

Task 6

ITI

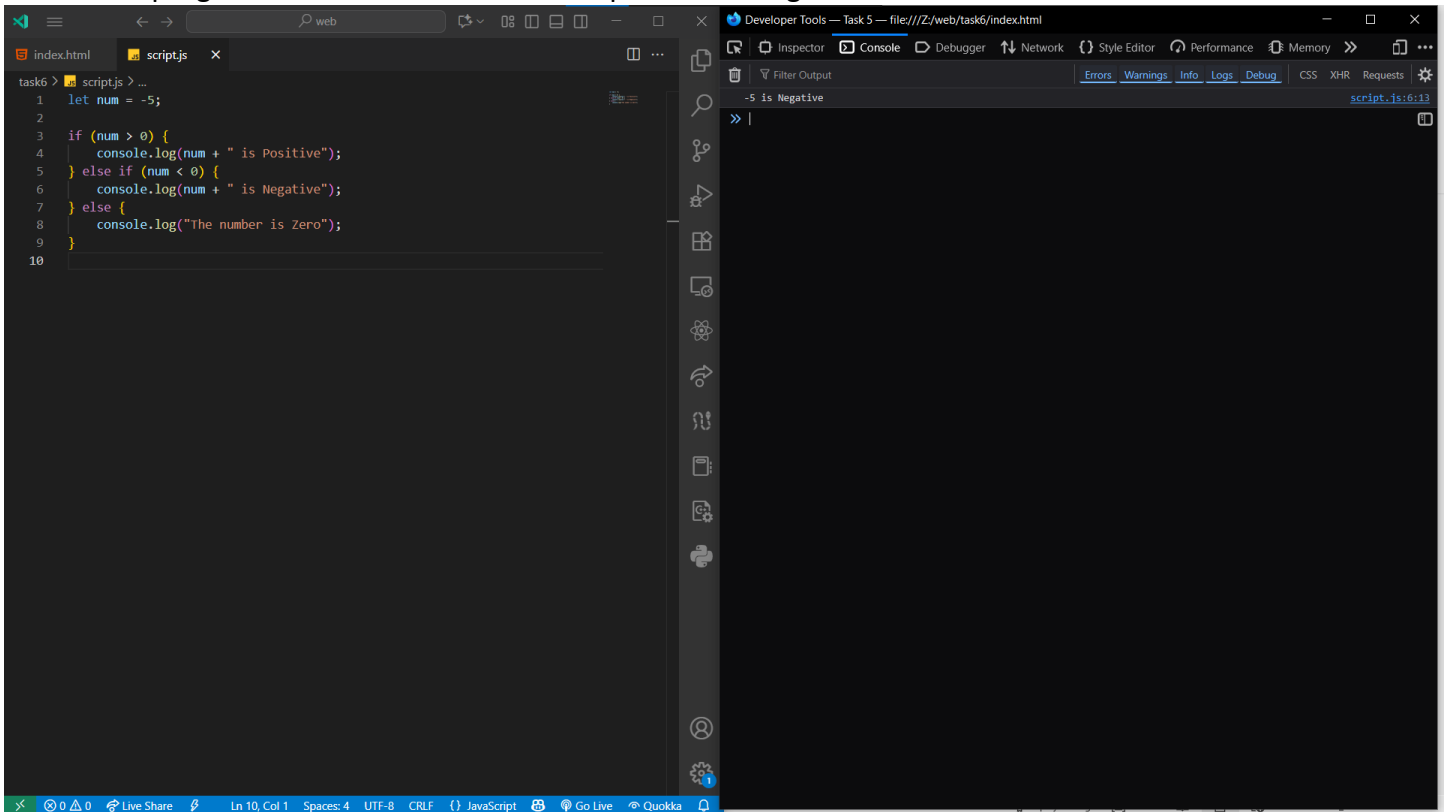
Full Stack

—

Mohamed Salem Mohamed Soliman

—

1. Create a program that checks if a number is positive or negative.



```
task6 > script.js > ...
1 let num = -5;
2
3 if (num > 0) {
4   console.log(num + " is Positive");
5 } else if (num < 0) {
6   console.log(num + " is Negative");
7 } else {
8   console.log("The number is Zero");
9 }
10
```

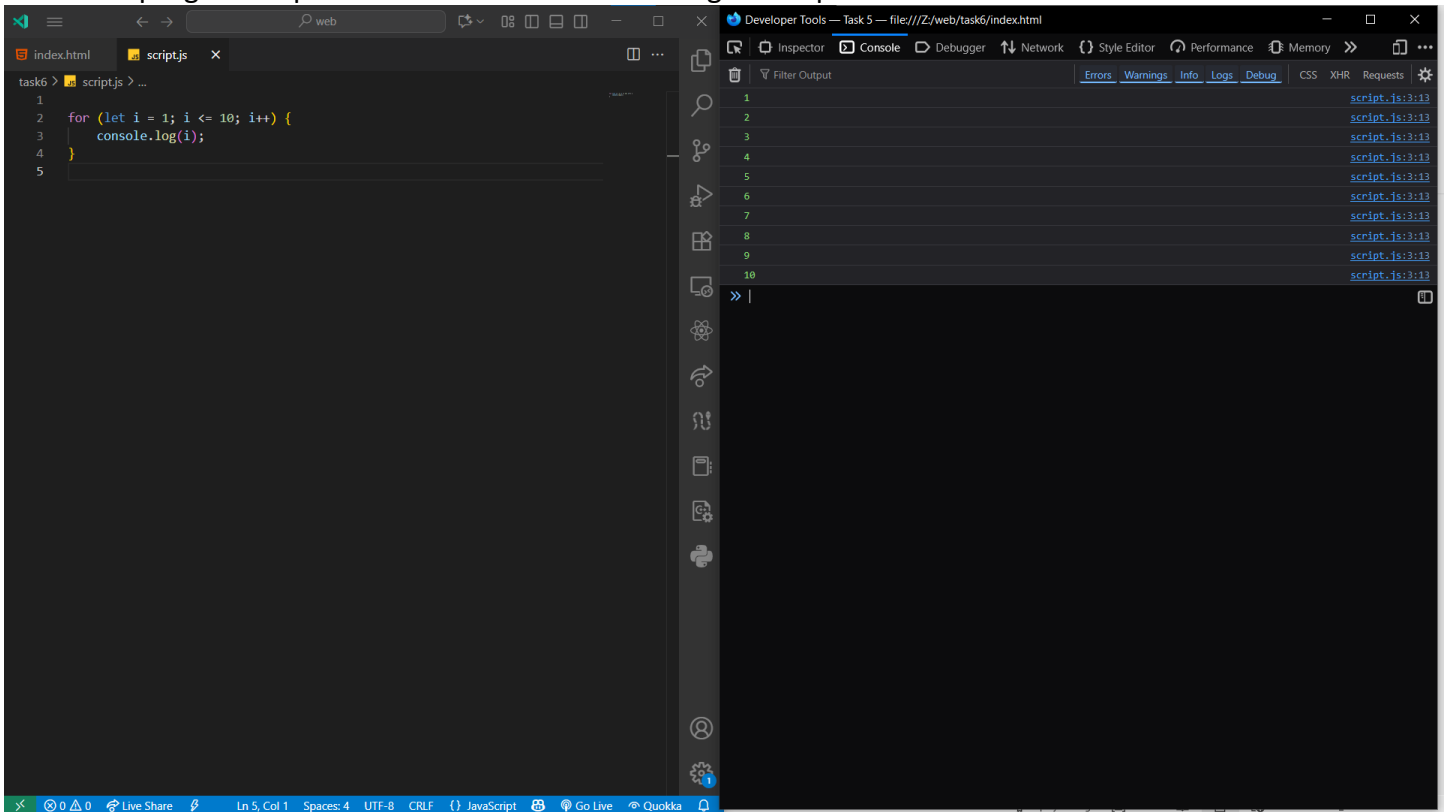
Developer Tools — Task 5 — file:///Z:/web/task6/index.html

