MODULE:4(JavaScript Basic & DOM)

- **What is JavaScript?**

Answer: JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side scripts to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

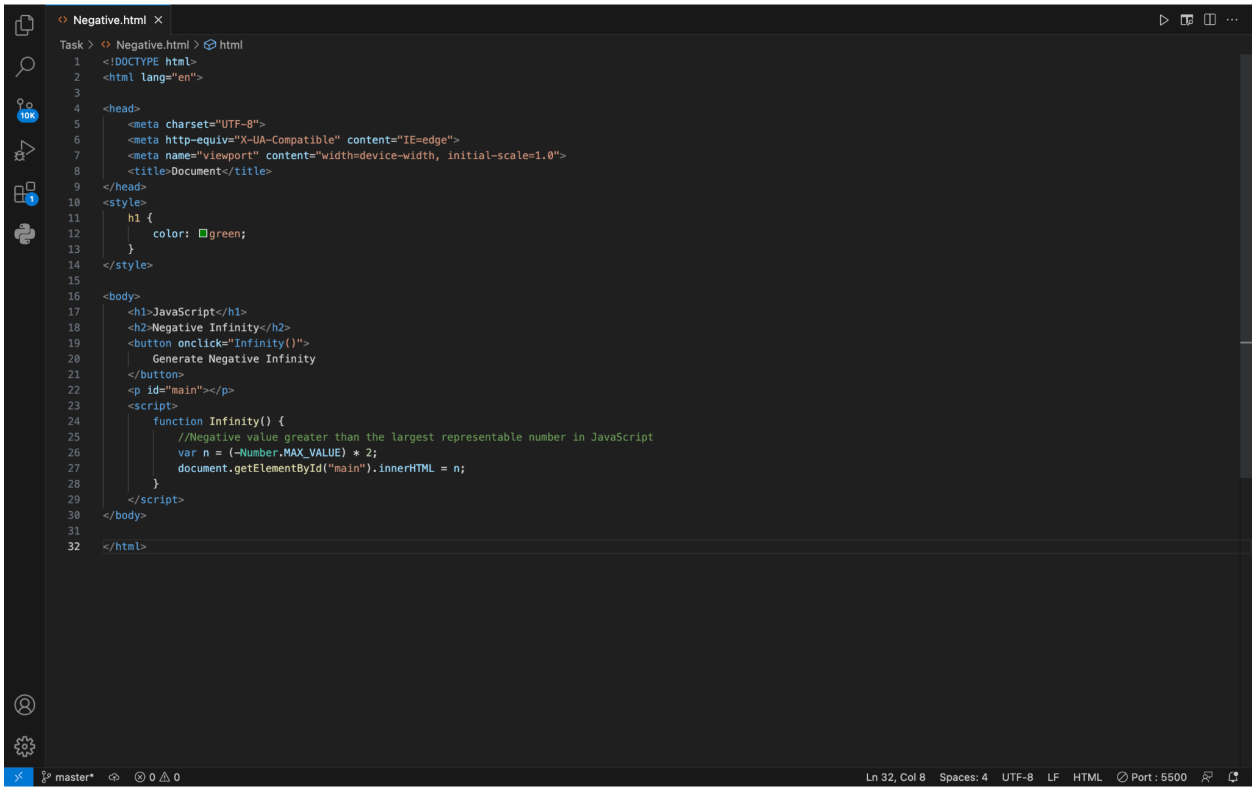
- **What is the use of isNaN function?**

Answer: In JavaScript NaN is short for "Not-a-Number".The isNaN() method returns true if a value is NaN.The isNaN() method converts the value to a number before testing it.
Syntax:

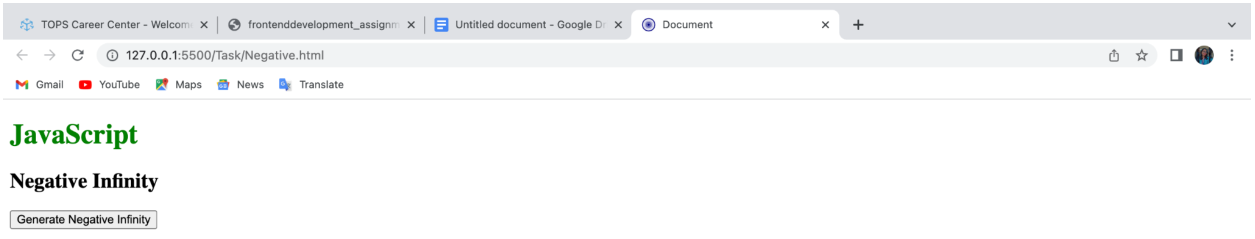
```
isNan(value)
```

- **What is negative Infinity?**

Answer: The negative infinity in JavaScript is a constant value that is used to represent a value that is the lowest available. This means that no other number is lesser than this value. It can be generated using a self-made function or by an arithmetic operation.
For example:



```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<style>
    h1 {
        color: green;
    }
</style>

<body>
    <h1>JavaScript</h1>
    <h2>Negative Infinity</h2>
    <button onclick="Infinity()">
        Generate Negative Infinity
    </button>
    <p id="main"></p>
    <script>
        function Infinity() {
            //Negative value greater than the largest representable number in JavaScript
            var n = (-Number.MAX_VALUE) * 2;
            document.getElementById("main").innerHTML = n;
        }
    </script>
</body>

</html>
```



127.0.0.1:5500/Task/Negative.html

JavaScript

Negative Infinity

Generate Negative Infinity

- **Which company developed JavaScript?**

Answer: JavaScript was invented by Brendan Eich in 1995. It was developed by Netscape. It can be used to program web browsers or even servers.

- **What are undeclared and undefined variables?**

Answer:
Undeclared Variables: Undeclared is a variable that has not been properly declared using var or let.
For example:

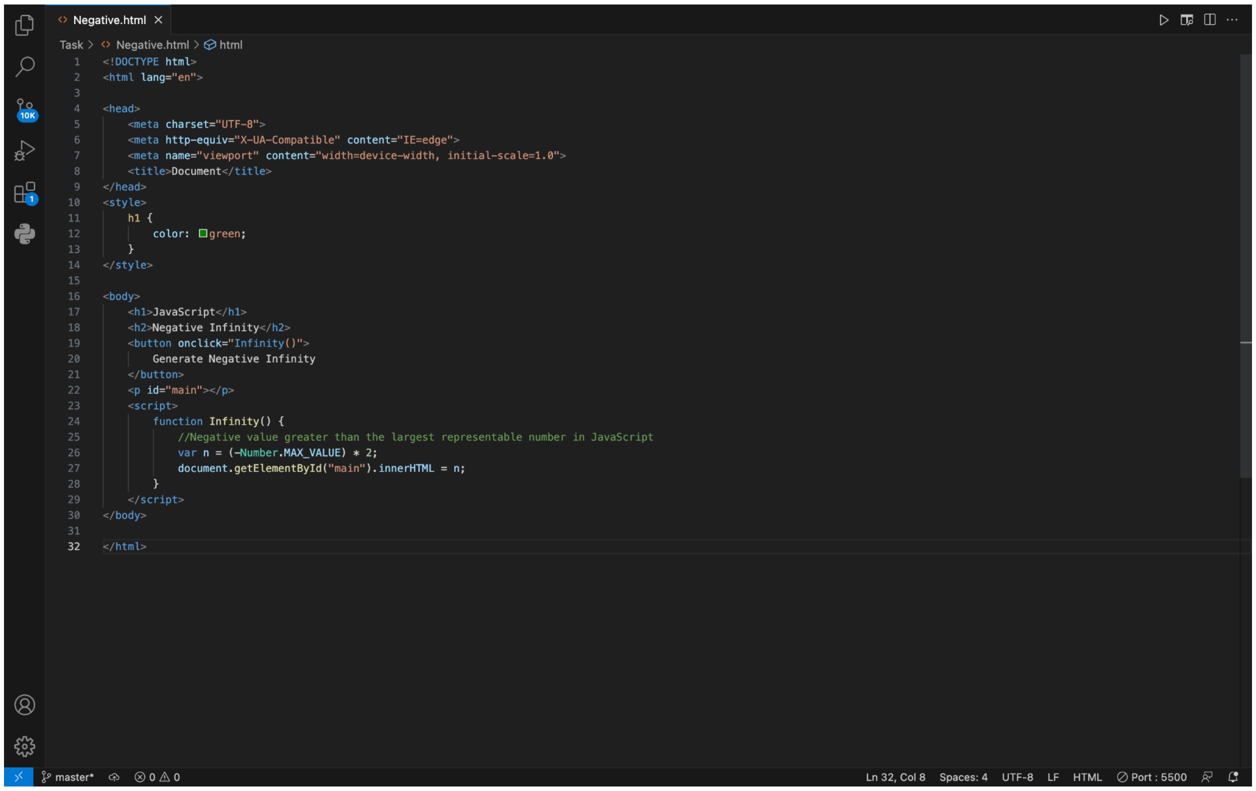
```
   <script>
       console.log(Hill);
       //Undeclared
   </script>
```

Undefined Variables: Undefined is a variable declared but that has not been assigned any value.
For example:

```
   <script>
       let Hill;
       console.log(Hill);
       //Undefined
   </script>
```

- **Write the code for adding new elements dynamically?**

Answer:



```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<style>
    h1 {
        color: green;
    }
</style>

<body>
    <h1>JavaScript</h1>
    <h2>Negative Infinity</h2>
    <button onclick="Infinity()">
        Generate Negative Infinity
    </button>
    <p id="main"></p>
    <script>
        function Infinity() {
            //Negative value greater than the largest representable number in JavaScript
            var n = (-Number.MAX_VALUE) * 2;
            document.getElementById("main").innerHTML = n;
        }
    </script>
</body>

</html>
```

