웹 프로그래밍

Web Programming

1420-7001

By

Dr. Abdul Majeed (조교수)

1st Semester, 2025

가천대학교
Gachon University

IT융합대학 컴퓨터공학부(컴퓨터공학전공)

Summary of the Previous Lesson

Part I

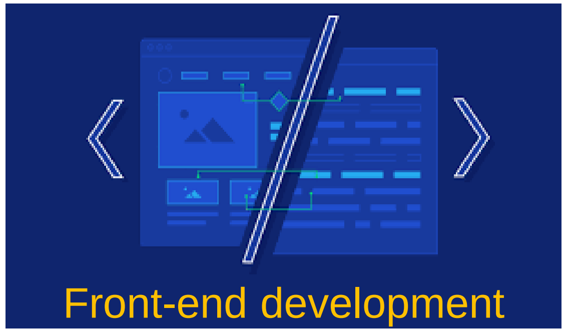
HTML Attributes <Classes, id, etc.>

- ❑ Style
- ❑ Class
- ❑ Id
- ❑ Examples of all concepts
- ❑ Differences b/w them

Part II

Working with Links

- ❑ Text links
- ❑ Image links
- ❑ Download link
- ❑ Email links
- ❑ Etc.



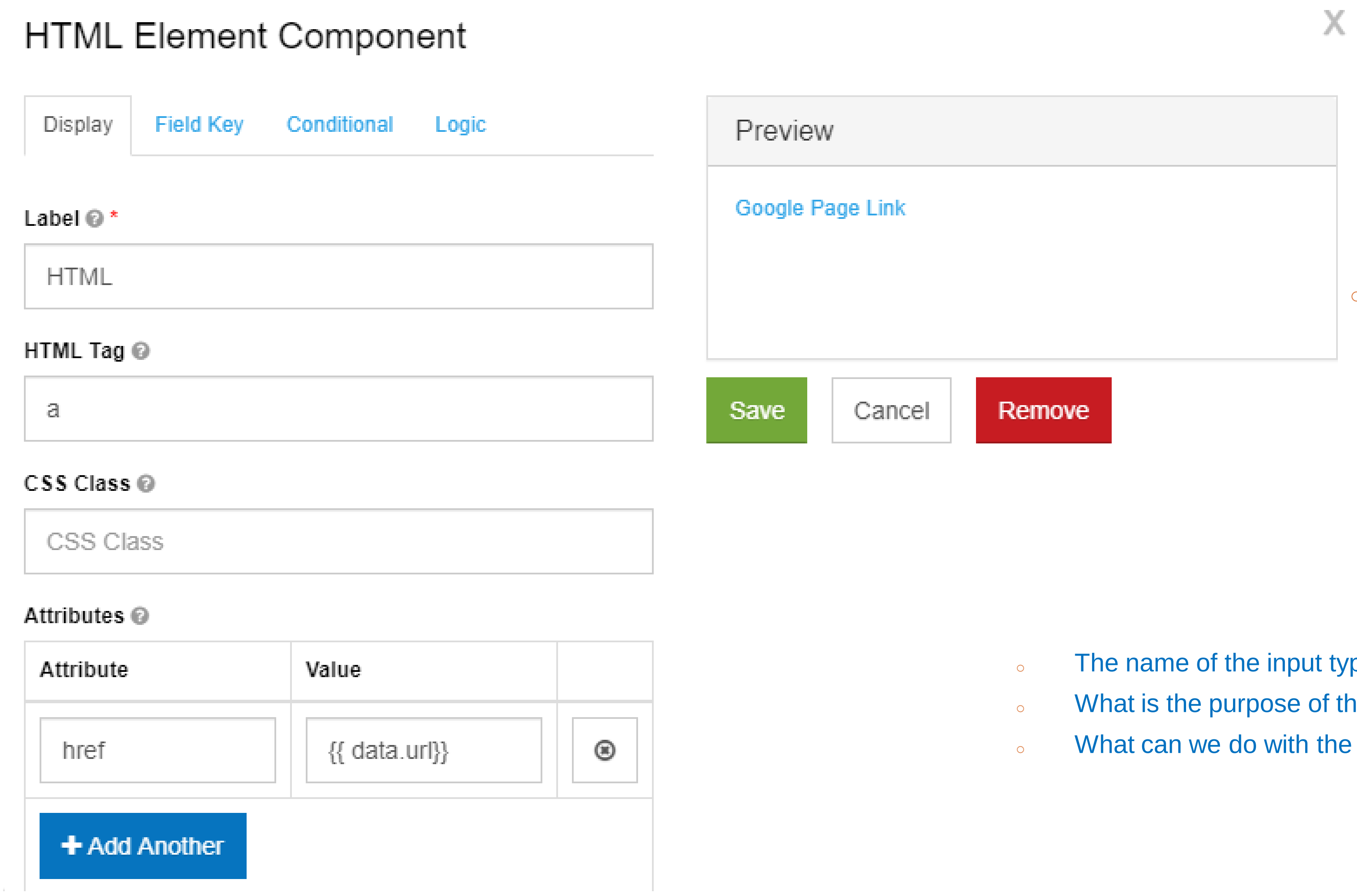
Part III

Working with forms

- ❑ Simple forms
- ❑ Elements of forms
- ❑ Input element (14 different types)
- ❑ Etc.

Note: Please execute all codes at least once on your computers.

Class Activity #: 06


HTML Element Component

Display | Field Key | Conditional | Logic

Label ? *

HTML

HTML Tag ?

a

CSS Class ?

CSS Class

Attributes ?

Attribute	Value	
href	{{ data.url}}	⊗

- What is the purpose of <u>? in the form label?</u>

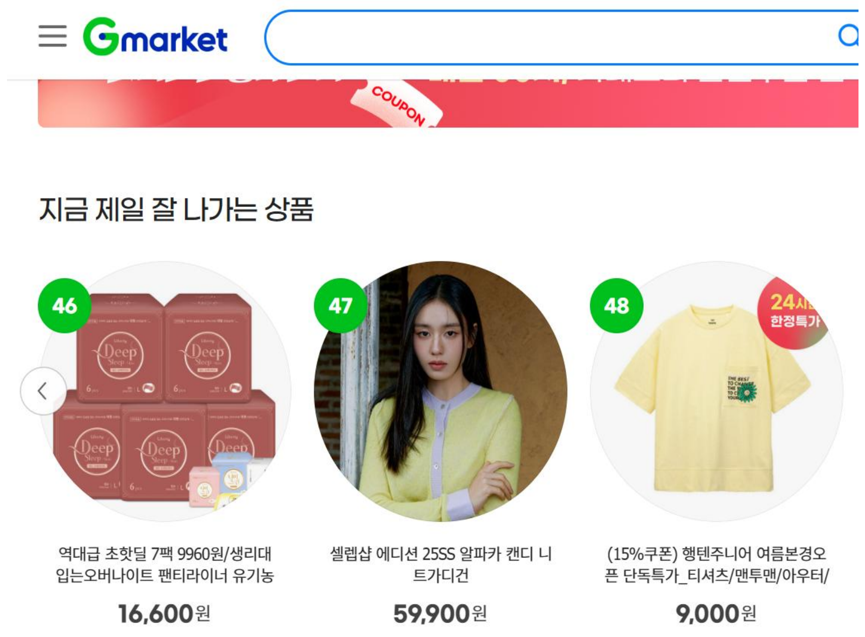
- The name of the input type other than text that is used in this figure.
- What is the purpose of the remove and cancel button in this figure?
- What can we do with the data entered by the user in this form?

Difference between ID and Class

- A class name can be used by multiple HTML elements, while an ID name must only be used by one HTML element within the page.
- Why?
  - Some parts are fixed → ID
  - Non-fixed parts → Class

Target attribute of an Anchor tag

- On some websites, the output is displayed on the same page, while some websites' response is displayed on other pages.