Inspector Console Debugger Network Style Editor Performance Memory

Filter Output

-5 is Negative script.js:6:13

2. Write a program to print numbers from 1 to 10 using for loop.



```
task6 > script.js > ...
1
2 for (let i = 1; i <= 10; i++) {
3   console.log(i);
4 }
5
```

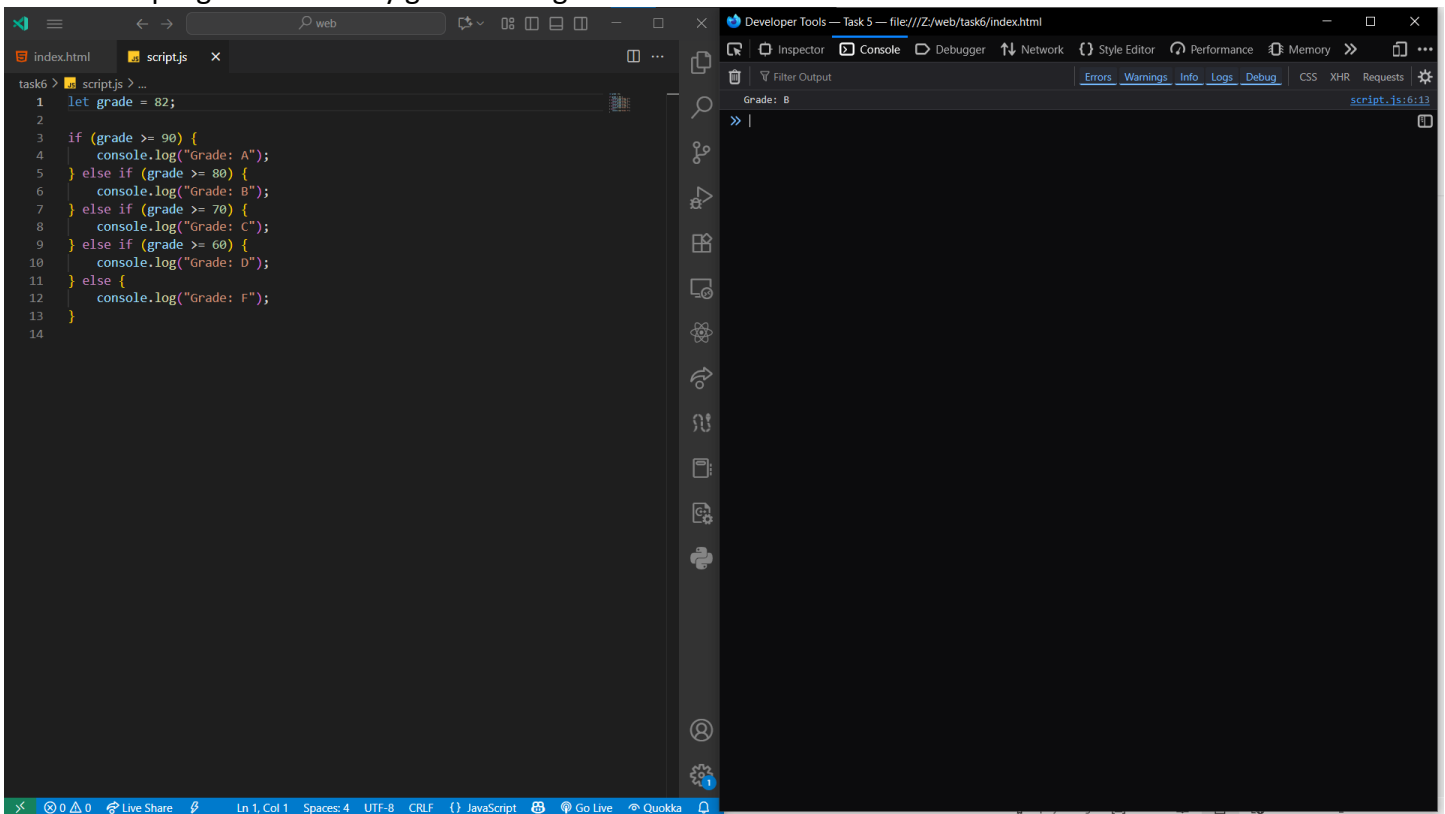
Developer Tools — Task 5 — file:///Z:/web/task6/index.html

Inspector Console Debugger Network Style Editor Performance Memory

Filter Output

1 script.js:3:13
2 script.js:3:13
3 script.js:3:13
4 script.js:3:13
5 script.js:3:13
6 script.js:3:13
7 script.js:3:13
8 script.js:3:13
9 script.js:3:13
10 script.js:3:13

3. Create a program to classify grades using if...else if



```
1 let grade = 82;
2
3 if (grade >= 90) {
4   console.log("Grade: A");
5 } else if (grade >= 80) {
6   console.log("Grade: B");
7 } else if (grade >= 70) {
8   console.log("Grade: C");
9 } else if (grade >= 60) {
10  console.log("Grade: D");
11 } else {
12  console.log("Grade: F");
13 }
14
```