- **What is the difference between ViewState and SessionState?**

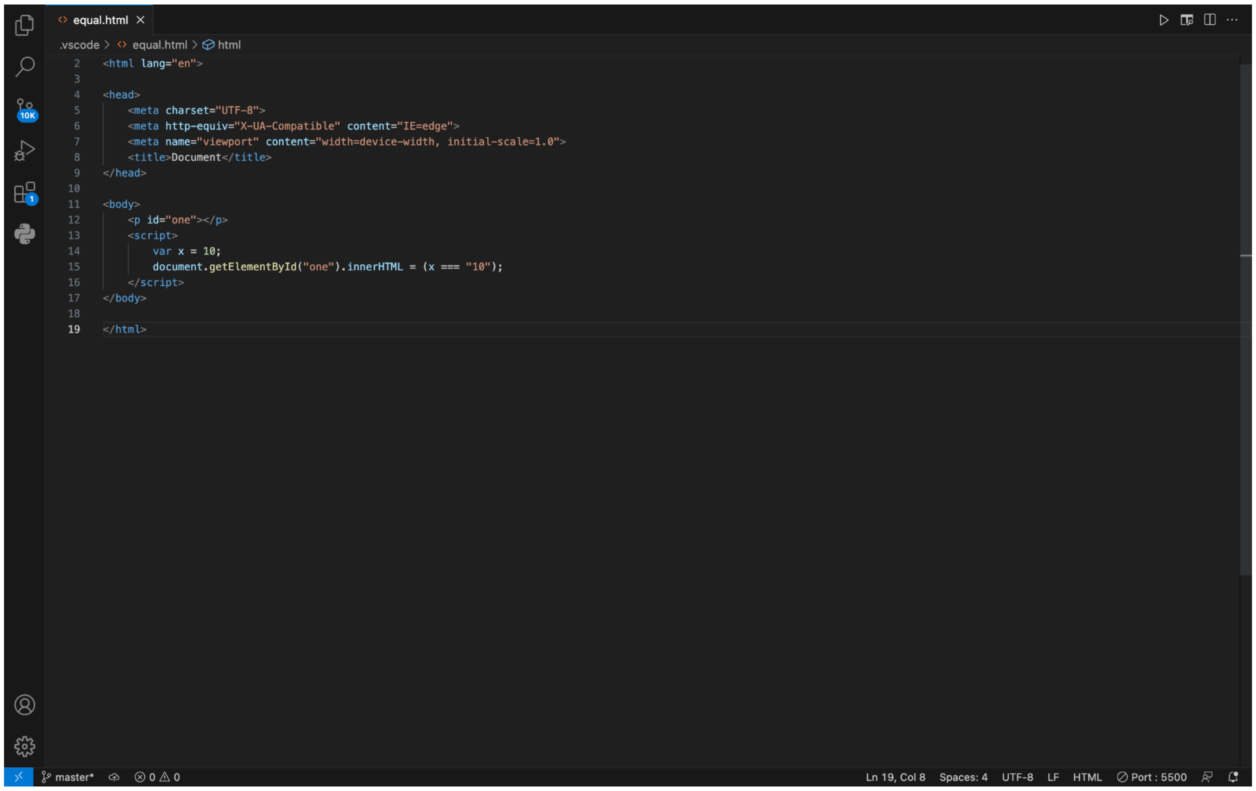
Answer:

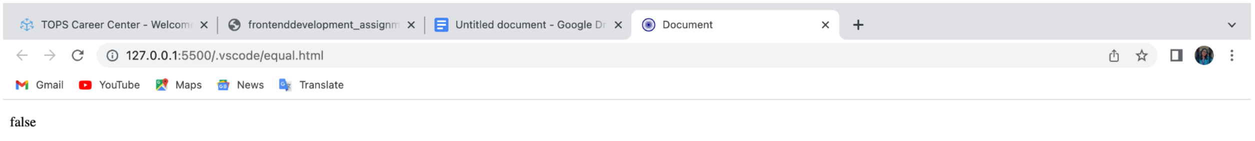
ViewState	SessionState
1.Maintained at page level only.	1.Maintained at session level.
2.View state can only be visible from a single page and not multiple pages.	2.Session state value availability is across all pages available in a user session.
3.It will retain values in the event of a postback operation occurring.	3.In session state, user data remains in the server. Data is available to the user until the

	browser is closed or there is session expiration.
4.Information is stored on the client’s end only.	4.Information is stored on the server.
5.Used to allow the persistence of page-instance-specific data.	5.Used for the persistence of user-specific data on the server’s end.
6.ViewState values are lost/cleared when a new page is loaded.	6.SessionState can be cleared by programmer or user or in case of timeouts.

- **What is === operator?**

Answer: === (Triple equals) is a strict equality comparison operator in JavaScript, which returns false for the values which are not of a similar type. This operator performs type casting for equality. If we compare 2 with “2” using ===, then it will return a false value.
For example:


```html
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>

<body>
    <p id="one"></p>
    <script>
        var x = 10;
        document.getElementById("one").innerHTML = (x === "10");
    </script>
</body>

</html>
```


127.0.0.1:5500/.vscode/equal.html

```
false
```

- **How can the style/class of an element be changed?**

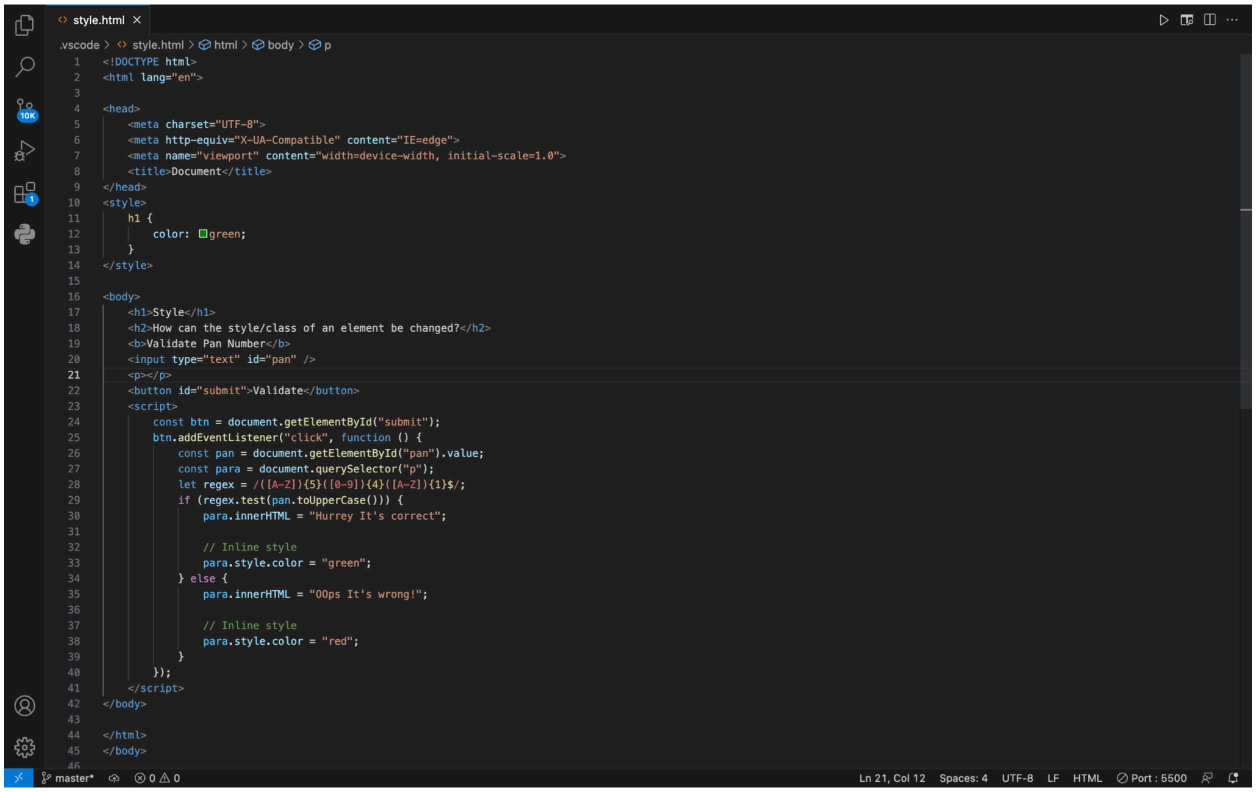
Answer: 1. Changing CSS with the help of the style property:
Syntax:

```
document.getElementById("id").style.property = new_style
```

2. The className Property: This property is used to set the current class of the element to the specified class.
Syntax:

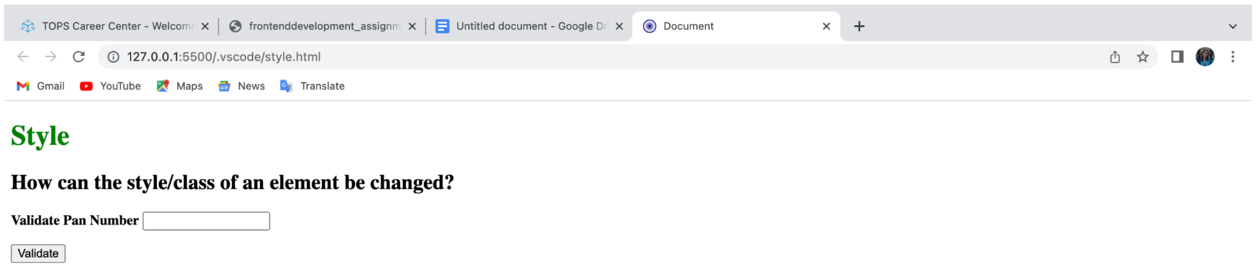
```
document.getElementById("id").className = class
```

For example:



```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<style>
    h1 {
        color: green;
    }
</style>

<body>
    <h1>Style</h1>
    <h2>How can the style/class of an element be changed?</h2>
    <b>Validate Pan Number</b>
    <input type="text" id="pan" />
    <p></p>
    <button id="submit">Validate</button>
    <script>
        const btn = document.getElementById("submit");
        btn.addEventListener("click", function () {
            const pan = document.getElementById("pan").value;
            const para = document.querySelector("p");
            let regex = /([A-Z]){5}([0-9]){4}([A-Z]){1}$/;
            if (regex.test(pan.toUpperCase())) {
                para.innerHTML = "Hurrey It's correct";

                // Inline style
                para.style.color = "green";
            } else {
                para.innerHTML = "OOps It's wrong!";

                // Inline style
                para.style.color = "red";
            }
        });
    </script>
</body>

</html>
</body>
```

style.html
.vscode > style.html > html > body > p
master* 0 0
Ln 21, Col 12 Spaces: 4 UTF-8 LF HTML Port : 5500


TOPS Career Center - Welcom
frontenddevelopment_assignm
Untitled document - Google D
Document
127.0.0.1:5500/.vscode/style.html
Gmail
YouTube
Maps
News
Translate

Style

How can the style/class of an element be changed?