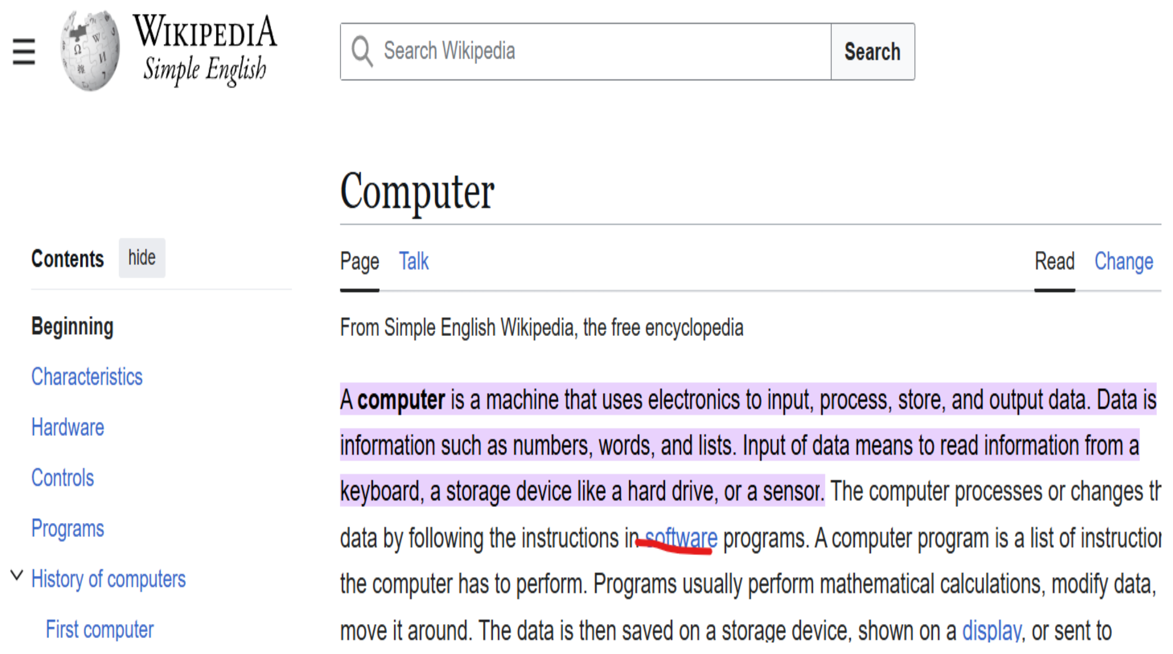
지금 제일 잘 나가는 상품

46 역대급 초핫딜 7팩 9960원/생리대 입는오버나이트 팬티라이너 유기농 16,600원

47 셀렙샵 에디션 25SS 알파카 캔디 니트가디건 59,900원

48 (15%쿠폰) 행텐주니어 여름본경오픈 단독특가_티셔츠/맨투맨/아우터/ 9,000원

<a. G-market>



Computer

From Simple English Wikipedia, the free encyclopedia

A **computer** is a machine that uses electronics to input, process, store, and output data. Data is information such as numbers, words, and lists. Input of data means to read information from a keyboard, a storage device like a hard drive, or a sensor. The computer processes or changes th
data by following the instructions in software programs. A computer program is a list of instructio
the computer has to perform. Programs usually perform mathematical calculations, modify data, move it around. The data is then saved on a storage device, shown on a display, or sent to

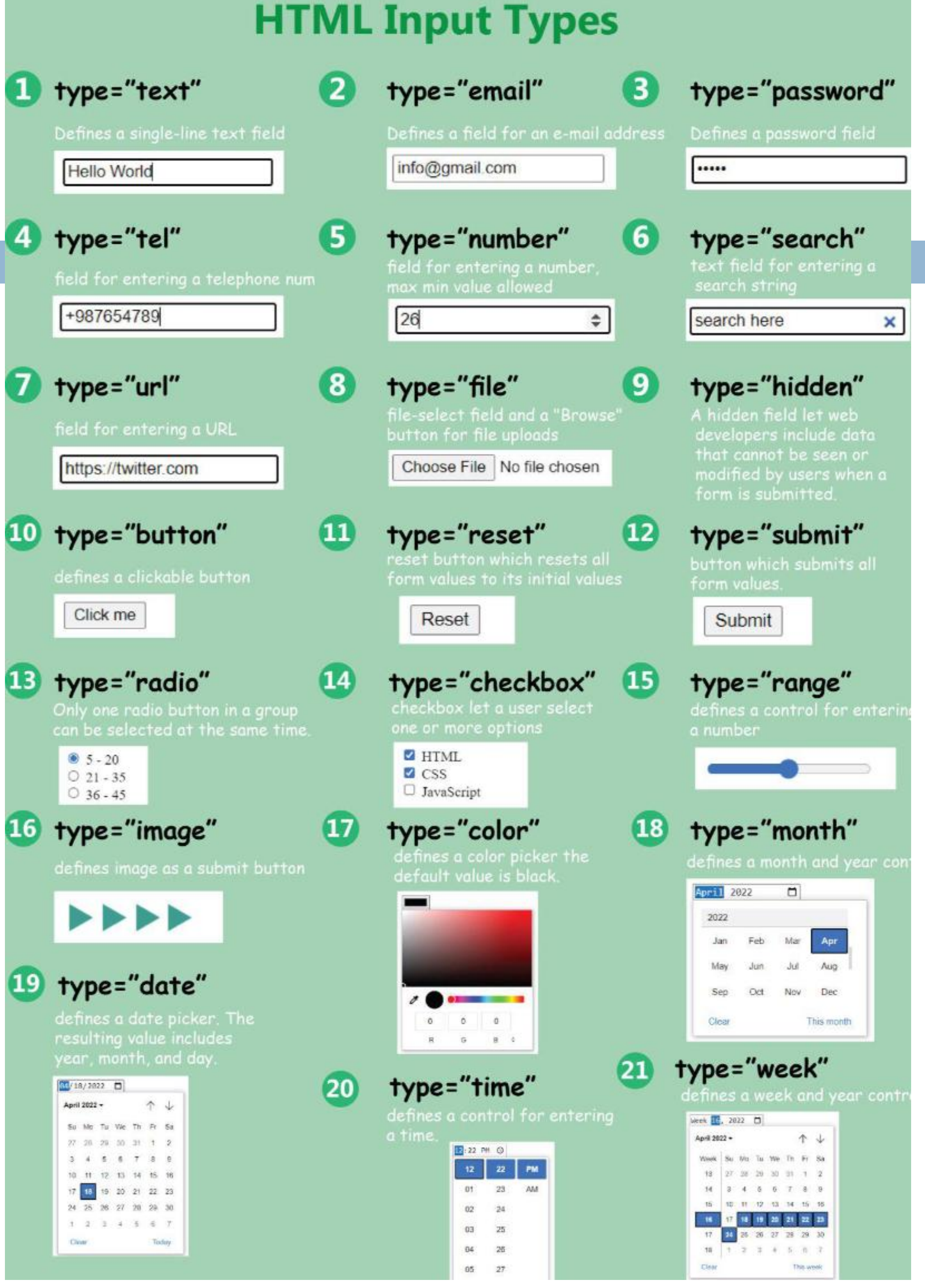
<b. wikipedia>

HTML Forms

A Tool/Way of Data Collection in Web Apps

HTML Input-Forms

Commonly used types



HTML Input Types

1 type="text"
Defines a single-line text field
Hello World

2 type="email"
Defines a field for an e-mail address
info@gmail.com

3 type="password"
Defines a password field
•••••

4 type="tel"
field for entering a telephone num
+987654789

5 type="number"
field for entering a number, max min value allowed
26

6 type="search"
text field for entering a search string
search here

7 type="url"
field for entering a URL
https://twitter.com

8 type="file"
file-select field and a "Browse" button for file uploads
Choose File No file chosen

9 type="hidden"
A hidden field let web developers include data that cannot be seen or modified by users when a form is submitted.