Developer Tools — Task 5 — file:///Z:/web/task6/index.html

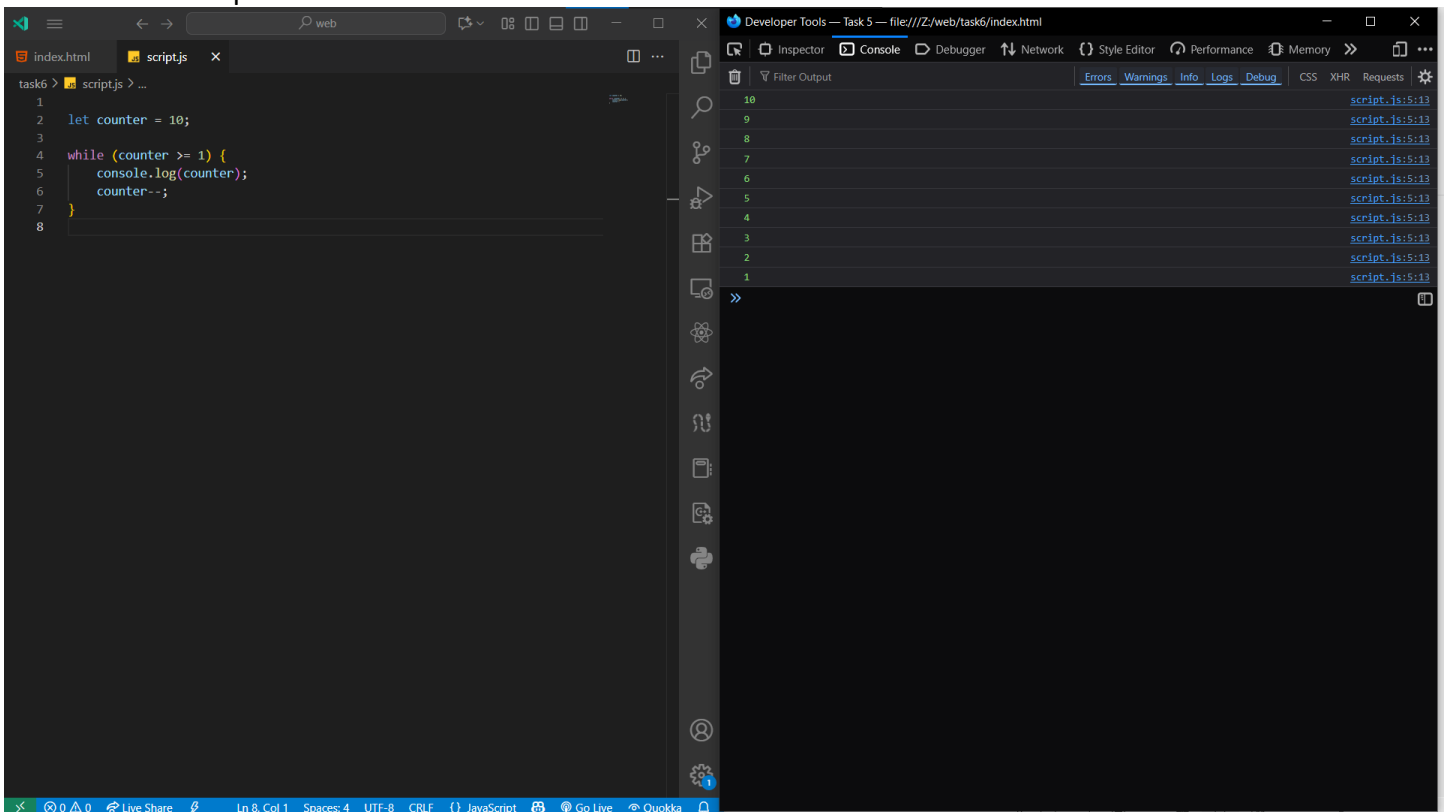
Inspector Console Debugger Network Style Editor Performance Memory

Filter Output

Errors Warnings Info Logs Debug CSS XHR Requests

Grade: B script.js:13

4. Use a while loop to count down from 10 to 1.



```
1
2 let counter = 10;
3
4 while (counter >= 1) {
5   console.log(counter);
6   counter--;
7 }
8
```

Developer Tools — Task 5 — file:///Z:/web/task6/index.html

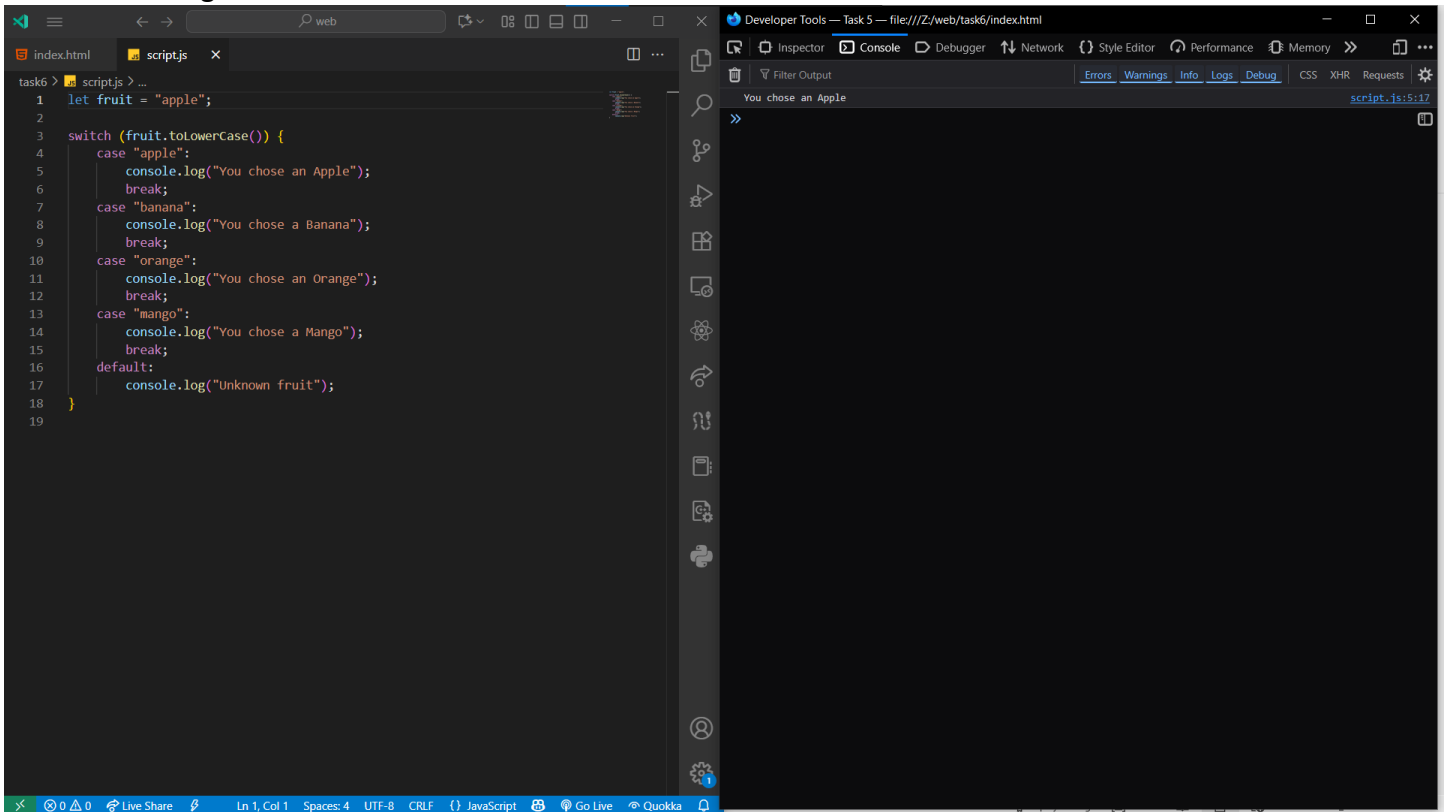
Inspector Console Debugger Network Style Editor Performance Memory