Validate Pan Number

Validate

- **How to read and write a file using JavaScript?**

Answer: Write operation on a file:

After the File System file is imported then, the writeFile() operation is called. The writeFile() method is used to write into the file in JavaScript. The syntax of this method is as follows −

writeFile(path,inputData,callBackFunction)

The writeFile() function accepts three parameters −

- Path − The first parameter is the path of the file or the name of the file into which the input data is to be written.

  If there is a file already, then the contents in the file are deleted and the input which is given by the user will get updated or if the file is not present, then the file with that will be created in the given path and the input information is written into it.

- inputData − The second parameter is the input data which contains the data to be written in the file that is opened.
- callBackFuntion − The third parameter is the function which is the call back function which takes the error as the parameter and shows the fault if the write operation fails.

For example:

```
const fs = require('fs')

let fInput = "You are reading the content from Tutorials Point"

fs.writeFile('tp.txt', fInput, (err) => {

    if (err) throw err;

    else {

        console.log("The file is updated with the given data")

    }

})
```

Reading from the file:

After the File System module is imported, the reading of the file in JavaScript can be done by using the readFile() function.

Syntax:

The syntax to read from a file is as follows −

readFile(path, format, callBackFunc)

The readFile() function accepts three parameters including one optional parameter.

- Path − The first parameter is the path of the test file from which the contents are to read. If the current location or directory is the same directory where the file which is to be opened and read is located then, only the file name has to be given.
- Format − The second parameter is the optional parameter which is the format of the text file. The format can be ASCII, utf-8 etc.
- CallBackFunc − The third parameter is the call back function which takes the error as the parameter and displays if the fault is any raised due to the error.

For example:

```
const fs = require('fs')

fs.readFile('tp.txt', (err, inputD) => {

    if (err) throw err;
```

```
        console.log(inputD.toString());

    })
```

Output:

Following is the output of the above example −

You are reading the content from Tutorials Point

- **What are all the looping structures in JavaScript?**

Answer: JavaScript supports different kinds of loops:

- for - loops through a block of code a number of times.
- for/in - loops through the properties of an object.
- for/of - loops through the values of an iterable object.
- while - loops through a block of code while a specified condition is true.
- do/while - also loops through a block of code while a specified condition is true.

- **How can you convert the string of any base to an integer in JavaScript?**

Answer: To convert a string to an integer parseInt(), Number(), and Unary operator(+) function is used in javascript. parseInt()

function returns Nan(not a number) when the string doesn’t contain a number. If a string with a number is sent, then only that number will be returned as the output. This function won't accept spaces. If any particular number with spaces is sent, then the part of the number that presents before space will be returned as the output. To convert a string into integer we can use parseInt(), Number() and Unary operator(+).

We can convert a string to javascript by the following methods:

- Using the parseInt() method
- Using the Number() method
- Using the Unary operator

Using the parseInt() method: JavaScript parseInt() Method is used to accept the string and radix parameter and convert it into an integer.

Syntax:

parseInt(Value, radix)

For example:

```
        function convertStoI() {
```

```
        let a = "100";

        let b = parseInt(a);

        console.log("Integer value is" + b);

        let d = parseInt("3 11 43");

        console.log('Integer value is ' + d);

    }

    convertStoI();
```

Output:

Integer value is 100

Integer value is 3

- **What is the function of the delete operator?**

Answer: The delete operator removes a property from an object. If the property's value is an object and there are no more references to the object, the object held by that property is eventually released automatically.

For example:

```
let emp = {

        firstName: "Surabhi",
    lastName: "Prajapati",
    salary: 400000
}
console.log(delete emp.salary);
console.log(emp);
```

Output:

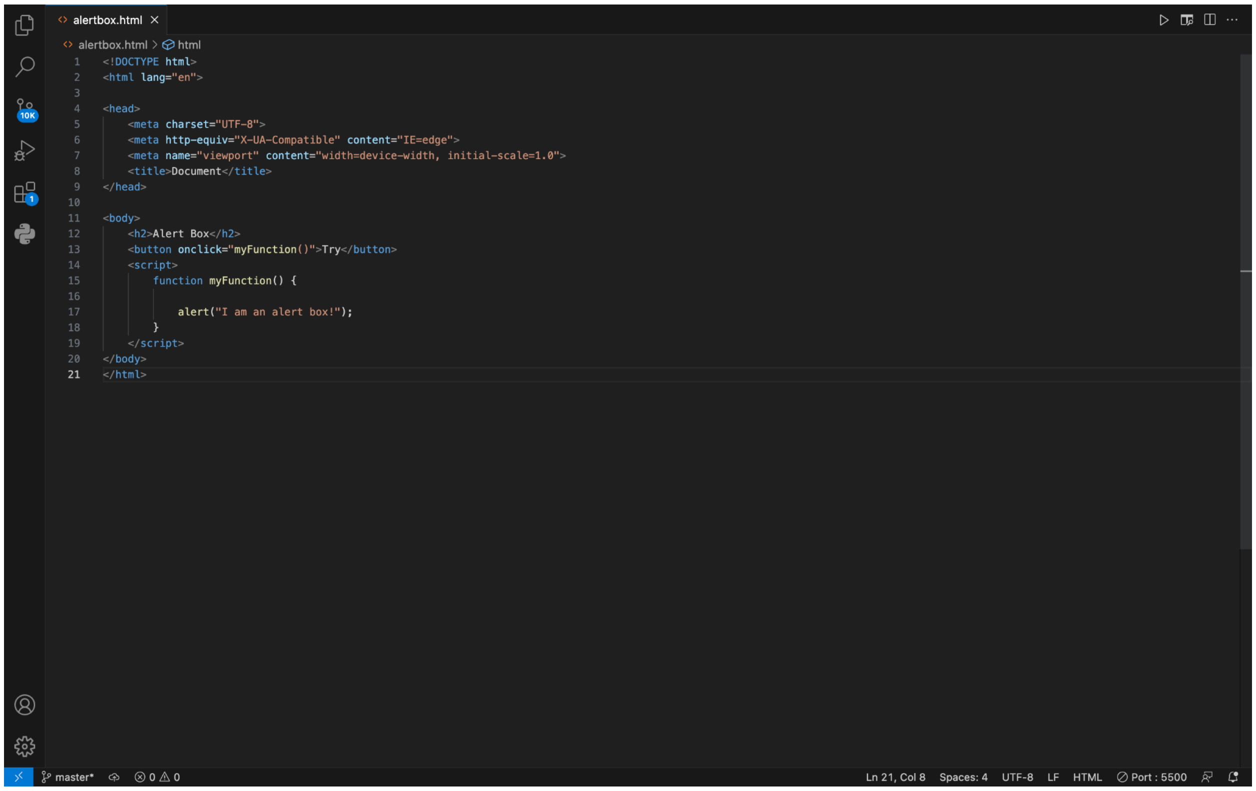
```
true

{"firstName":"Surabhi","lastName":"Prajapati"}
```

- **What are all the types of Pop up boxes available in JavaScript?**

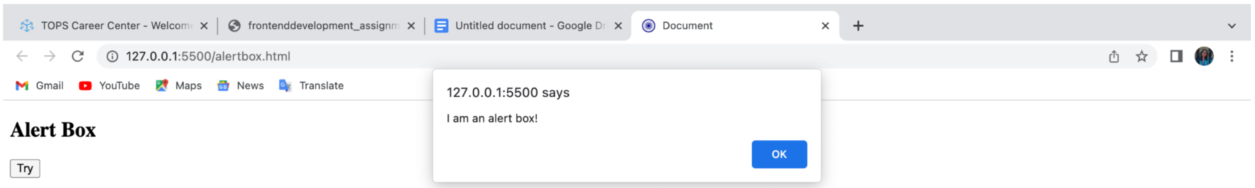
Answer: JavaScript has three kinds of popup boxes: Alert box, Confirm box, and Prompt box.

Alert Box: An alert box is often used if you want to make sure information comes through to the user.When an alert box pops up, the user will have to click "OK" to proceed.

For example:



```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>

<body>
    <h2>Alert Box</h2>
    <button onclick="myFunction()">Try</button>
    <script>
        function myFunction() {

            alert("I am an alert box!");
        }
    </script>
</body>
</html>
```



127.0.0.1:5500/alertbox.html

127.0.0.1:5500 says

I am an alert box!