10 type="button"
defines a clickable button
Click me

11 type="reset"
reset button which resets all form values to its initial values
Reset

12 type="submit"
button which submits all form values.
Submit

13 type="radio"
Only one radio button in a group can be selected at the same time.
◉ 5 - 20
○ 21 - 35
○ 36 - 45

14 type="checkbox"
checkbox let a user select one or more options
☑ HTML
☑ CSS
☐ JavaScript

15 type="range"
defines a control for enterin
a number

16 type="image"
defines image as a submit button

17 type="color"
defines a color picker the default value is black.

18 type="month"
defines a month and year con
April 2022

19 type="date"
defines a date picker. The resulting value includes year, month, and day.
04/18/2022

20 type="time"
defines a control for entering a time.
12:22 PM

21 type="week"
defines a week and year contr
Week 16, 2022

Designing HTML-Forms

- ⌘ **Label**: It describes the purpose of the form element.
- ⌘ **Input**: It accepts the data entered in the form, such as text, password, email, address, etc.
- ⌘ **Textarea**: This allows people to enter long text content.
- ⌘ **Button**: It provides a clickable button so an element can perform a function.
- ⌘ **Select**: It lets users scroll through a list of available options in a drop-down list box and choose one from them.
- ⌘ **Fieldset**: It can enclose the various form elements and group-related data in a box.
- ⌘ **Legend**: It puts captions for fieldset components.
- ⌘ **Datalist**: It identifies pre-defined list options for input controls.
- ⌘ **Output**: It displays the results of the performed calculations.
- ⌘ **Option**: It defines available choices in a drop-down list.
- ⌘ **Opt group**: It creates group options in a drop-down list.

To create forms, you need to put input fields, preformatted text, lists, and tables.

Designing HTML-Forms [Elements of the forms-Inputs]

- ⌘ `<input type="button">`
- ⌘ `<input type="checkbox">`
- ⌘ `<input type="color">`
- ⌘ `<input type="date">`
- ⌘ `<input type="datetime-local">`
- ⌘ `<input type="email">`
- ⌘ `<input type="file">`
- ⌘ `<input type="hidden">`
- ⌘ `<input type="image">`
- ⌘ `<input type="month">`
- ⌘ `<input type="number">`
- ⌘ `<input type="password">`
- ⌘ `<input type="radio">`
- ⌘ `<input type="range">`
- ⌘ `<input type="reset">`
- ⌘ `<input type="search">`
- ⌘ `<input type="submit">`
- ⌘ `<input type="tel">`
- ⌘ `<input type="text">`
- ⌘ `<input type="time">`
- ⌘ `<input type="url">`
- ⌘ `<input type="week">`

22 different types of inputs.
[Overall, there might be more]

Designing HTML-Forms [Elements of the forms-Label]

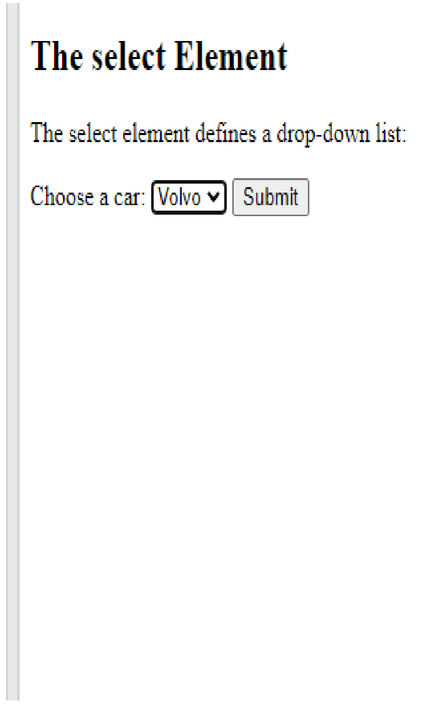
- ❖ The <label> element defines a label for several form elements.
- ❖ The <label> element is useful for screen-reader users, because the screen-reader will read out loud the label when the user focus on the input element.
- ❖ The <label> element also help users who have difficulty clicking on very small regions (such as radio buttons or checkboxes) - because when the user clicks the text within the <label> element, it toggles the radio button/checkbox.
- ❖ The for attribute of the <label> tag should be equal to the id attribute of the <input> element to bind them together.

Designing HTML-Forms [Elements of the forms-Select]

The <select> element defines a drop-down list.

```
<label for="cars">Choose a car:</label>
<select id="cars" name="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

```
<!DOCTYPE html>
<html>
<body>

<h2>The select Element</h2>

<p>The select element defines a drop-down list:</p>

<form action="/action_page.php">
  <label for="cars">Choose a car:</label>
  <select id="cars" name="cars">
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
    <option value="fiat">Fiat</option>
    <option value="audi">Audi</option>
  </select>
  <input type="submit">
</form>

</body>
</html>
```



The select Element

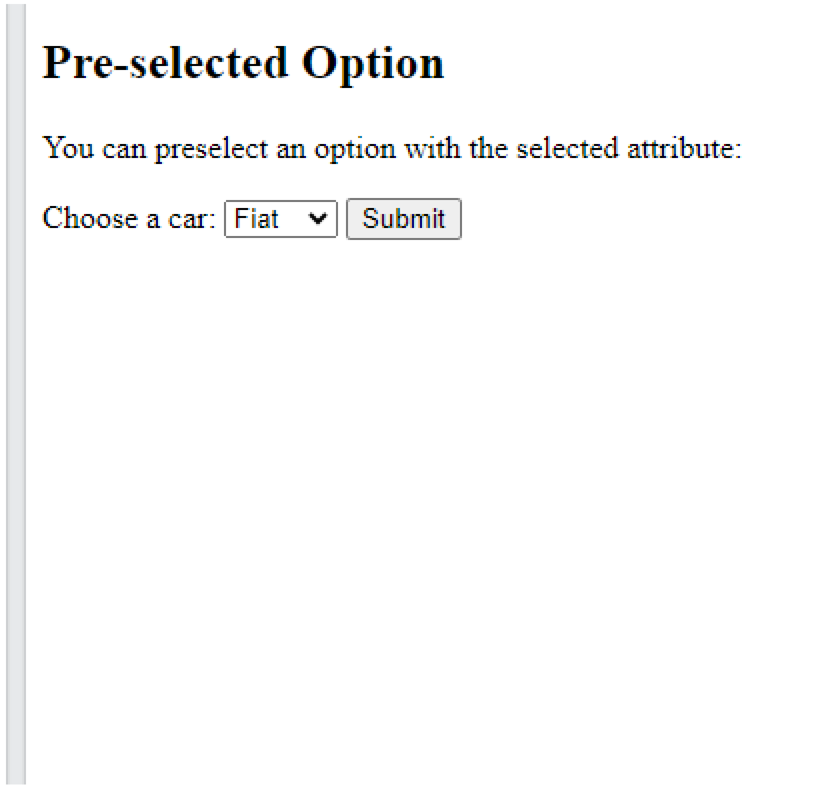
The select element defines a drop-down list:

Choose a car: Volvo Submit

Designing HTML-Forms [Elements of the forms-Select]

- The <option> element defines an option that can be selected.
- By default, the first item in the drop-down list is selected.
- To define a pre-selected option, add the selected attribute to the option

```html
<!DOCTYPE html>
<html>
<body>

<h2>Pre-selected Option</h2>

<p>You can preselect an option with the selected attribute:</p>

<form action="/action_page.php">
  <label for="cars">Choose a car:</label>
  <select id="cars" name="cars">
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
    <option value="fiat" selected>Fiat</option>
    <option value="audi">Audi</option>
  </select>
  <input type="submit">
</form>