Filter Output

Errors Warnings Info Logs Debug CSS XHR Requests

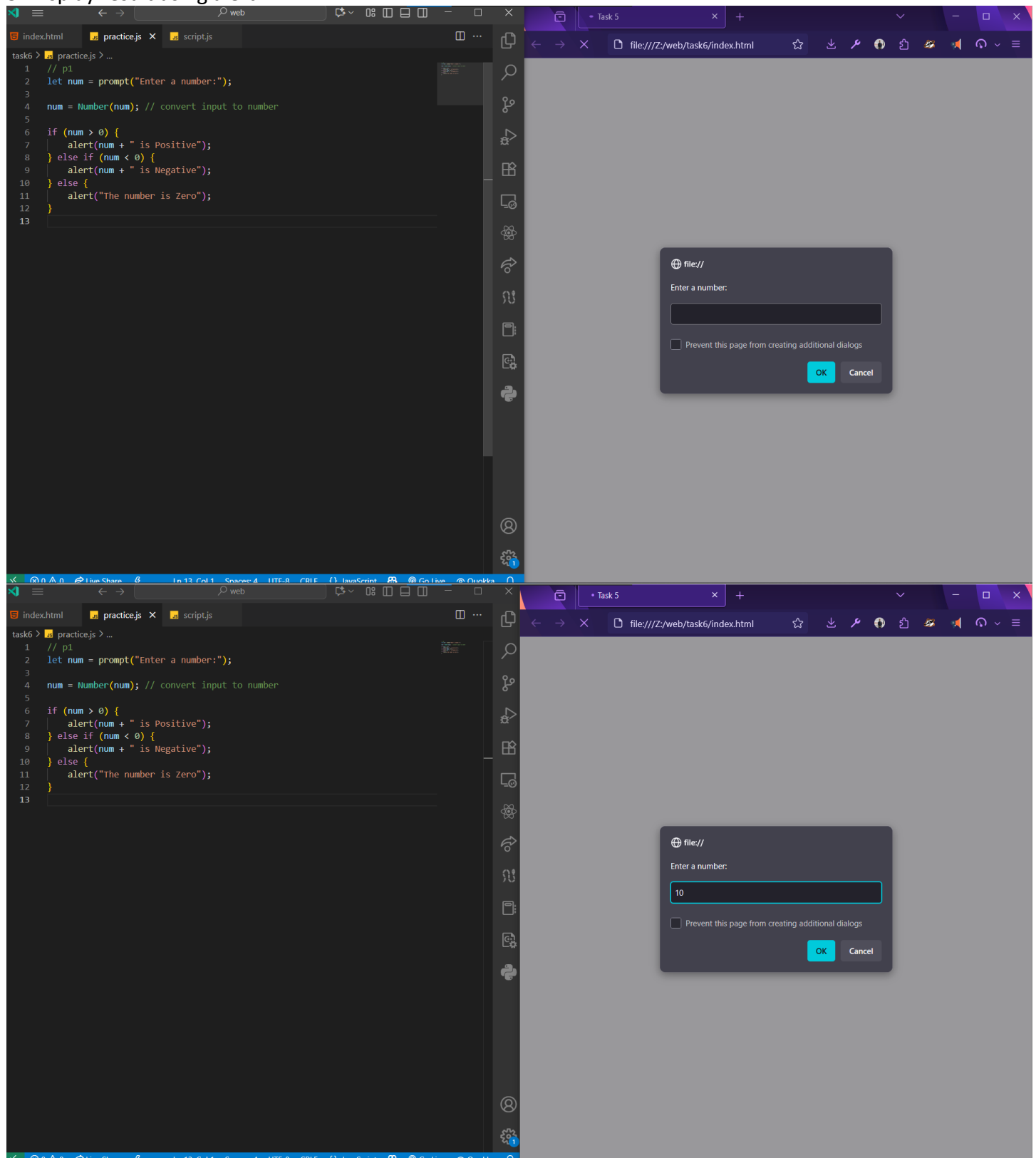
10 script.js:5:13
9 script.js:5:13
8 script.js:5:13
7 script.js:5:13
6 script.js:5:13
5 script.js:5:13
4 script.js:5:13
3 script.js:5:13
2 script.js:5:13
1 script.js:5:13

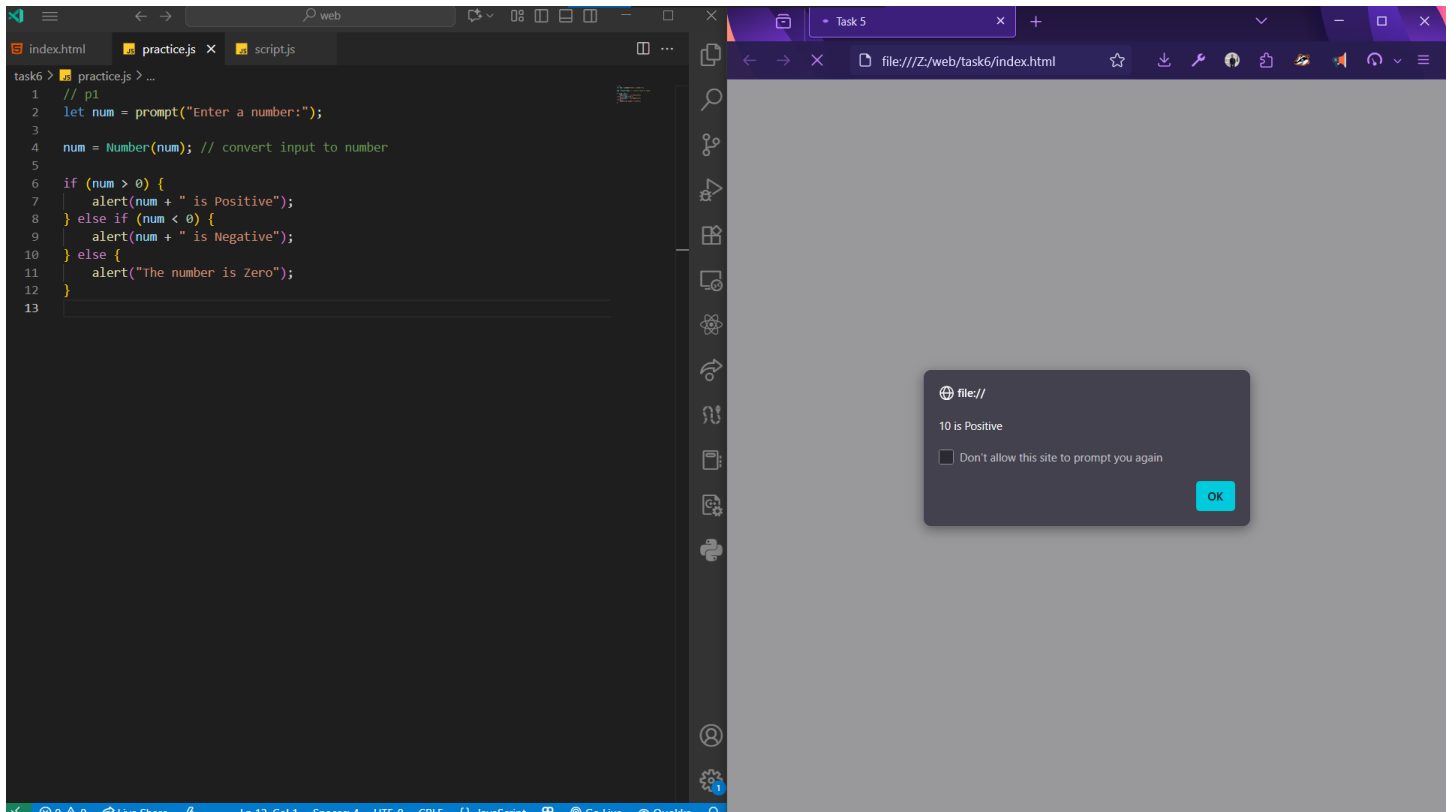
5. Practice using switch to detect fruit names.