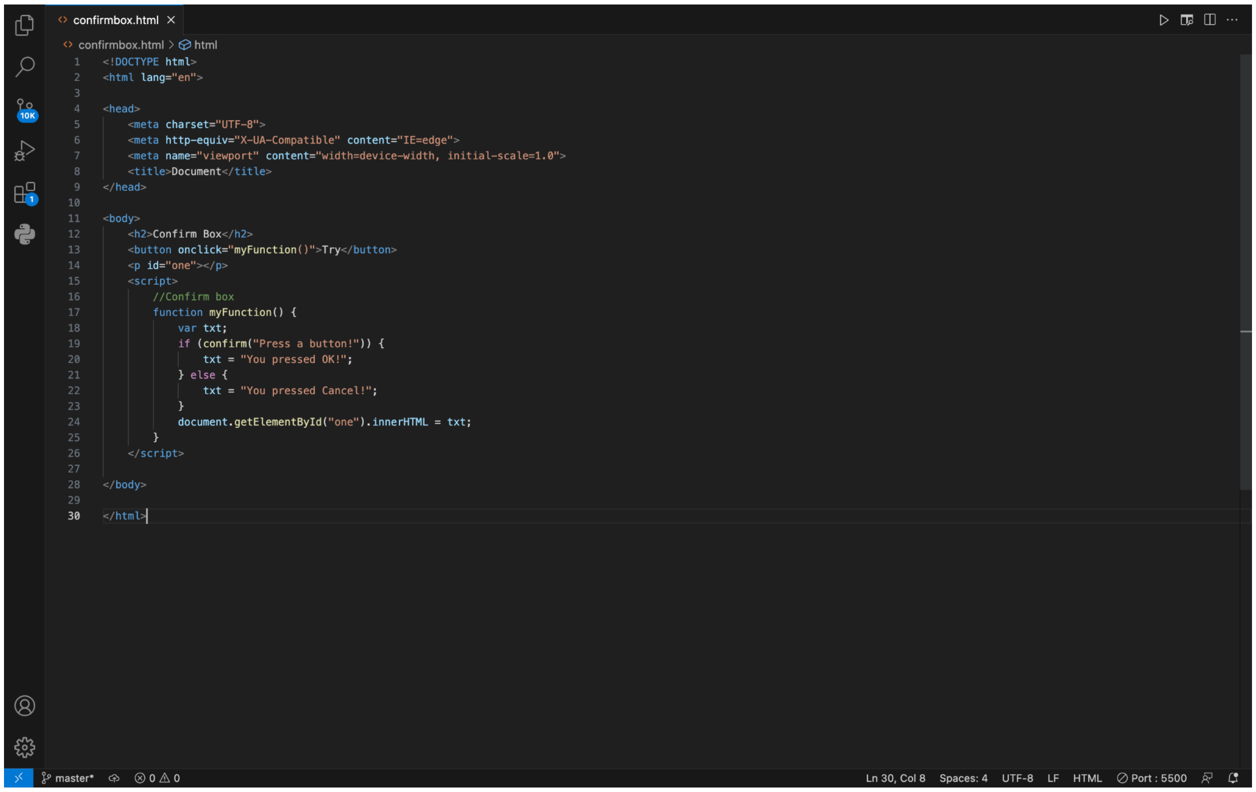
OK

Alert Box

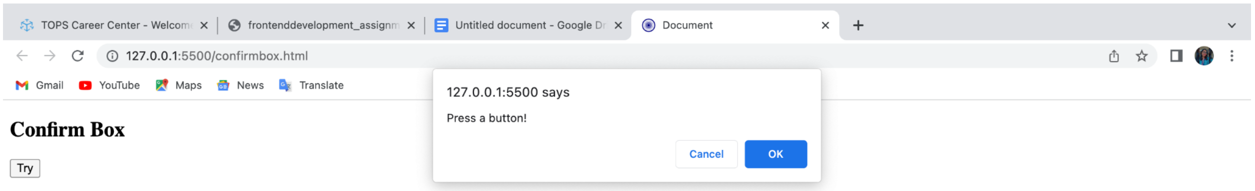
Try

Confirm Box: A confirm box is often used if you want the user to verify or accept something. When a confirm box pops up, the user will have to click either "OK" or "Cancel" to proceed. If the user clicks "OK", the box returns true. If the user clicks "Cancel", the box returns false.

For example:



```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>

<body>
    <h2>Confirm Box</h2>
    <button onclick="myFunction()">Try</button>
    <p id="one"></p>
    <script>
        //Confirm box
        function myFunction() {
            var txt;
            if (confirm("Press a button!")) {
                txt = "You pressed OK!";
            } else {
                txt = "You pressed Cancel!";
            }
            document.getElementById("one").innerHTML = txt;
        }
    </script>

</body>

</html>
```


TOPS Career Center - Welcome
frontenddevelopment_assignm
Untitled document - Google Dr
Document
127.0.0.1:5500/confirmbox.html
Gmail
YouTube
Maps
News
Translate

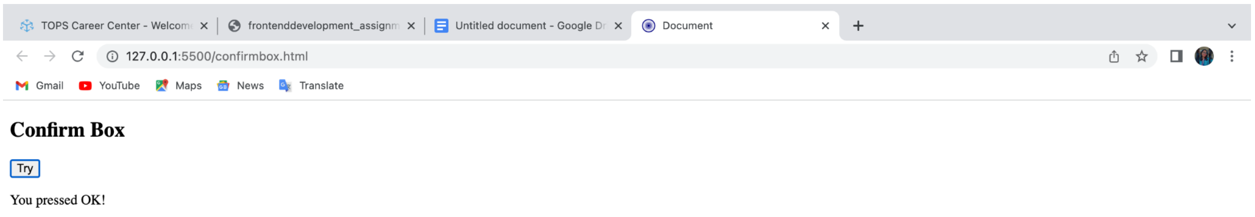
Confirm Box

Try

127.0.0.1:5500 says

Press a button!

Cancel OK


TOPS Career Center - Welcome
frontenddevelopment_assignm
Untitled document - Google Dr
Document
127.0.0.1:5500/confirmbox.html
Gmail
YouTube
Maps
News
Translate

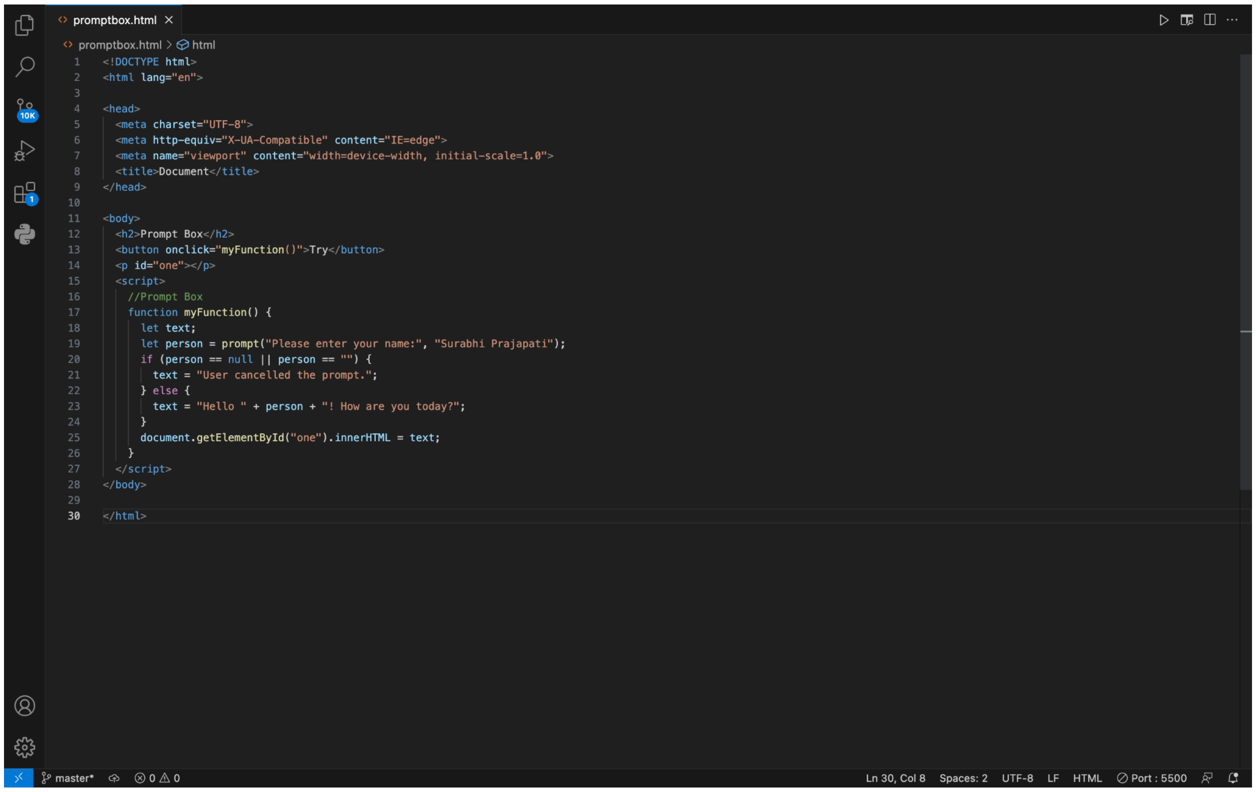
Confirm Box

Try

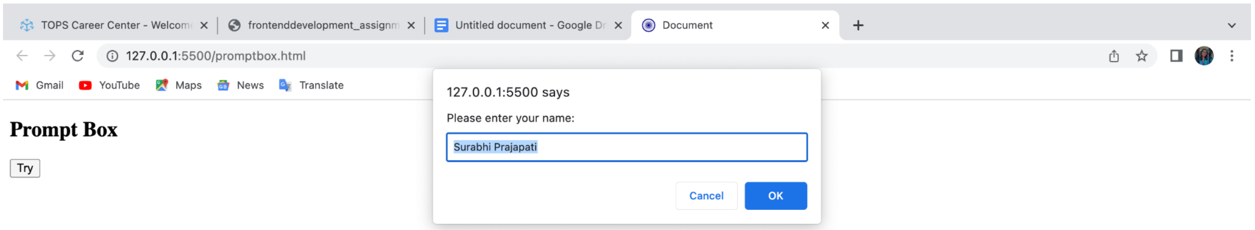
You pressed OK!

Prompt Box: A prompt box is often used if you want the user to input a value before entering a page. When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value. If the user clicks "OK" the box returns the input value. If the user clicks "Cancel" the box returns null.

For example:



```html
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>

<body>
  <h2>Prompt Box</h2>
  <button onclick="myFunction()">Try</button>
  <p id="one"></p>
  <script>
    //Prompt Box
    function myFunction() {
      let text;
      let person = prompt("Please enter your name:", "Surabhi Prajapati");
      if (person == null || person == "") {
        text = "User cancelled the prompt.";
      } else {
        text = "Hello " + person + "! How are you today?";
      }
      document.getElementById("one").innerHTML = text;
    }
  </script>
</body>

</html>
```


TOPS Career Center - Welcom
frontenddevelopment_assignm
Untitled document - Google D
Document
127.0.0.1:5500/promptbox.html
Gmail
YouTube
Maps
News
Translate

Prompt Box

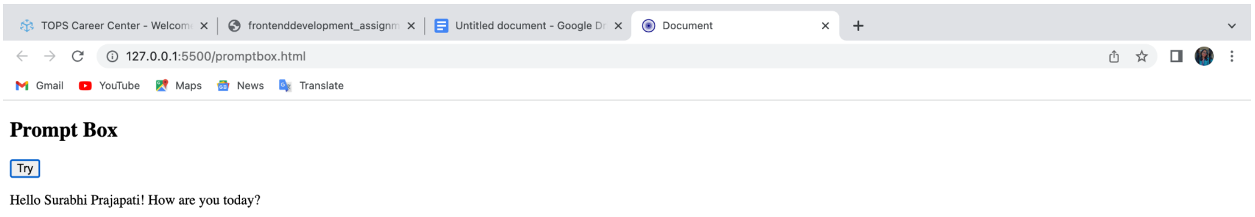
Try

127.0.0.1:5500 says

Please enter your name:

Surabhi Prajapati

Cancel OK


TOPS Career Center - Welcom
frontenddevelopment_assignm
Untitled document - Google D
Document
127.0.0.1:5500/promptbox.html
Gmail
YouTube
Maps
News
Translate

Prompt Box

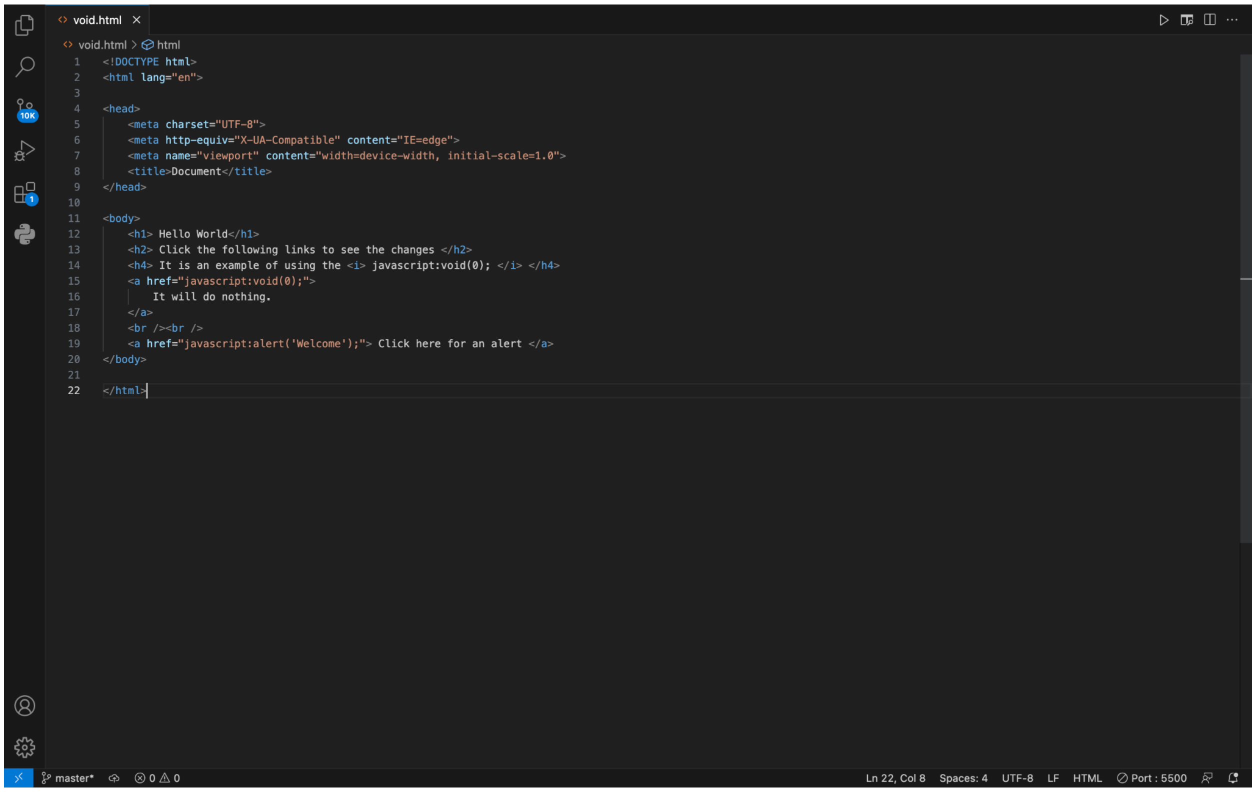
Try

Hello Surabhi Prajapati! How are you today?

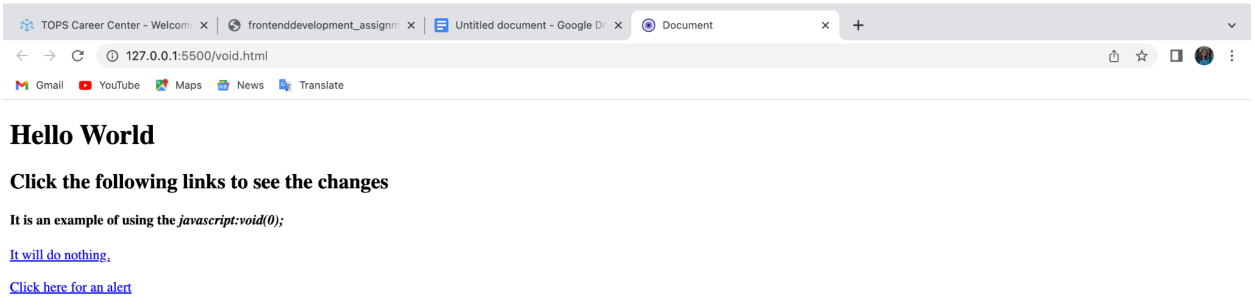
- **What is the use of Void (0)?**

Answer: The void operator is used to evaluate an expression and returns the undefined. Generally, this operator is used for obtaining the undefined primitive value. It is often used with hyperlinks. Usually the browser refreshes the page or loads a new page on clicking a link. The javascript:void(0) can be used when we don't want to refresh or load a new page in the browser on clicking a hyperlink.

For example:



```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>

<body>
    <h1> Hello World</h1>
    <h2> Click the following links to see the changes </h2>
    <h4> It is an example of using the <i> javascript:void(0); </i> </h4>
    <a href="javascript:void(0);">
        It will do nothing.
    </a>
    <br /><br />
    <a href="javascript:alert('Welcome');"> Click here for an alert </a>
</body>

</html>
```


TOPS Career Center - Welcom
frontenddevelopment_assignm
Untitled document - Google Dr
Document
127.0.0.1:5500/void.html
Gmail
YouTube
Maps
News
Translate

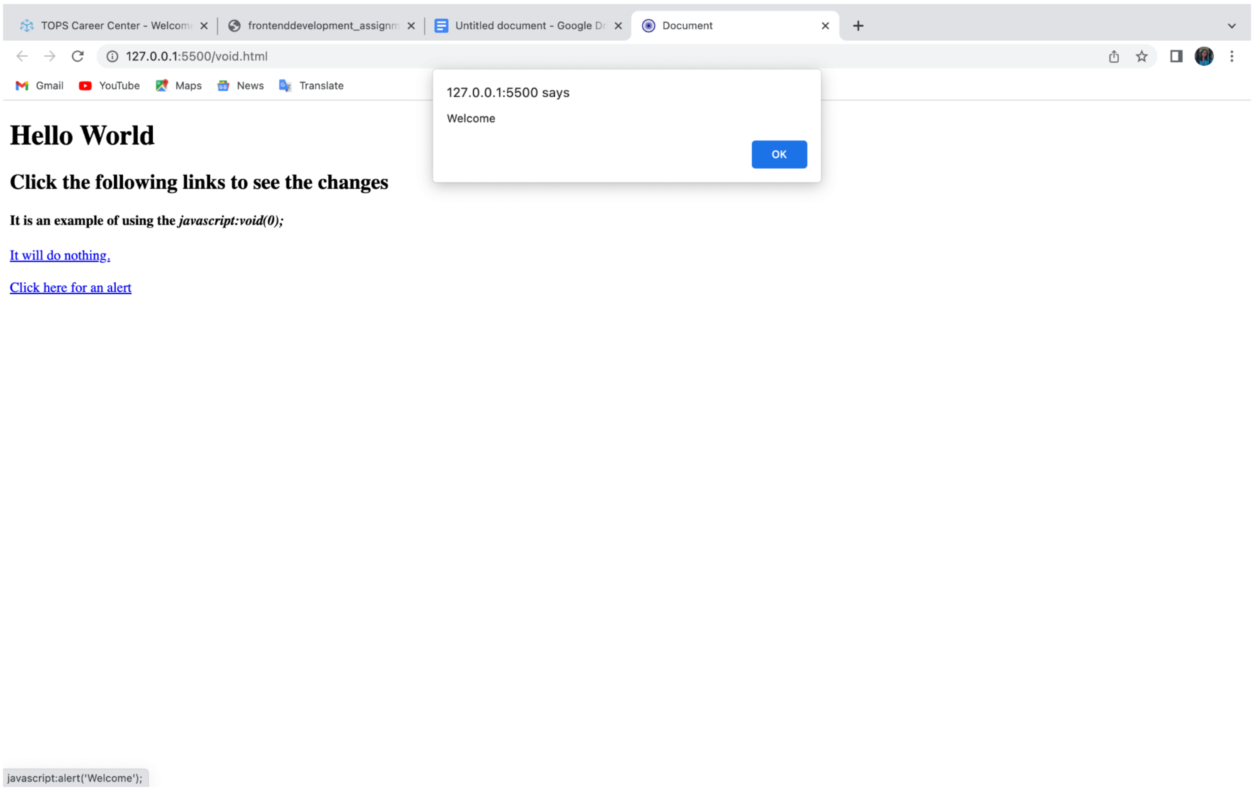
Hello World

Click the following links to see the changes

It is an example of using the *javascript:void(0);*

It will do nothing.

Click here for an alert


TOPS Career Center - Welcom
frontenddevelopment_assignm
Untitled document - Google Dr
Document
127.0.0.1:5500/void.html
Gmail
YouTube
Maps
News
Translate
OK

127.0.0.1:5500 says

Welcome

Hello World

Click the following links to see the changes

It is an example of using the *javascript:void(0);*

It will do nothing.