</body>
</html>
```



Pre-selected Option

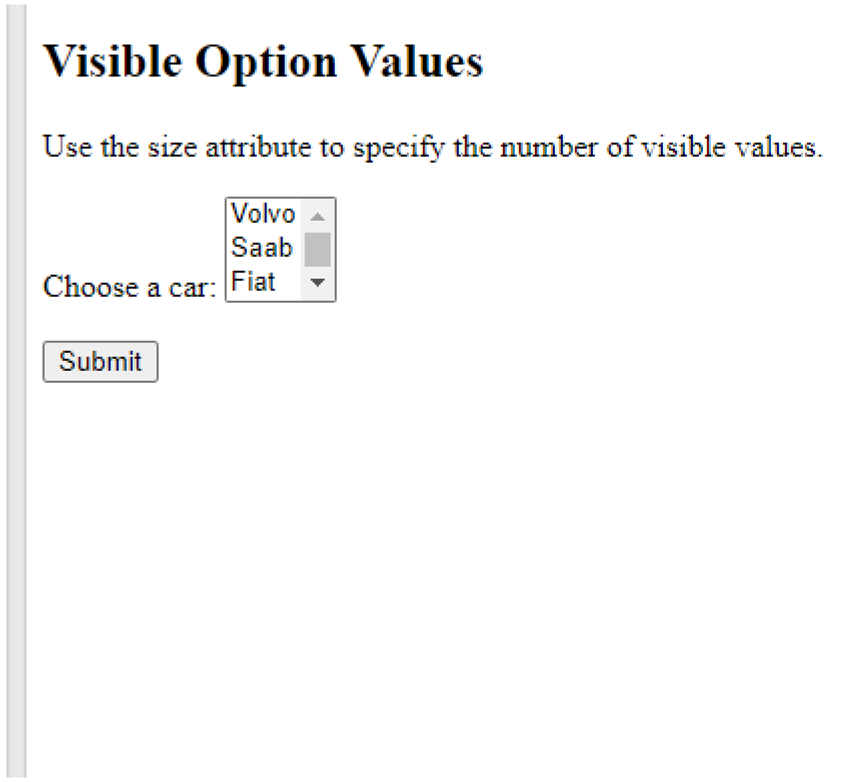
You can preselect an option with the selected attribute:

Designing HTML-Forms [Elements of the forms-Select]

Visible Values:

Use the size attribute to specify the number of visible values.

```
<!DOCTYPE html>
<html>
<body>

<h2>Visible Option Values</h2>

<p>Use the size attribute to specify the number of visible values.</p>

<form action="/action_page.php">
  <label for="cars">Choose a car:</label>
  <select id="cars" name="cars" size="3">
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
    <option value="fiat">Fiat</option>
    <option value="audi">Audi</option>
  </select><br><br>
  <input type="submit">
</form>

</body>
</html>
```



Visible Option Values

Use the size attribute to specify the number of visible values.

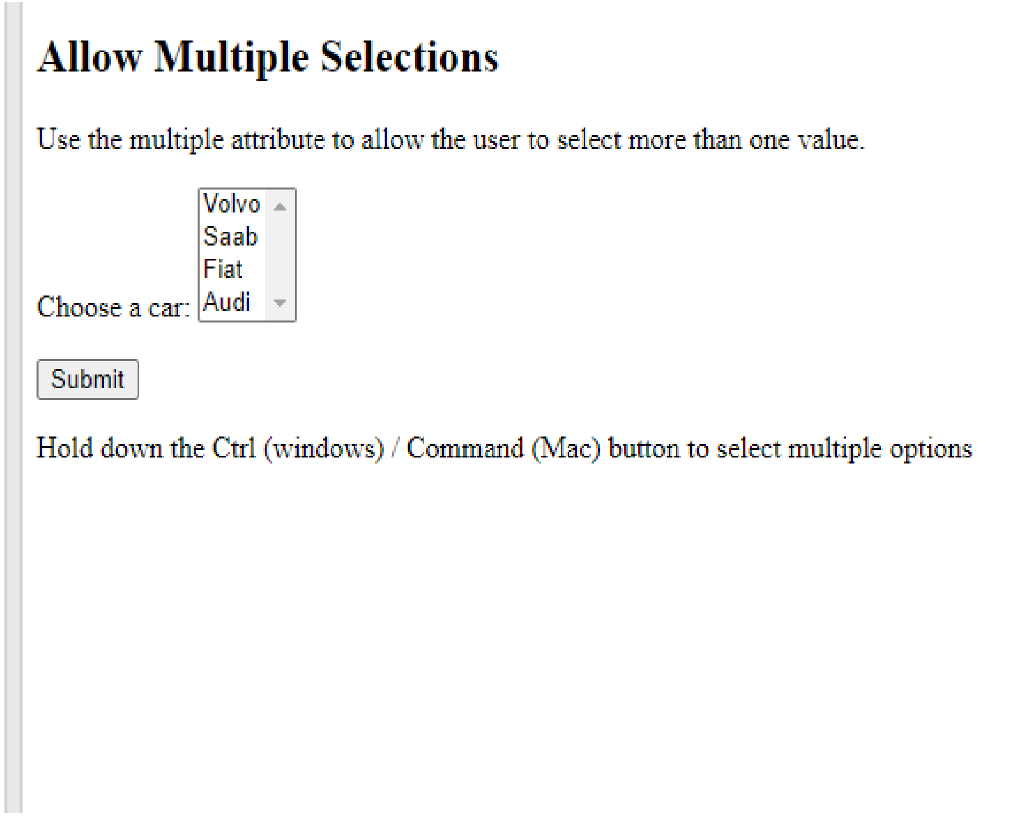
Choose a car:

Designing HTML-Forms [Elements of the forms-Select]

Allow Multiple Selections:

Use the multiple attribute to allow the user to select more than one value.

```
<!DOCTYPE html>
<html>
<body>

<h2>Allow Multiple Selections</h2>

<p>Use the multiple attribute to allow the user to select more than one value.</p>

<form action="/action_page.php">
  <label for="cars">Choose a car:</label>
  <select id="cars" name="cars" size="4" multiple>
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
    <option value="fiat">Fiat</option>
    <option value="audi">Audi</option>
  </select><br><br>
  <input type="submit">
</form>

<p>Hold down the Ctrl (windows) / Command (Mac) button to select multiple options.</p>