Practical 1: Number Check

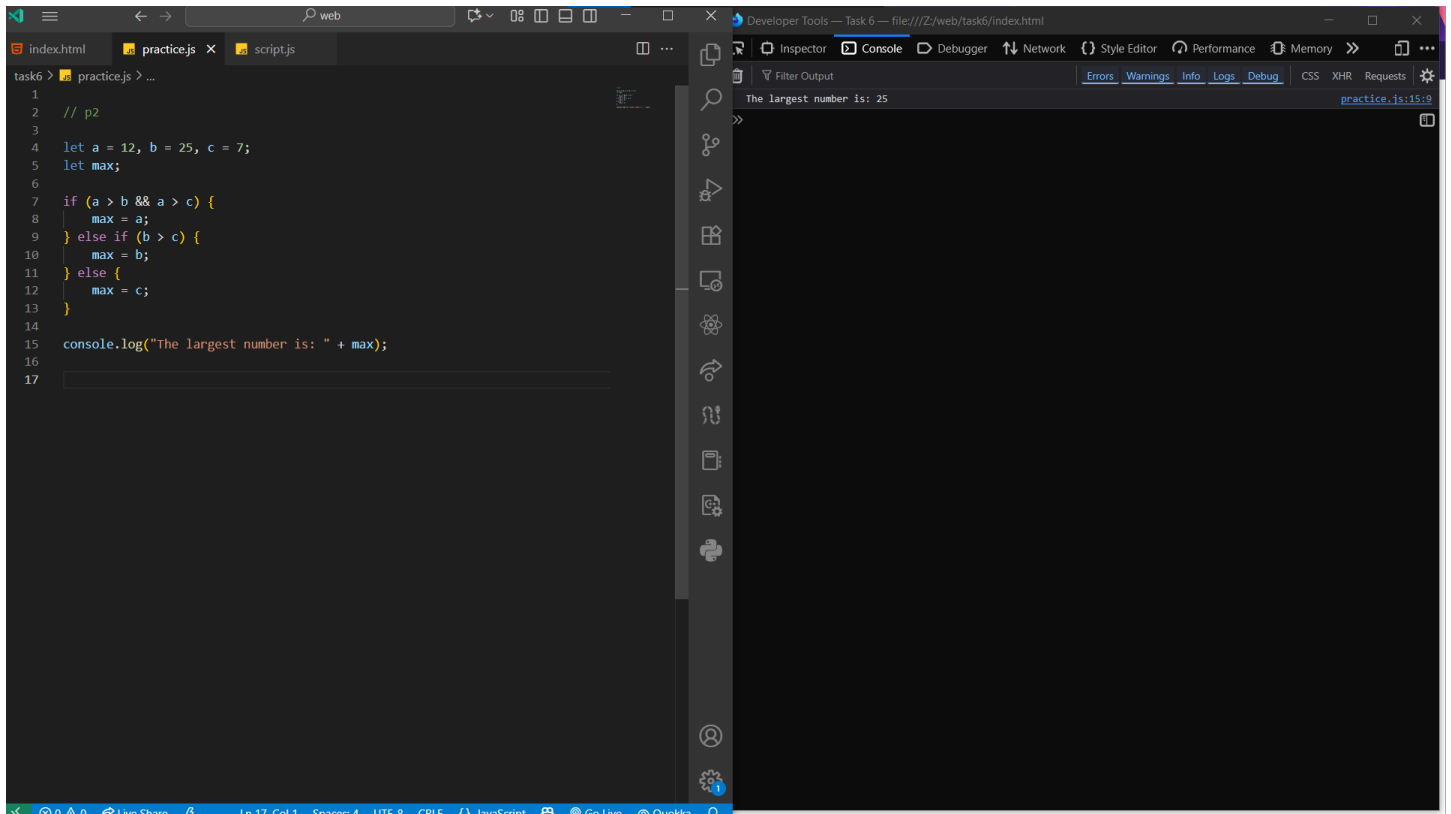
1. Prompt user for number.
2. Use if-else to check if positive, negative or zero.
3. Display result using alert.





Practical 2: Largest of 3

USE IF-ELSE-IF TO COMPARE THREE NUMBERS. STORE RESULT IN 'MAX'. PRINT USING CONSOLE.LOG.

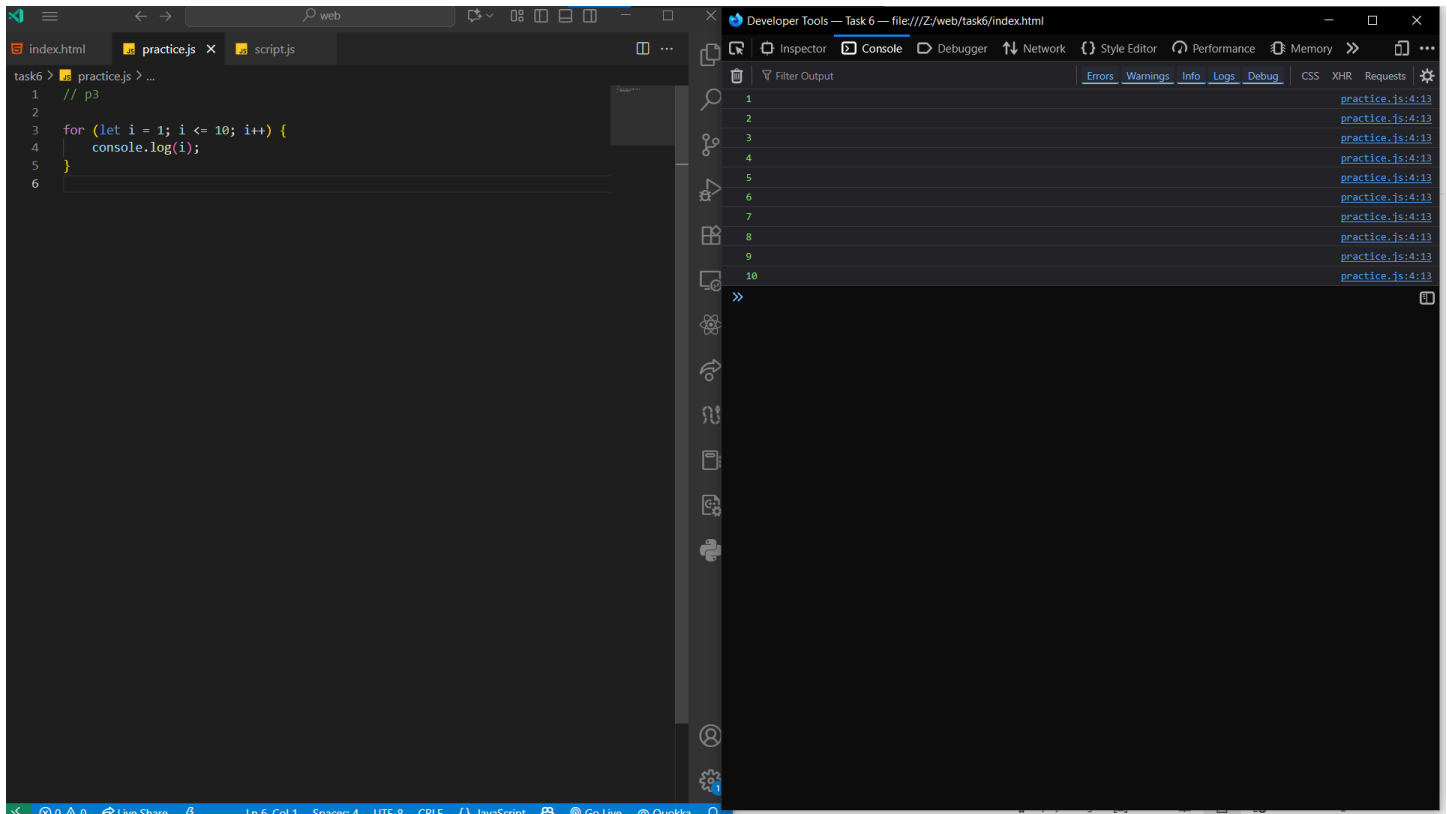


```
1
2 // p2
3
4 let a = 12, b = 25, c = 7;
5 let max;
6
7 if (a > b && a > c) {
8   max = a;
9 } else if (b > c) {
10   max = b;
11 } else {
12   max = c;
13 }
14
15 console.log("The largest number is: " + max);
16
17
```