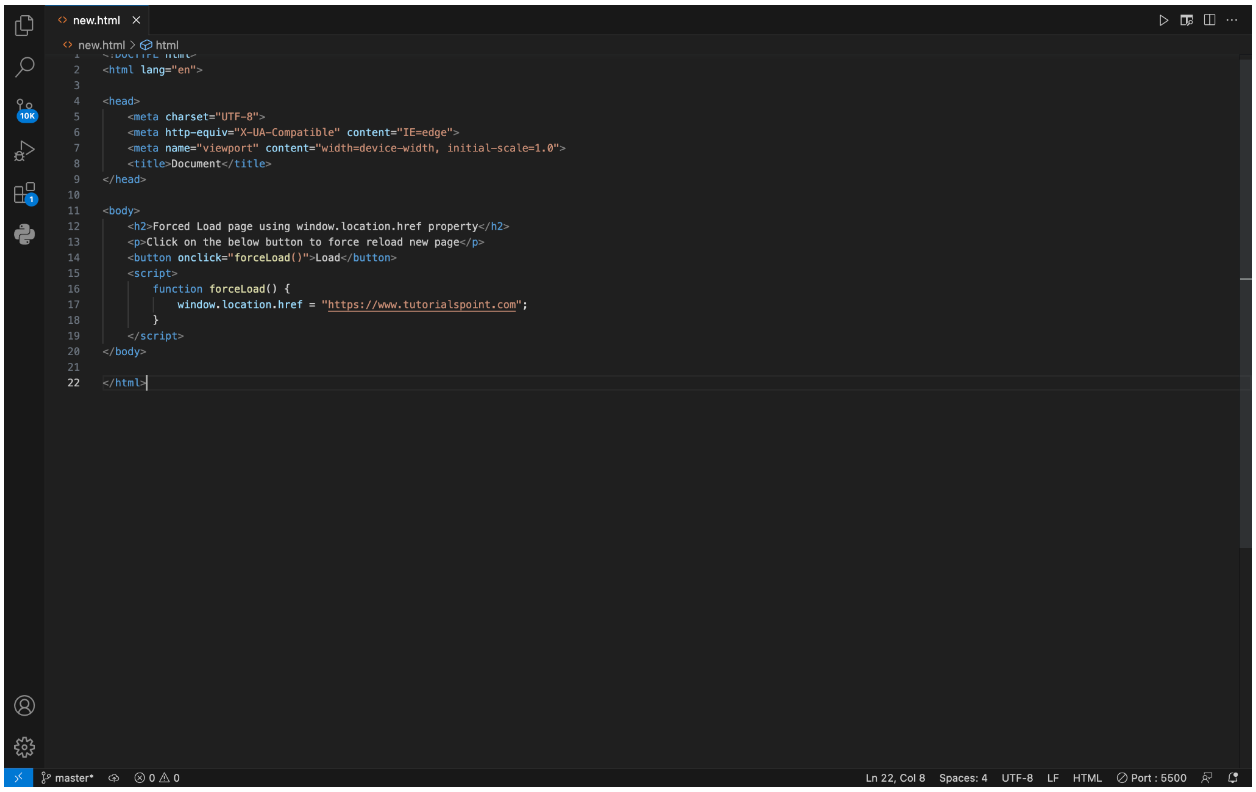
Click here for an alert

javascript:alert('Welcome');

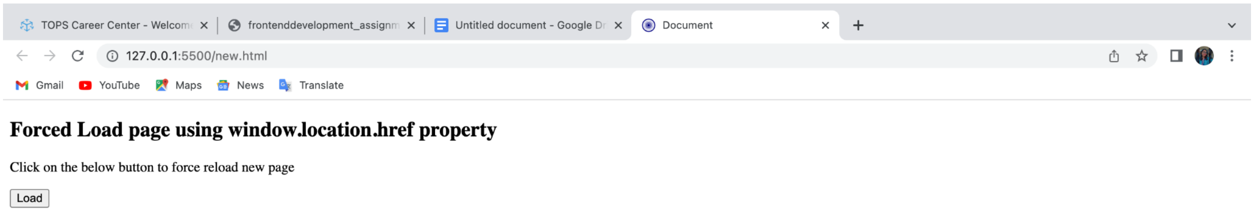
- **How can a page be forced to load another page in JavaScript?**

Answer: In JavaScript, we can use window.location object to force a page to load another page. We can use the location object to set the URL of a new page. There are different ways – window.location.href property, window.location.assign() and window.location.replace() methods, to set the URL of a new page using the location object.

For example:



```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>

<body>
    <h2>Forced Load page using window.location.href property</h2>
    <p>Click on the below button to force reload new page</p>
    <button onclick="forceLoad()">Load</button>
    <script>
        function forceLoad() {
            window.location.href = "https://www.tutorialspoint.com";
        }
    </script>
</body>

</html>
```


TOPS Career Center - Welcom
frontenddevelopment_assignm
Untitled document - Google Dr
Document
127.0.0.1:5500/new.html
Gmail
YouTube
Maps
News
Translate

Forced Load page using window.location.href property

Click on the below button to force reload new page

Load

- **What are the disadvantages of using innerHTML in JavaScript?**

Answer: Disadvantages of using innerHTML property in JavaScript:

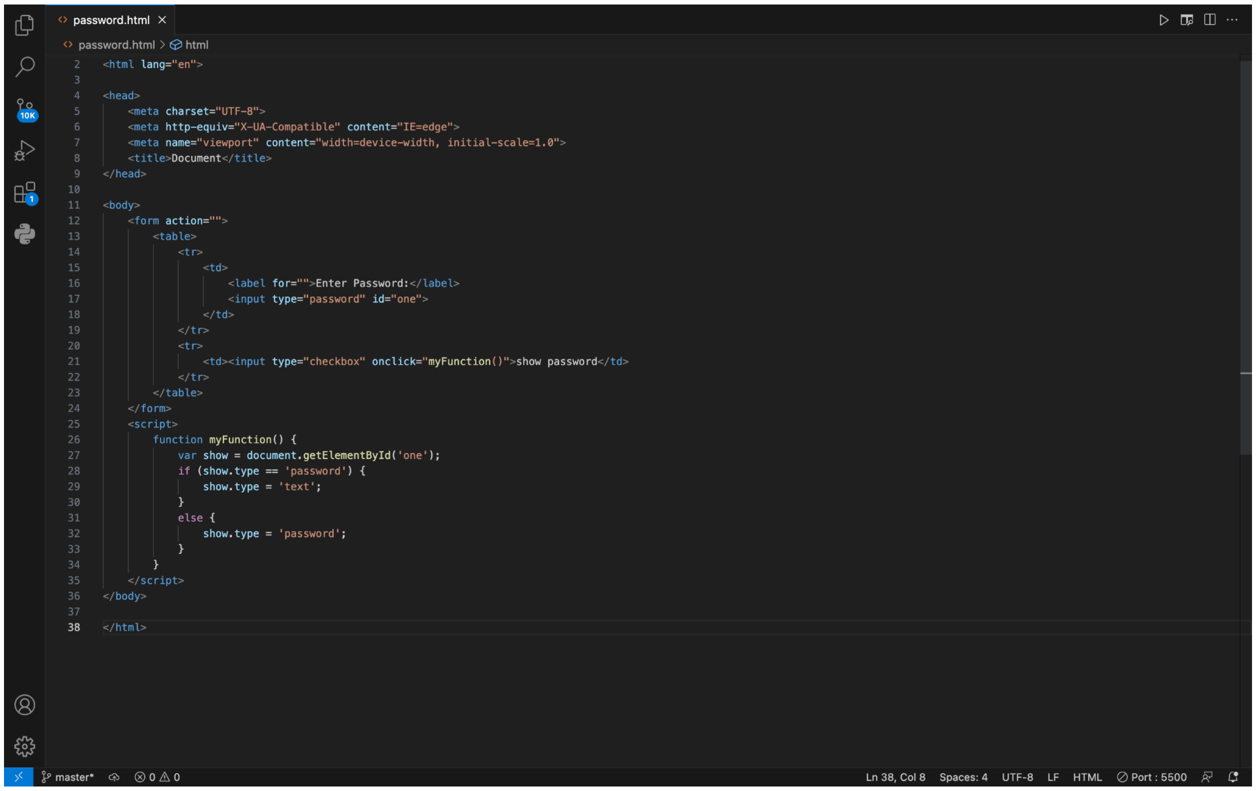
- The use of innerHTML is very slow: The process of using innerHTML is much slower as its contents are slowly built, also already parsed contents and elements are also re-parsed which takes time.
- Preserves event handlers attached to any DOM elements: The event handlers do not get attached to the new elements created by setting innerHTML automatically. To

do so one has to keep track of the event handlers and attach it to new elements manually. This may cause a memory leak on some browsers.

- Content is replaced everywhere: Either you add, append, delete or modify contents on a webpage using innerHTML, all contents is replaced, also all the DOM nodes inside that element are reparsed and recreated.
- Appending to innerHTML is not supported: Usually, += is used for appending in JavaScript. But on appending to an Html tag using innerHTML, the whole tag is re-parsed.

- **Create password field with show hide functionalities.**

Answer:

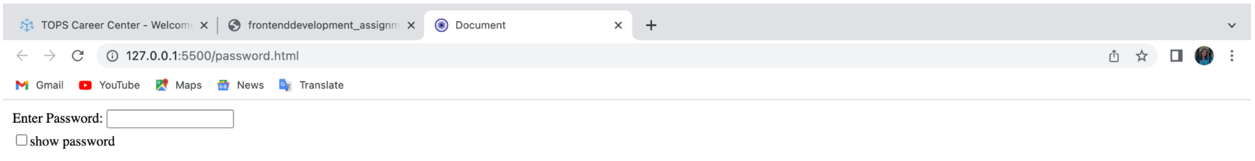


password.html

password.html > html

```
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>

<body>
    <form action="">
        <table>
            <tr>
                <td>
                    <label for="">Enter Password:</label>
                    <input type="password" id="one">
                </td>
            </tr>
            <tr>
                <td><input type="checkbox" onclick="myFunction()">show password</td>
            </tr>
        </table>
    </form>
    <script>
        function myFunction() {
            var show = document.getElementById('one');
            if (show.type == 'password') {
                show.type = 'text';
            }
            else {
                show.type = 'password';
            }
        }
    </script>
</body>

</html>
```