</body>
</html>
```



Allow Multiple Selections

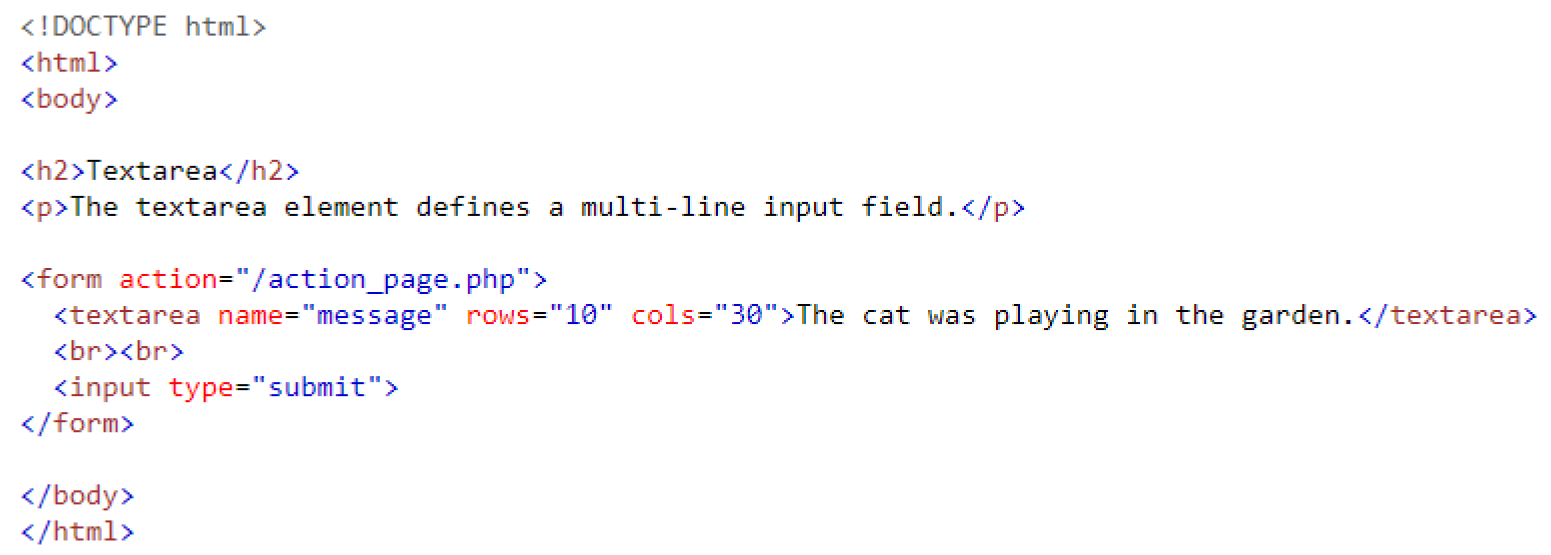
Use the multiple attribute to allow the user to select more than one value.

Choose a car: Volvo / Saab / Fiat / Audi

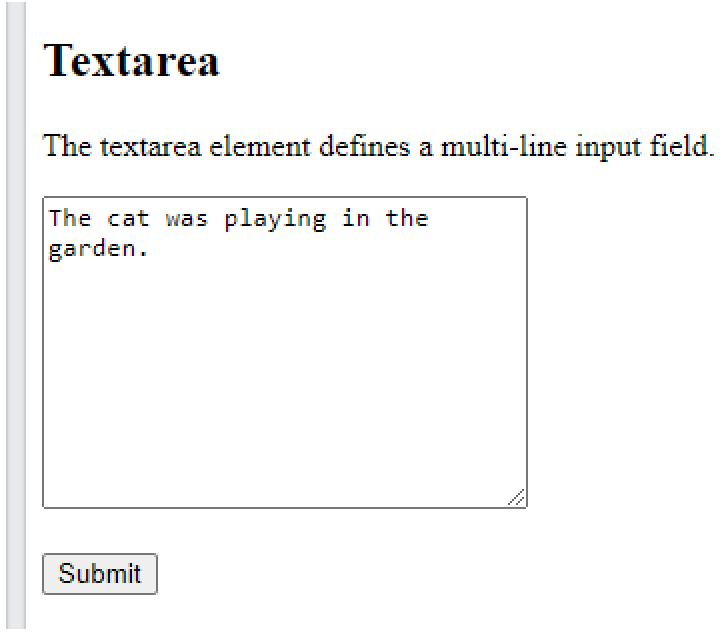
Hold down the Ctrl (windows) / Command (Mac) button to select multiple options

Designing HTML-Forms [Elements of the forms-Text-Area]

⌘ The <textarea> element defines a multi-line input field (a text area).



```
<!DOCTYPE html>
<html>
<body>

<h2>Textarea</h2>
<p>The textarea element defines a multi-line input field.</p>

<form action="/action_page.php">
  <textarea name="message" rows="10" cols="30">The cat was playing in the garden.</textarea>
  <br><br>
  <input type="submit">
</form>

</body>
</html>
```



Textarea

The textarea element defines a multi-line input field.

```
The cat was playing in the
garden.
```

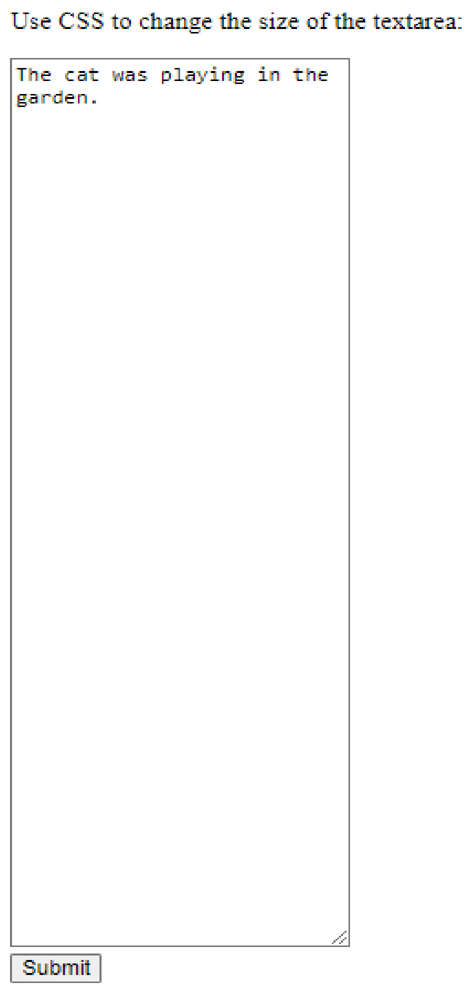
⌘ The rows attribute specifies the visible number of lines in a text area.

⌘ The cols attribute specifies the visible width of a text area.

Designing HTML-Forms [Elements of the forms-Text-Area]

You can also define the size of the text area by using CSS.

```
<textarea name="message" style="width:200px; height:600px;">
The cat was playing in the garden.
</textarea>
```

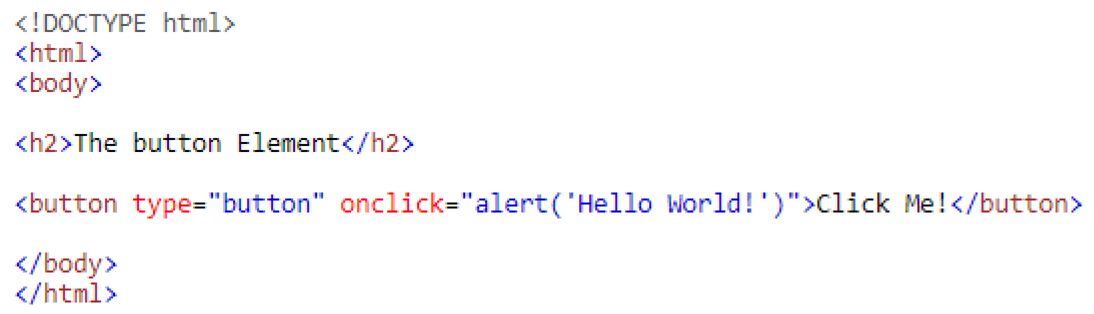

Use CSS to change the size of the textarea:

```
The cat was playing in the
garden.
```

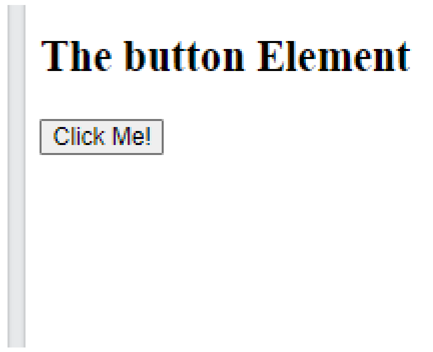
- The rows attribute specifies the visible number of lines in a text area.
- The cols attribute specifies the visible width of a text area.

Designing HTML-Forms [Elements of the forms-Button]

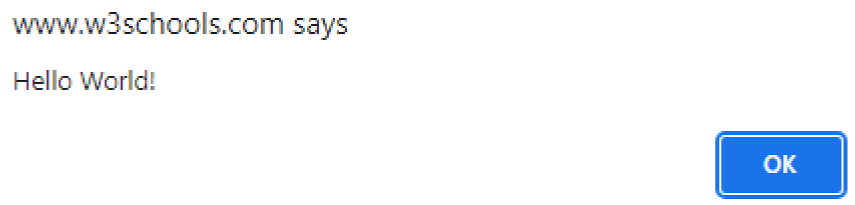
⌘ The <button> element defines a clickable button.