The screenshot shows a web browser window with the developer tools open. The console tab is active, displaying the output of the JavaScript code: "The largest number is: 25". The code in the editor uses if-else-if statements to compare three numbers (a=12, b=25, c=7) and store the maximum value in the variable 'max'. The result is then printed to the console using console.log.

Practical 3: For Loop 1-10

USE FOR LOOP: I FROM 1 TO 10. PRINT EACH NUMBER USING CONSOLE.LOG.



The screenshot shows a web browser window with the developer tools open. The console tab is active, displaying the output of a JavaScript script. The script uses a for loop to iterate from 1 to 10, logging each number to the console. The output shows the numbers 1 through 10, each followed by a timestamp and the file path.

```
task6 > practice.js > ...  
1 // p3  
2  
3 for (let i = 1; i <= 10; i++) {  
4   console.log(i);  
5 }  
6
```

Developer Tools — Task 6 — file:///Z:/web/task6/index.html

Inspector Console Debugger Network Style Editor Performance Memory

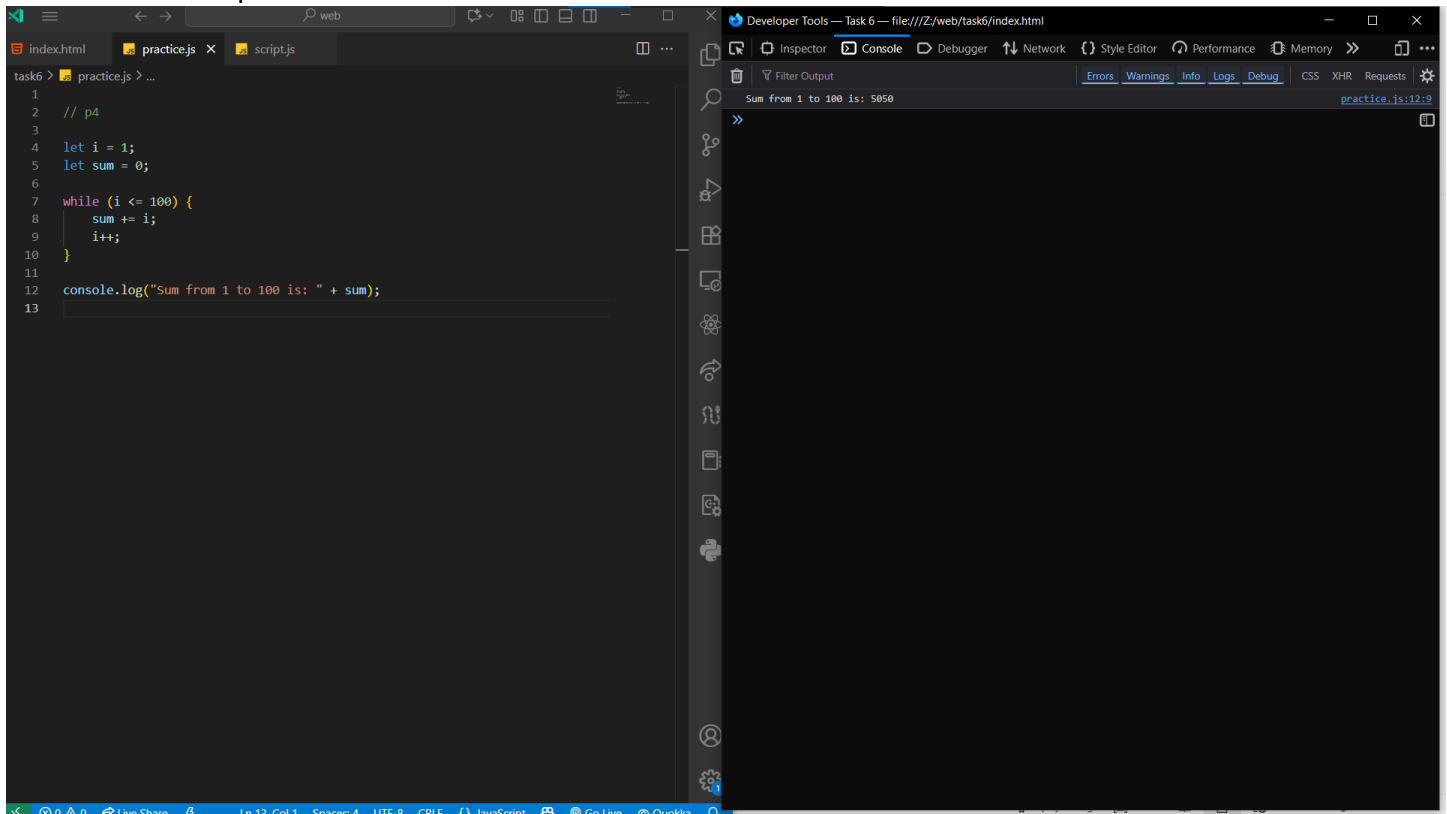
Filter Output

Errors Warnings Info Logs Debug CSS XHR Requests

1 practice.js:4:13
2 practice.js:4:13
3 practice.js:4:13
4 practice.js:4:13
5 practice.js:4:13
6 practice.js:4:13
7 practice.js:4:13
8 practice.js:4:13
9 practice.js:4:13
10 practice.js:4:13

Practical 4: While Sum 1-100

Use Use while loop to sum numbers from 1 to 100. Initialize Initialize i and sum variables. Print Print total sum.