master* ⊗ 0 ⚠ 0 Ln 38, Col 8 Spaces: 4 UTF-8 LF HTML Port : 5500



TOPS Career Center - Welcome | frontenddevelopment_assignm | Document

127.0.0.1:5500/password.html

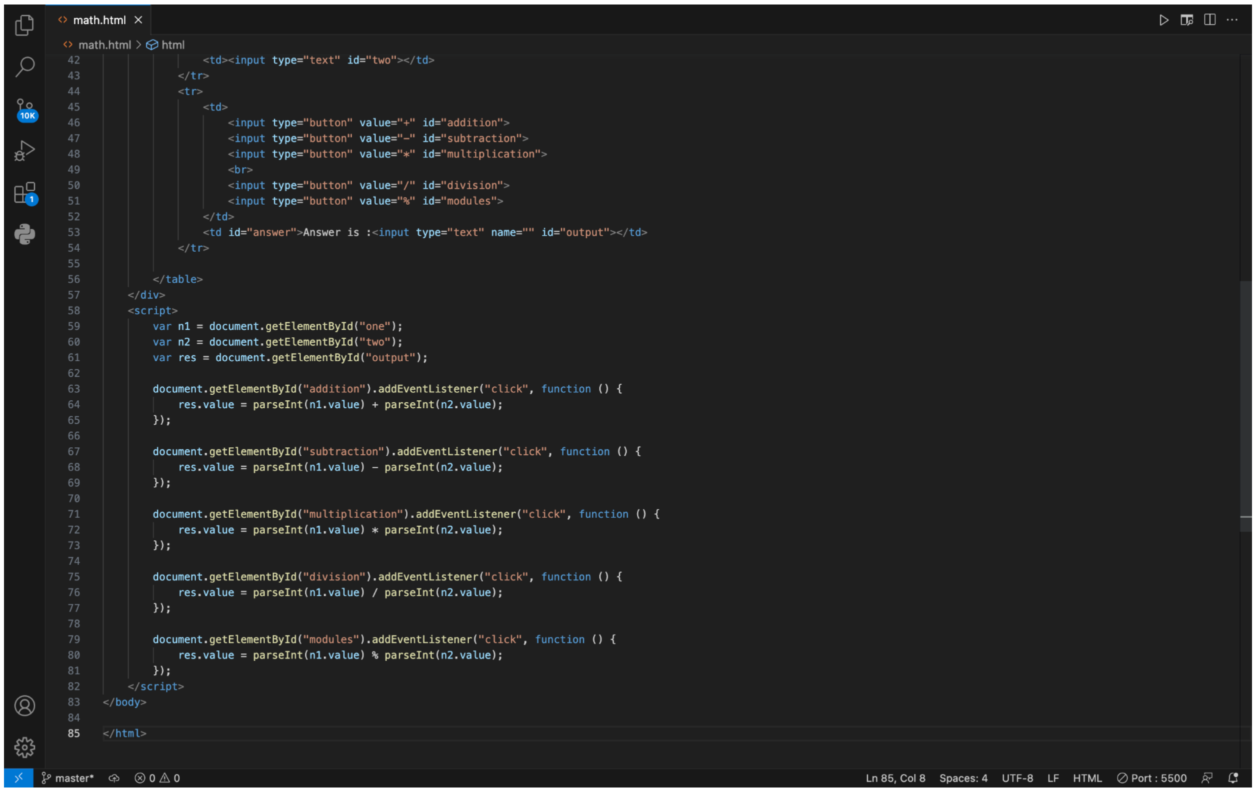
Gmail YouTube Maps News Translate

Enter Password:

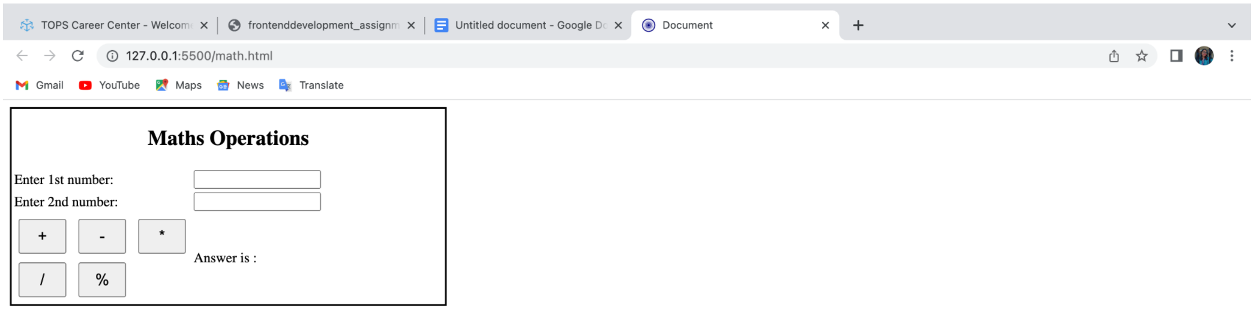
☐ show password

- **Create basic math operations in JS.**

Answer:



```
                <td><input type="text" id="two"></td>
            </tr>
            <tr>
                <td>
                    <input type="button" value="+" id="addition">
                    <input type="button" value="-" id="subtraction">
                    <input type="button" value="*" id="multiplication">
                    <br>
                    <input type="button" value="/" id="division">
                    <input type="button" value="%" id="modules">
                </td>
                <td id="answer">Answer is :<input type="text" name="" id="output"></td>
            </tr>

        </table>
    </div>
    <script>
        var n1 = document.getElementById("one");
        var n2 = document.getElementById("two");
        var res = document.getElementById("output");

        document.getElementById("addition").addEventListener("click", function () {
            res.value = parseInt(n1.value) + parseInt(n2.value);
        });

        document.getElementById("subtraction").addEventListener("click", function () {
            res.value = parseInt(n1.value) - parseInt(n2.value);
        });

        document.getElementById("multiplication").addEventListener("click", function () {
            res.value = parseInt(n1.value) * parseInt(n2.value);
        });

        document.getElementById("division").addEventListener("click", function () {
            res.value = parseInt(n1.value) / parseInt(n2.value);
        });

        document.getElementById("modules").addEventListener("click", function () {
            res.value = parseInt(n1.value) % parseInt(n2.value);
        });
    </script>
</body>

</html>
```


TOPS Career Center - Welcom
frontenddevelopment_assignm
Untitled document - Google D
Document
127.0.0.1:5500/math.html
Gmail YouTube Maps News Translate

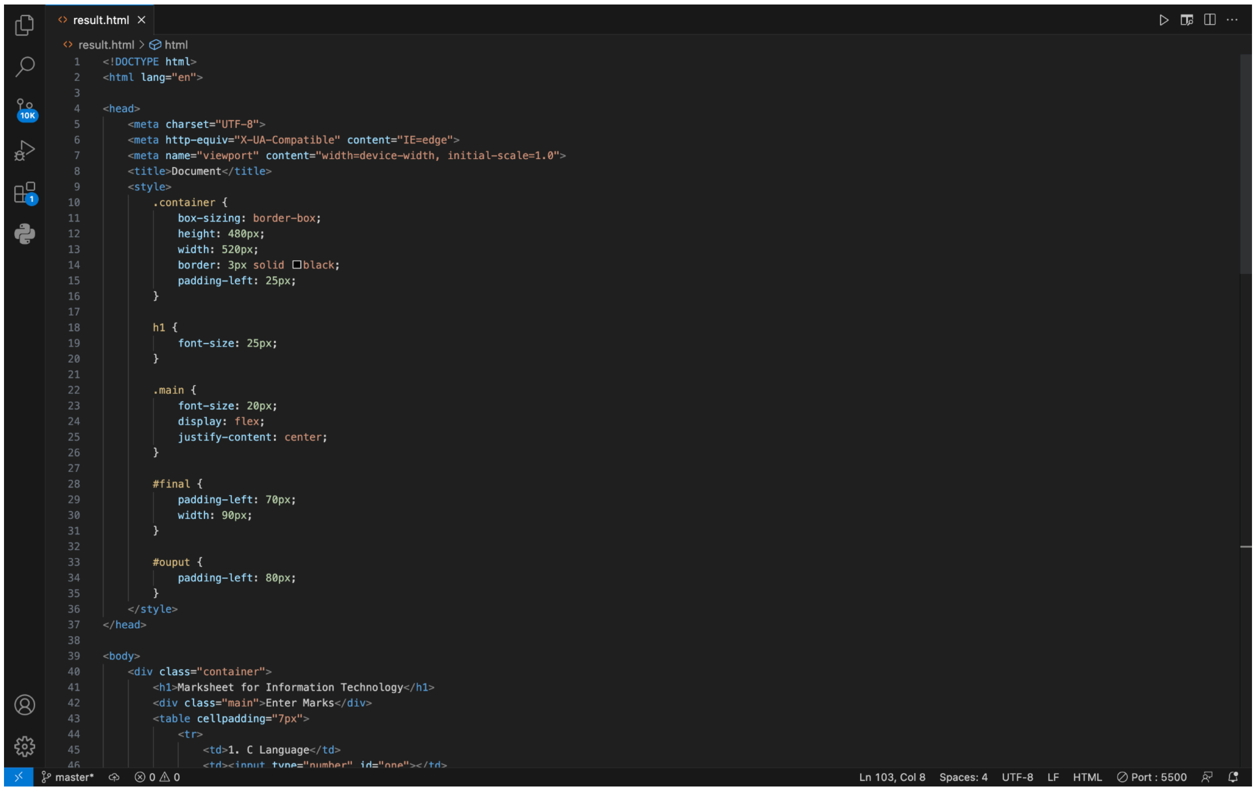
Maths Operations

Enter 1st number:
Enter 2nd number:

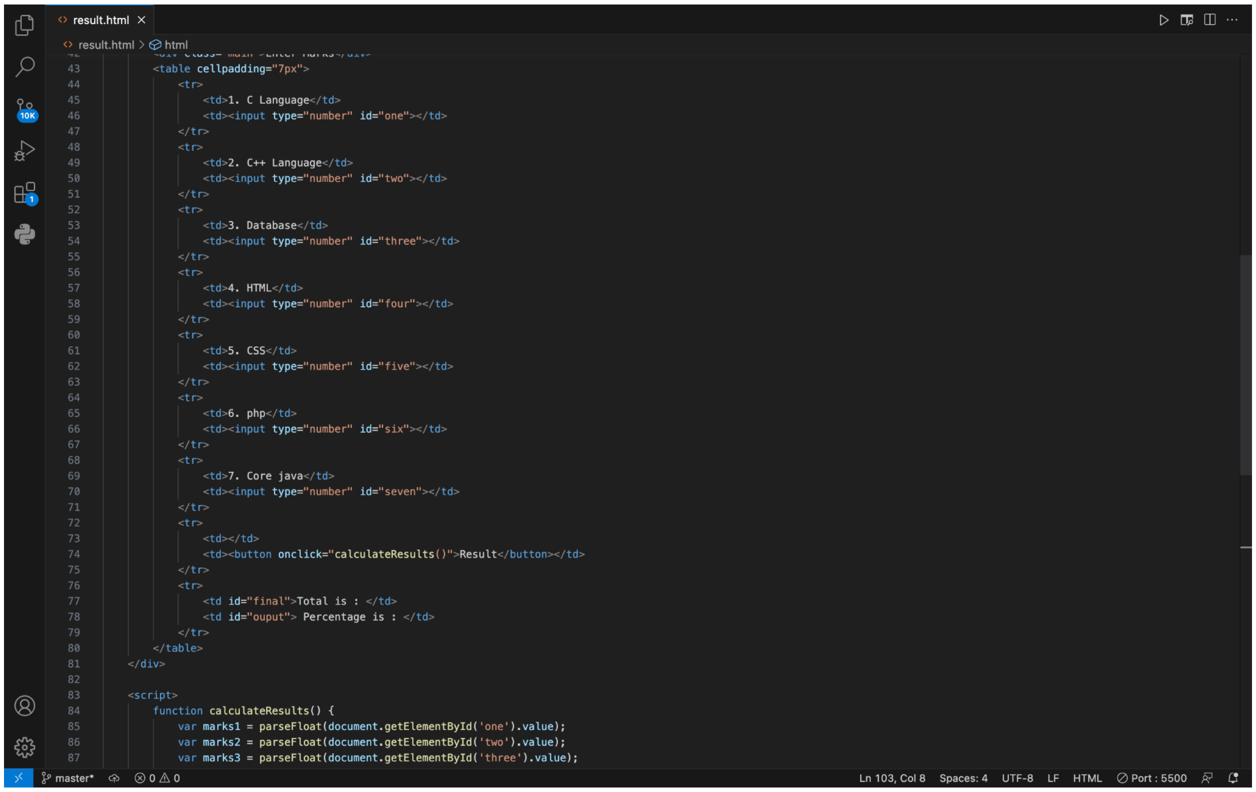
Answer is :

- **Create results.**

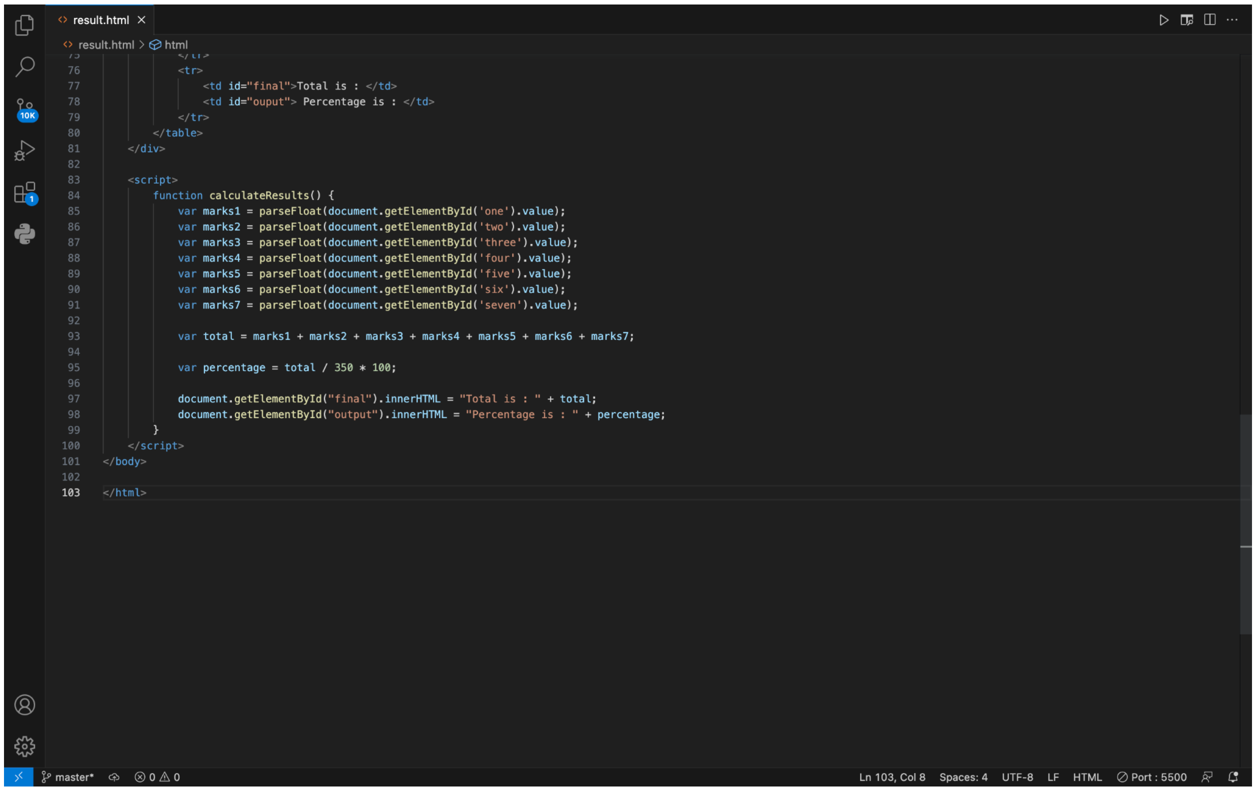
Answer:



```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>
        .container {
            box-sizing: border-box;
            height: 480px;
            width: 520px;
            border: 3px solid black;
            padding-left: 25px;
        }

        h1 {
            font-size: 25px;
        }

        .main {
            font-size: 20px;
            display: flex;
            justify-content: center;
        }

        #final {
            padding-left: 70px;
            width: 90px;
        }

        #ouput {
            padding-left: 80px;
        }
    </style>
</head>

<body>
    <div class="container">
        <h1>Marksheet for Information Technology</h1>
        <div class="main">Enter Marks</div>
        <table cellpadding="7px">
            <tr>
                <td>1. C Language</td>
                <td><input type="number" id="one"></td>
```

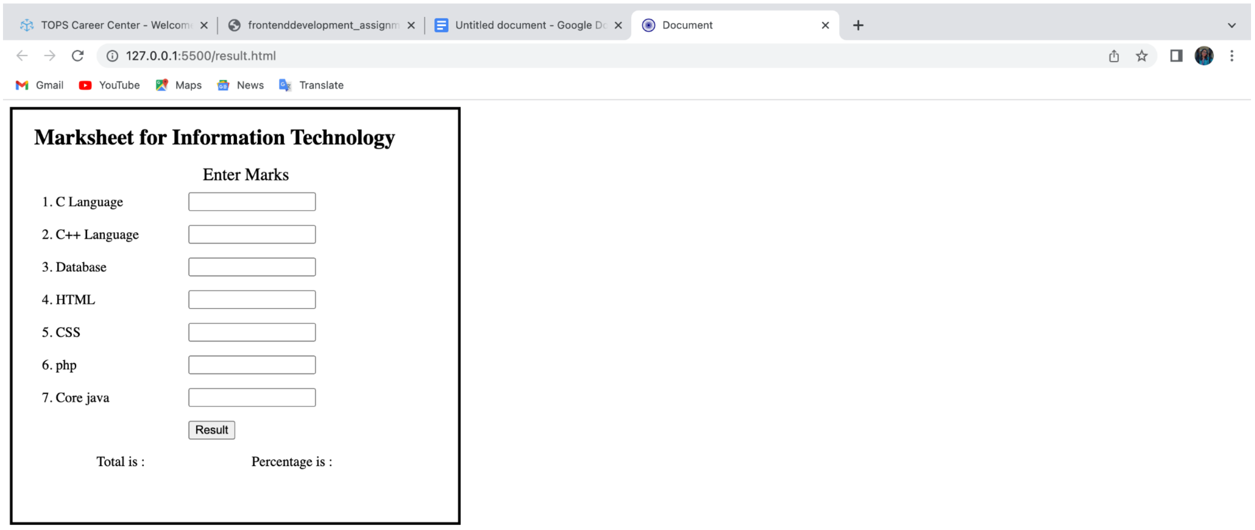


```html
        <table cellpadding="7px">
            <tr>
                <td>1. C Language</td>
                <td><input type="number" id="one"></td>
            </tr>
            <tr>
                <td>2. C++ Language</td>
                <td><input type="number" id="two"></td>
            </tr>
            <tr>
                <td>3. Database</td>
                <td><input type="number" id="three"></td>
            </tr>
            <tr>
                <td>4. HTML</td>
                <td><input type="number" id="four"></td>
            </tr>
            <tr>
                <td>5. CSS</td>
                <td><input type="number" id="five"></td>
            </tr>
            <tr>
                <td>6. php</td>
                <td><input type="number" id="six"></td>
            </tr>
            <tr>
                <td>7. Core java</td>
                <td><input type="number" id="seven"></td>
            </tr>
            <tr>
                <td></td>
                <td><button onclick="calculateResults()">Result</button></td>
            </tr>
            <tr>
                <td id="final">Total is : </td>
                <td id="ouput"> Percentage is : </td>
            </tr>
        </table>
    </div>

    <script>
        function calculateResults() {
            var marks1 = parseFloat(document.getElementById('one').value);
            var marks2 = parseFloat(document.getElementById('two').value);
            var marks3 = parseFloat(document.getElementById('three').value);
```



result.html

```html
            <tr>
                <td id="final">Total is : </td>
                <td id="ouput"> Percentage is : </td>
            </tr>
        </table>
    </div>

    <script>
        function calculateResults() {
            var marks1 = parseFloat(document.getElementById('one').value);
            var marks2 = parseFloat(document.getElementById('two').value);
            var marks3 = parseFloat(document.getElementById('three').value);
            var marks4 = parseFloat(document.getElementById('four').value);
            var marks5 = parseFloat(document.getElementById('five').value);
            var marks6 = parseFloat(document.getElementById('six').value);
            var marks7 = parseFloat(document.getElementById('seven').value);

            var total = marks1 + marks2 + marks3 + marks4 + marks5 + marks6 + marks7;

            var percentage = total / 350 * 100;

            document.getElementById("final").innerHTML = "Total is : " + total;
            document.getElementById("output").innerHTML = "Percentage is : " + percentage;
        }
    </script>
</body>

</html>
```



127.0.0.1:5500/result.html

Marksheet for Information Technology

Enter Marks

1. C Language
2. C++ Language
3. Database
4. HTML
5. CSS
6. php
7. Core java

Result

Total is : Percentage is :