```
<!DOCTYPE html>
<html>
<body>

<h2>The button Element</h2>

<button type="button" onclick="alert('Hello World!')">Click Me!</button>

</body>
</html>
```



The button Element



www.w3schools.com says

Hello World!

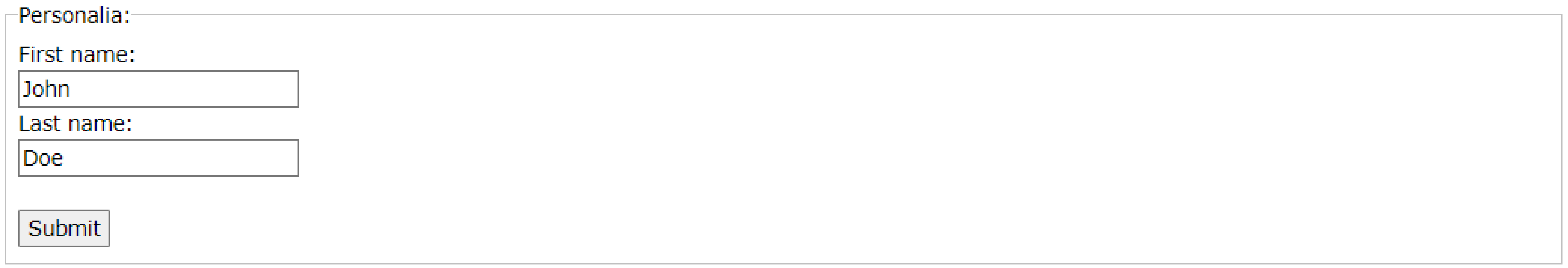
Note: Always specify the type attribute for the button element. Different browsers may use different default types for the button element.

Designing HTML-Forms [Elements of the forms-Field set & Legend]

- ⌘ The <fieldset> element is used to group related data in a form.
- ⌘ The <legend> element defines a caption for the <fieldset> element

```
<form action="/action_page.php">
  <fieldset>
    <legend>Personalia:</legend>
    <label for="fname">First name:</label><br>
    <input type="text" id="fname" name="fname" value="John"><br>
    <label for="lname">Last name:</label><br>
    <input type="text" id="lname" name="lname" value="Doe"><br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```

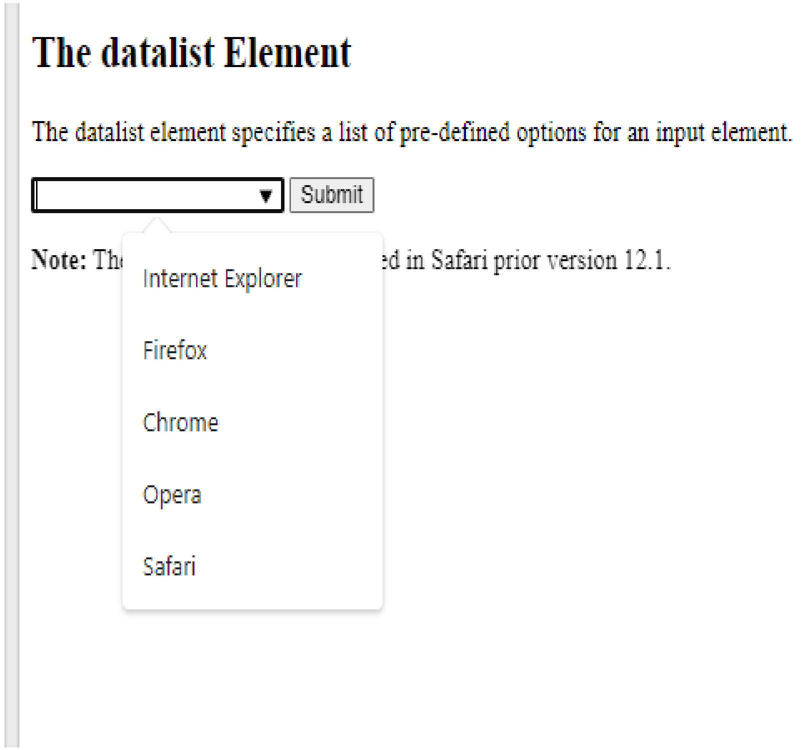
This is how the HTML code above will be displayed in a browser:


Personalia:
First name:
John
Last name:
Doe
Submit

Designing HTML-Forms [Elements of the forms-Data list]

- ⌘ The <datalist> element specifies a list of pre-defined options for an <input> element.
- ⌘ Users will see a drop-down list of the pre-defined options as they input data.
- ⌘ The list attribute of the <input> element, must refer to the id attribute of the <datalist> element.

```
<!DOCTYPE html>
<html>
<body>

<h2>The datalist Element</h2>

<p>The datalist element specifies a list of pre-defined options for an input element.</p>

<form action="/action_page.php">
  <input list="browsers" name="browser">
  <datalist id="browsers">
    <option value="Internet Explorer">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist>
  <input type="submit">
</form>

<p><b>Note:</b> The datalist tag is not supported in Safari prior version 12.1.</p>

</body>
</html>
```



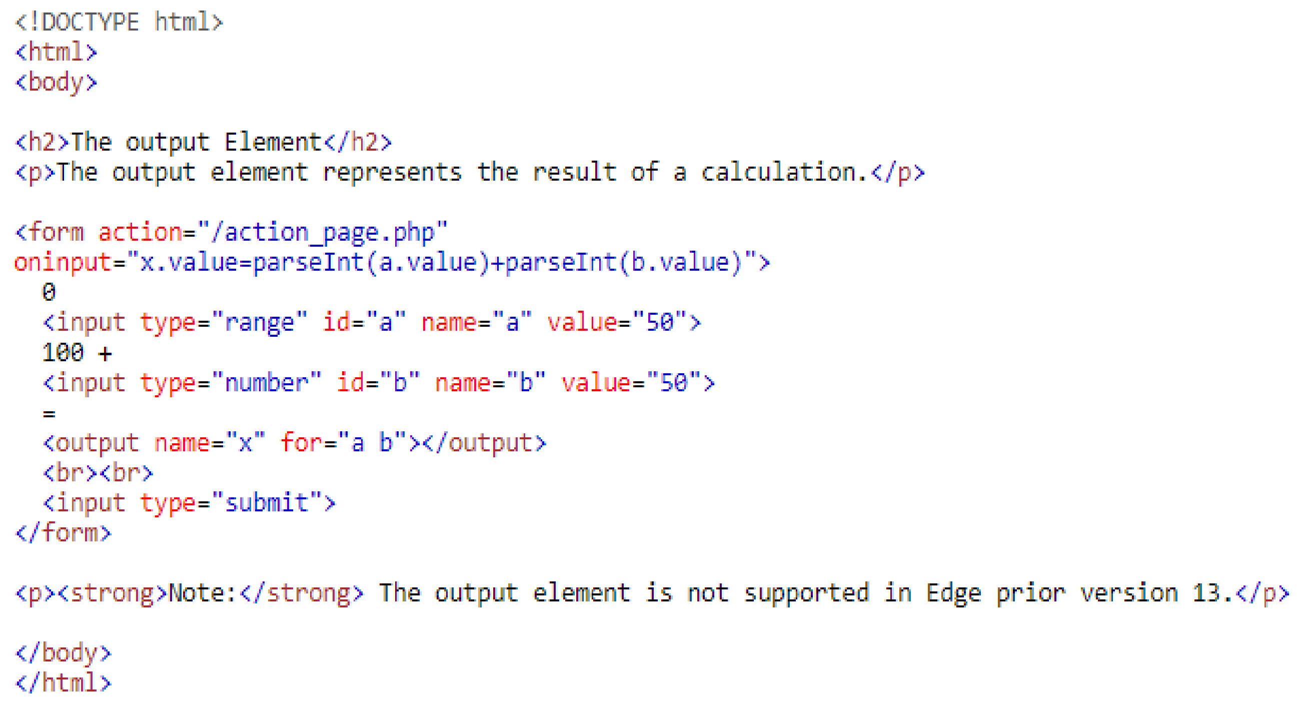
The datalist Element

The datalist element specifies a list of pre-defined options for an input element.

