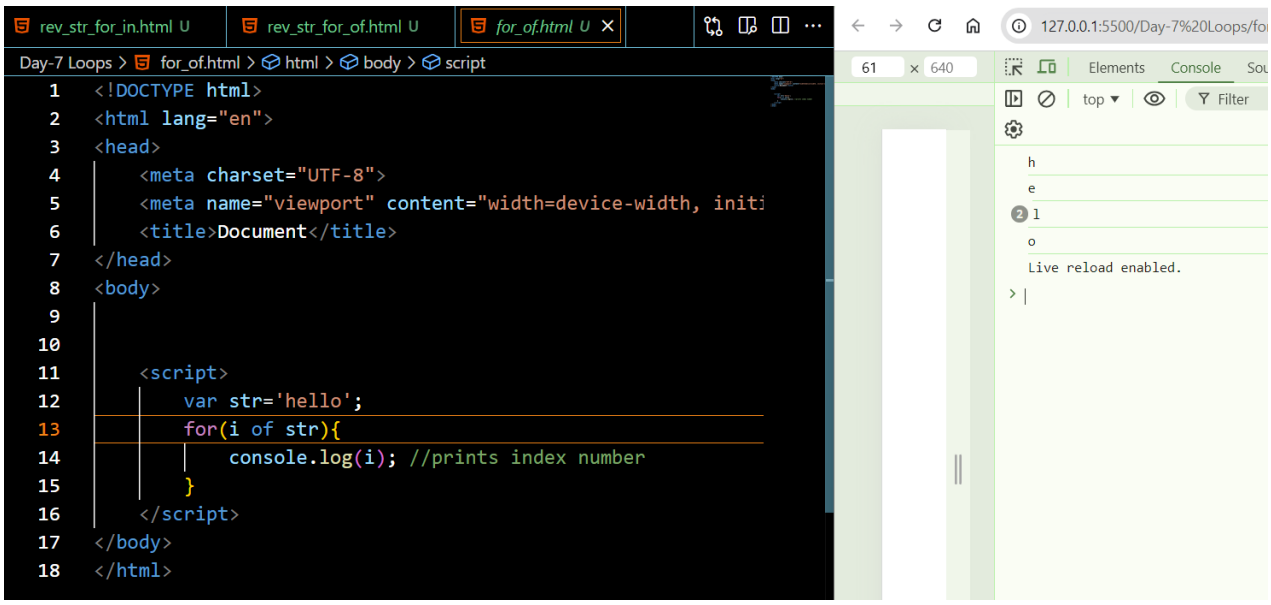
1) **Iterates over Values**:

- The for...of loop iterates over the values of an iterable object.
- This loop does not work with objects unless they implement the iterable protocol.

2) **Use with Arrays and Other Iterables**:

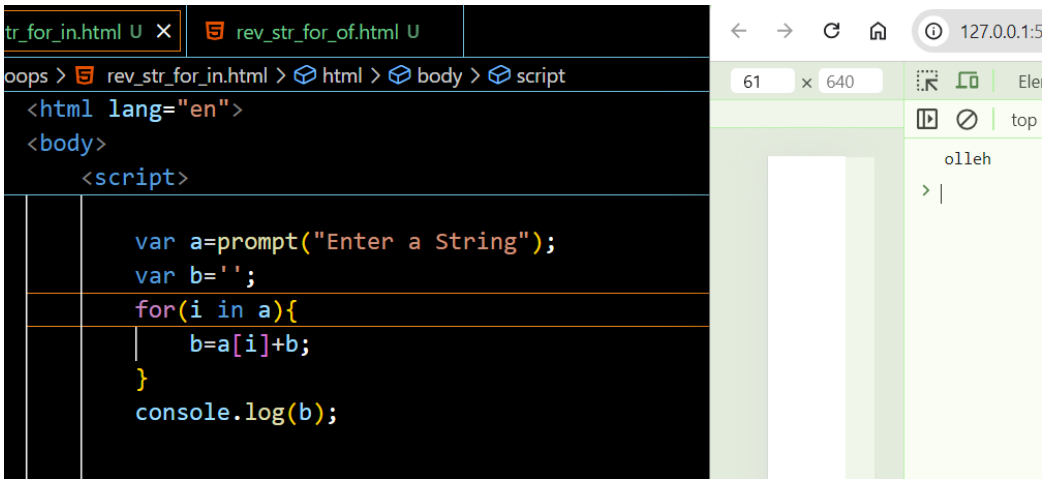
- Commonly used with arrays, strings, maps, sets, and other iterable objects.

Example:



TASK :

1. Reverse a string Input : hello output : olleh



```
<html lang="en">
<body>
    <!-- TASK-1: Reverse a string Input : hello ou
    <script>

        var a=prompt("Enter a String");
        var b='';
        for(i of a){
            b=i+b;
        }
        console.log(b);
```

```
olleh
```

2. Find the even numbers in the array - [23,54,67,64.45,95,98].

```
<!-- Find the even numbers in the array -
 [ 23,54,67,64.45,95,98].
-->
<script>
var a=[ 23,54,67,64,45,95,98];
for(i of a){
    if(i%2==0){
        console.log(i);
    }
}
</script>
</body>
</html>
```

```
54
64
98
Live reload enabled.
```

3. Iterate an object values & keys using a for of loop.

```
<html lang="en">
<body>
    <script>
        const obj = {
        name: "Alice",
        age: 30,
        profession: "Engineer"
        };

    // Using Object.entries() to get key-value pairs
    for (let [key, value] of Object.entries(obj)) {
        console.log(key,':',value);
    }
    </script>
</body>
</html>
```

```
name : Alice
age : 30
profession : Engineer
```