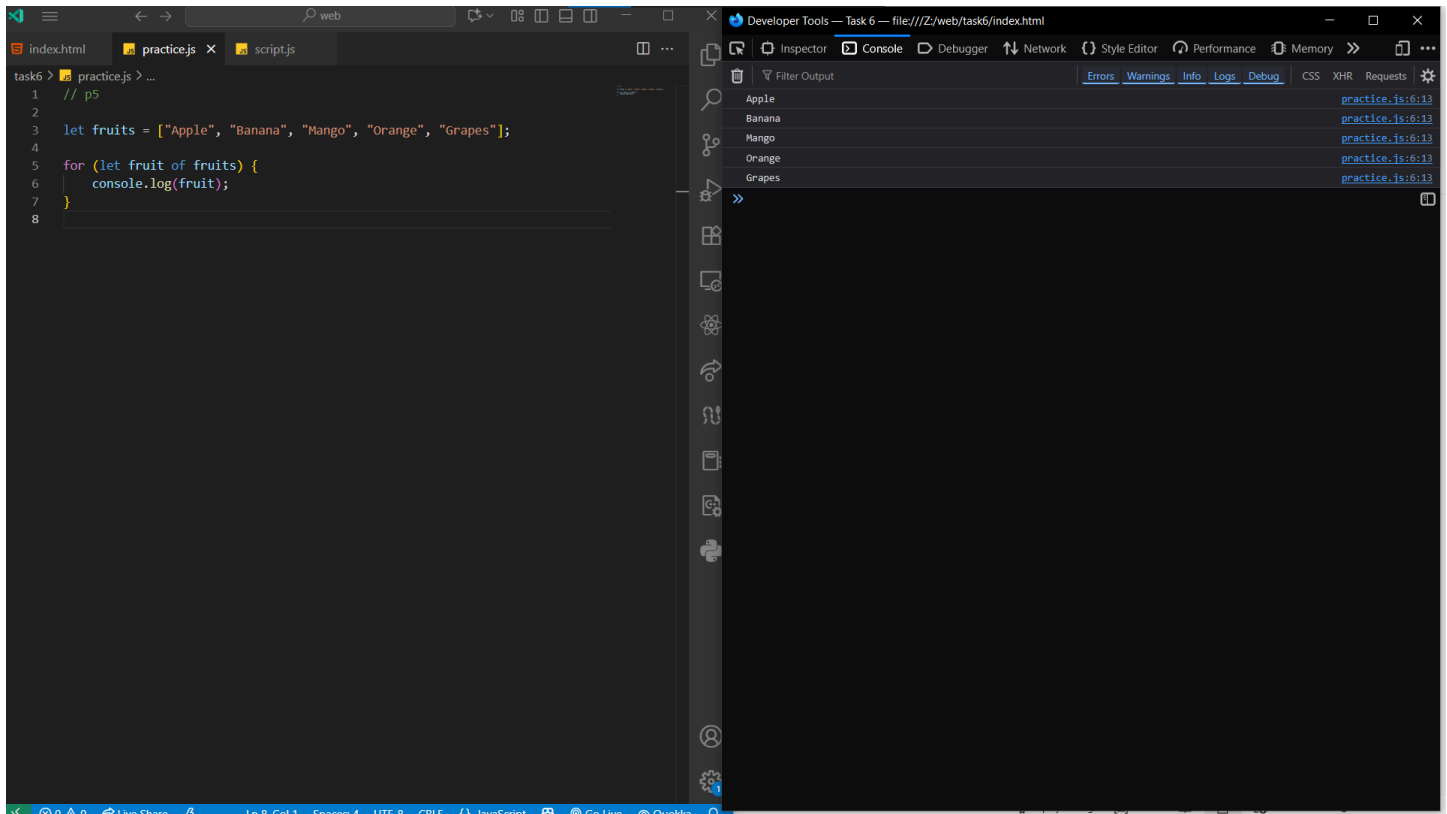
The screenshot shows a web browser window with the developer tools open. The left pane displays the source code of a file named `task6/index.html`. The code is as follows:

```
1
2 // p4
3
4 let i = 1;
5 let sum = 0;
6
7 while (i <= 100) {
8     sum += i;
9     i++;
10 }
11
12 console.log("Sum from 1 to 100 is: " + sum);
13
```

The right pane of the developer tools shows the console output, which reads: "Sum from 1 to 100 is: 5050". The console also indicates the log message was sent from `practice.js:12:9`.

Practical 5: Loop Array

CREATE ARRAY OF FRUITS. USE FOR...OF TO ITERATE. PRINT EACH FRUIT.



The screenshot shows a web browser window with the developer tools open. The console displays the output of a JavaScript script. The script defines an array of fruits and uses a for...of loop to iterate over each fruit and log it to the console.

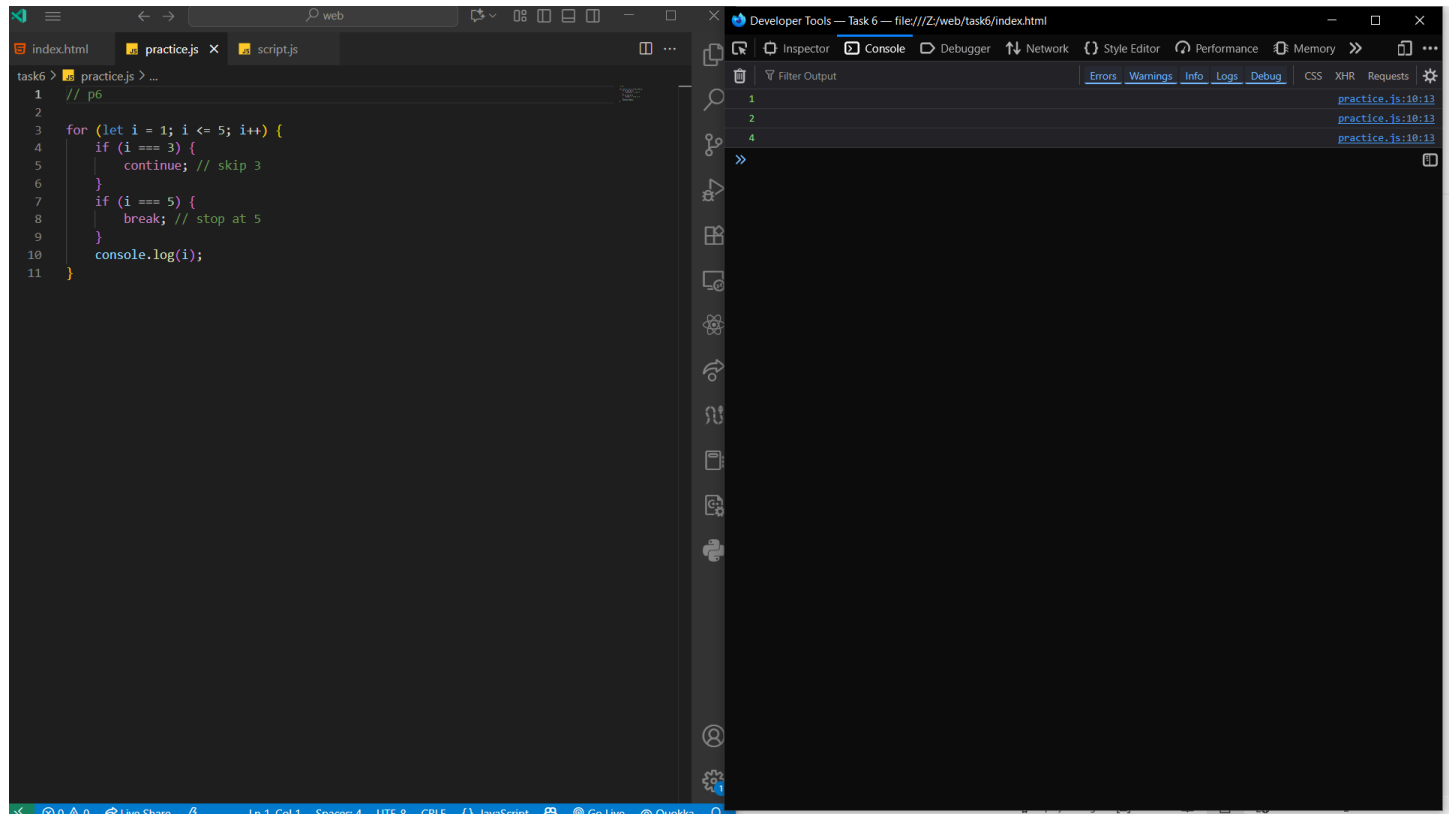
```
task6 > practice.js > ...  
1 // p5  
2  
3 let fruits = ["Apple", "Banana", "Mango", "Orange", "Grapes"];  
4  
5 for (let fruit of fruits) {  
6   console.log(fruit);  
7 }  
8
```