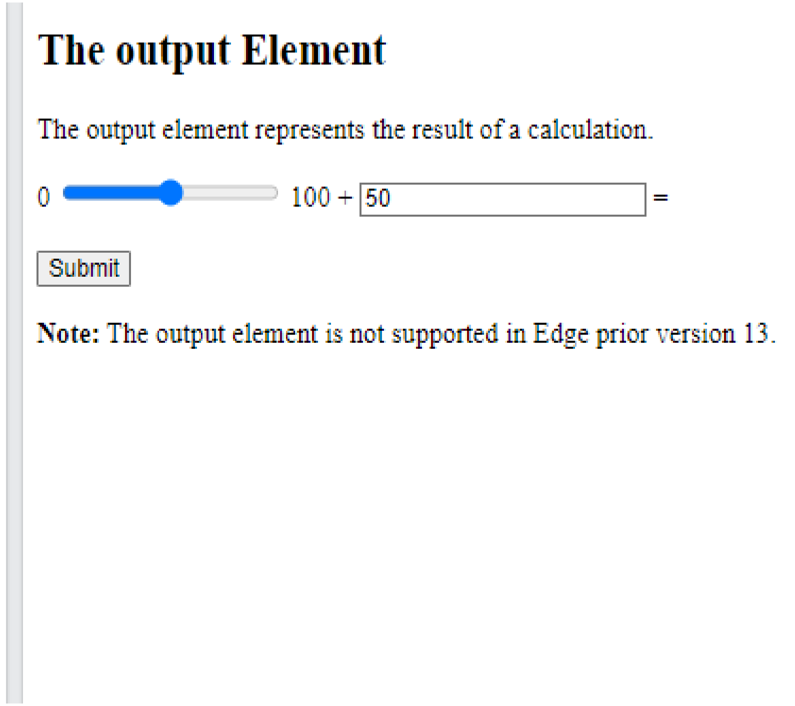
Note: Th ed in Safari prior version 12.1.

Designing HTML-Forms [Elements of the forms-Output]

⌘ The <output> element represents the result of a calculation (like one performed by a script).



```
<!DOCTYPE html>
<html>
<body>

<h2>The output Element</h2>
<p>The output element represents the result of a calculation.</p>

<form action="/action_page.php"
oninput="x.value=parseInt(a.value)+parseInt(b.value)">
  0
  <input type="range" id="a" name="a" value="50">
  100 +
  <input type="number" id="b" name="b" value="50">
  =
  <output name="x" for="a b"></output>
  <br><br>
  <input type="submit">
</form>

<p><strong>Note:</strong> The output element is not supported in Edge prior version 13.</p>

</body>
</html>
```



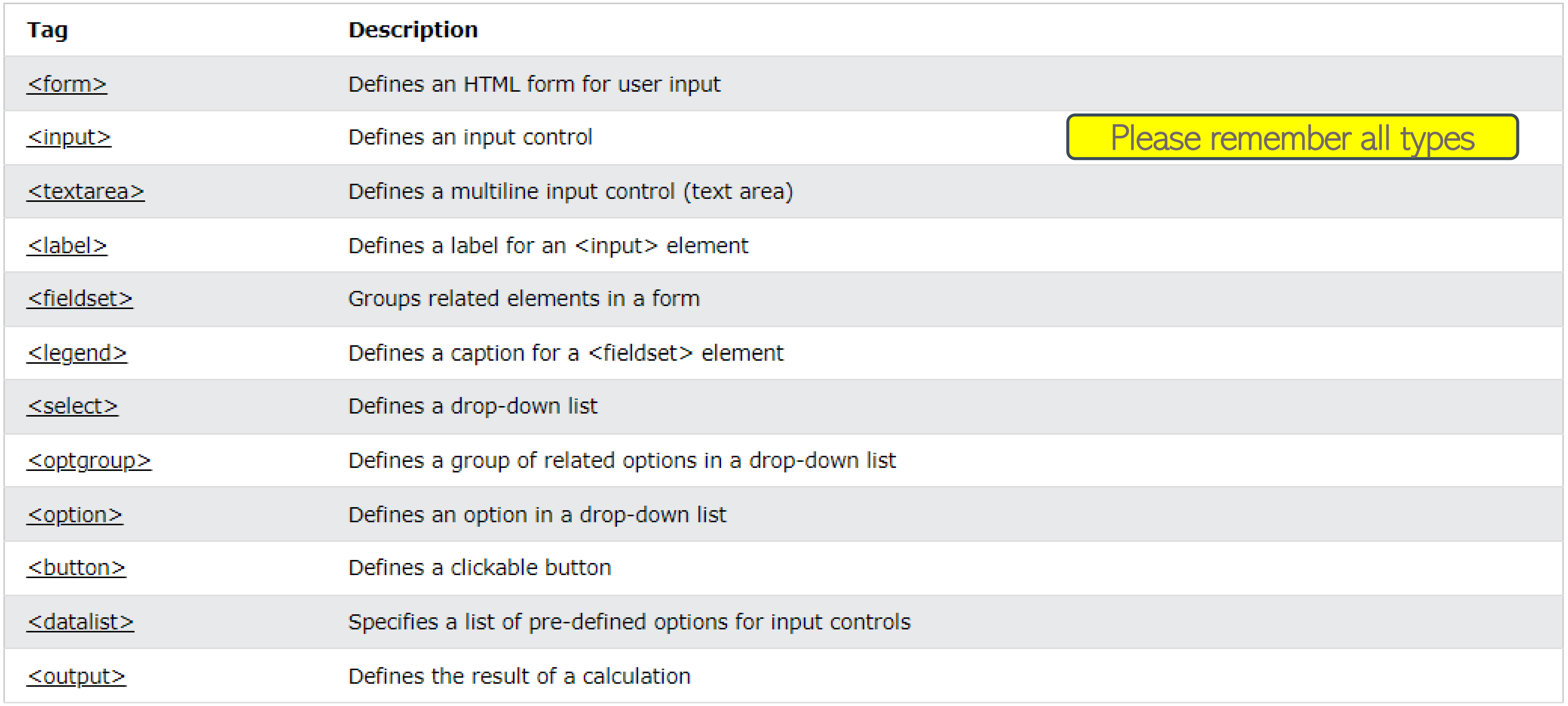
The output Element

The output element represents the result of a calculation.

0 100 + 50 =

Note: The output element is not supported in Edge prior version 13.

Designing HTML-Forms [Summary of Elements]