The console output shows the following fruits being logged:

- Apple
- Banana
- Mango
- Orange
- Grapes

Practical 6: Break & Continue

USE FOR LOOP FROM 1 TO 5. SKIP 3 WITH CONTINUE. STOP AT 5 WITH BREAK.



The screenshot shows a web browser window with the developer tools open. The left pane displays the source code of a file named `task6`. The code is as follows:

```
1 // p6
2
3 for (let i = 1; i <= 5; i++) {
4   if (i === 3) {
5     continue; // skip 3
6   }
7   if (i === 5) {
8     break; // stop at 5
9   }
10  console.log(i);
11 }
```

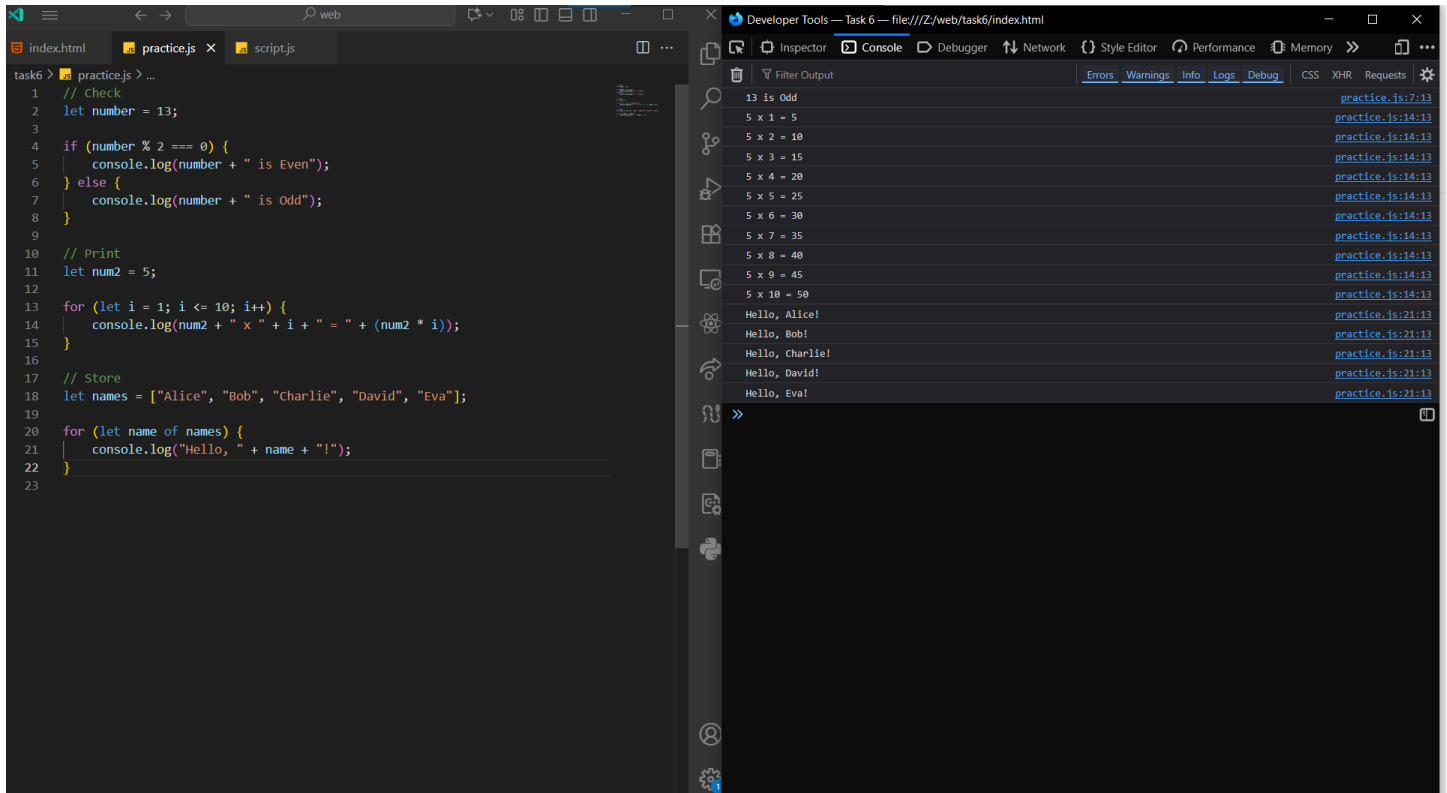
The right pane of the developer tools shows the console output, which contains three log entries:

```
1 practice.js:10:13
2 practice.js:10:13
3 practice.js:10:13
```

The first two log entries correspond to the values 1 and 2, which were printed when the loop iteration for `i=3` was skipped. The third log entry corresponds to the value 4, which was printed after the loop iteration for `i=5` was terminated by the `break` statement. The value 5 was not printed.

Student Tasks

Store 5 names in array and display with for...of. Print multiplication table of 5. Check if number is even or odd.



The screenshot shows a web browser window with the developer tools open. The left pane displays the source code of a file named `task6/index.html`. The code is as follows:

```
1 // Check
2 let number = 13;
3
4 if (number % 2 === 0) {
5   console.log(number + " is Even");
6 } else {
7   console.log(number + " is Odd");
8 }
9
10 // Print
11 let num2 = 5;
12
13 for (let i = 1; i <= 10; i++) {
14   console.log(num2 + " x " + i + " = " + (num2 * i));
15 }
16
17 // Store
18 let names = ["Alice", "Bob", "Charlie", "David", "Eva"];
19
20 for (let name of names) {
21   console.log("Hello, " + name + "!");
22 }
23
```

The right pane shows the console output, which includes the following messages:

- 13 is Odd
- 5 x 1 = 5
- 5 x 2 = 10
- 5 x 3 = 15
- 5 x 4 = 20
- 5 x 5 = 25
- 5 x 6 = 30
- 5 x 7 = 35
- 5 x 8 = 40
- 5 x 9 = 45
- 5 x 10 = 50
- Hello, Alice!
- Hello, Bob!
- Hello, Charlie!
- Hello, David!
- Hello, Eva!