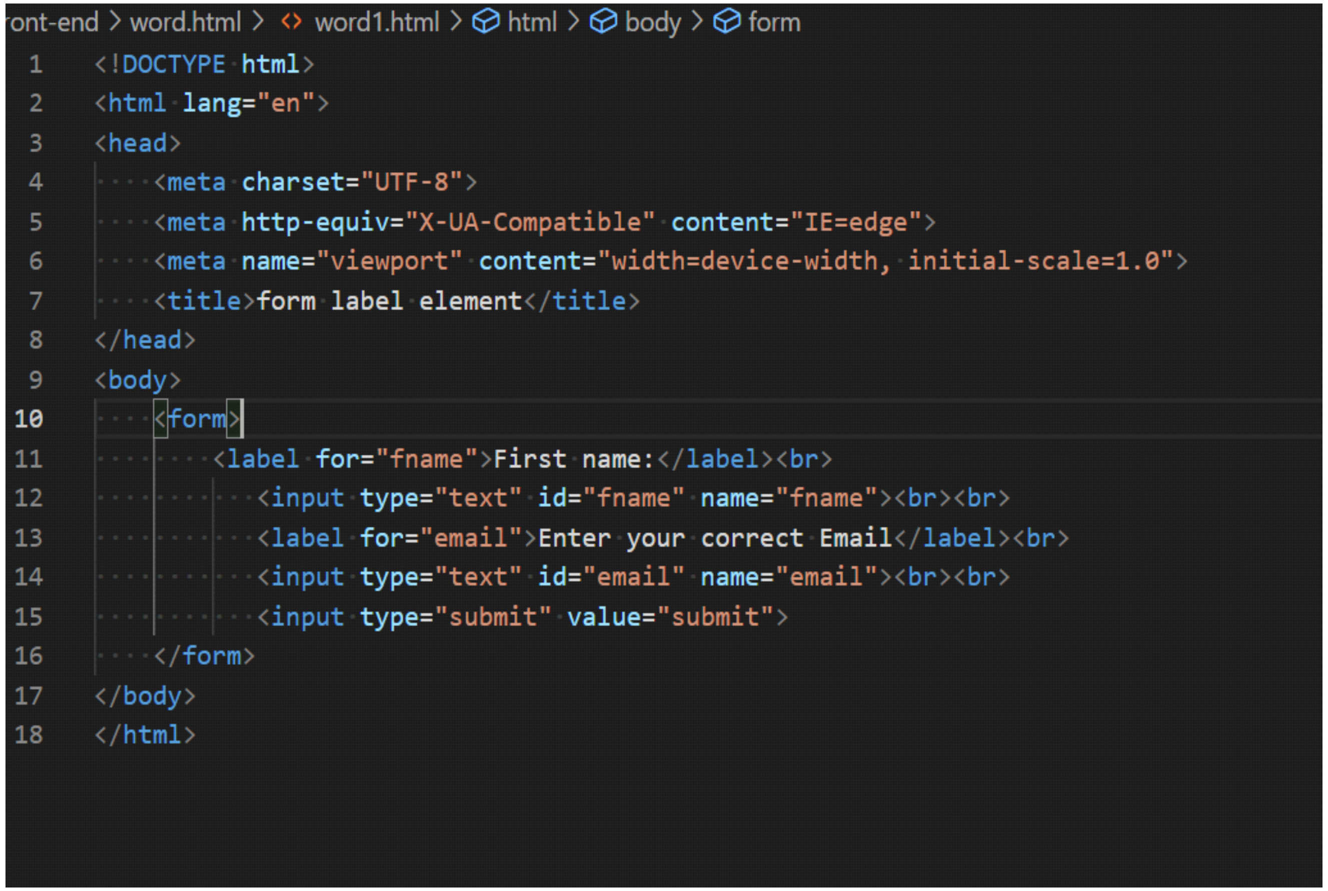


Tag	Description
<form>	Defines an HTML form for user input
<input>	Defines an input control
<textarea>	Defines a multiline input control (text area)
<label>	Defines a label for an <input> element
<fieldset>	Groups related elements in a form
<legend>	Defines a caption for a <fieldset> element
<select>	Defines a drop-down list
<optgroup>	Defines a group of related options in a drop-down list
<option>	Defines an option in a drop-down list
<button>	Defines a clickable button
<datalist>	Specifies a list of pre-defined options for input controls
<output>	Defines the result of a calculation

Please remember all types

Summary of the Forms

- HTML Forms
- Forms Elements
- Forms Attributes-Input
  - ⌘Forms Input Types<input type="button">
  - ⌘<input type="checkbox">
  - ⌘<input type="color">
  - ⌘<input type="date">
  - ⌘<input type="datetime-local">
  - ⌘<input type="email">
  - ⌘<input type="file">
  - ⌘<input type="hidden">
  - ⌘<input type="image">
  - ⌘<input type="month">
  - ⌘<input type="number">
- ❑ Forms Attributes-label
- ❑ Forms Attributes-select



```
<!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>form label element</title>
</head>
<body>
    <form>
        <label for="fname">First name:</label><br>
            <input type="text" id="fname" name="fname"><br><br>
            <label for="email">Enter your correct Email</label><br>
            <input type="text" id="email" name="email"><br><br>
            <input type="submit" value="submit">
    </form>
</body>
</html>
```

JavaScript

World's most popular programming language!!!

JavaScript Concepts-Why Study JavaScript?

⌘ JavaScript is one of the 3 languages all web developers must learn:

1. HTML to define the content of web pages.

2. CSS to specify the layout of web pages.

3. JavaScript to program the behaviour of web pages.

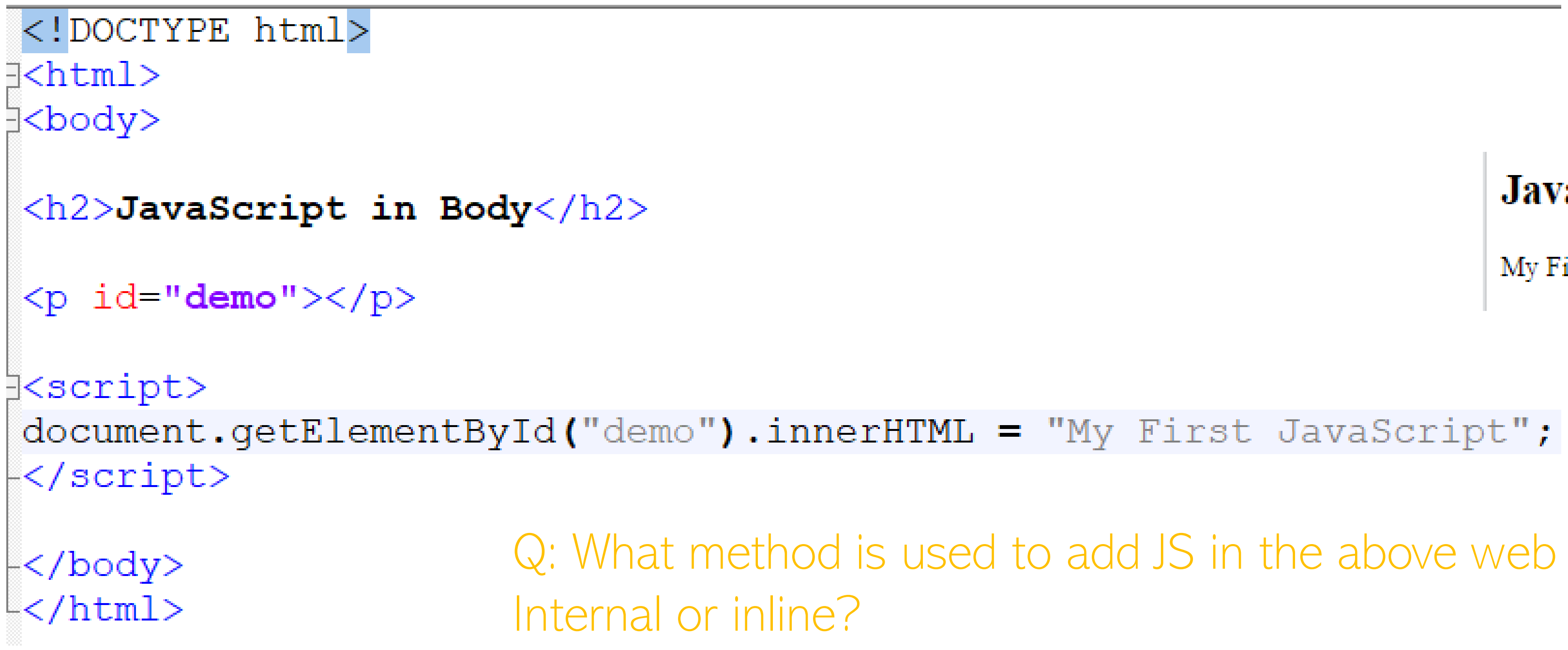
JavaScript Concepts-JS Functions & Events

- A JavaScript function is a block of JavaScript code, that can be executed when "called" for.
- For example, a function can be called when an **event** occurs, like when the user clicks a button.
- Event can be a click, mouseover, page open, page close, etc.
- You will learn much more about functions and events in later lectures.

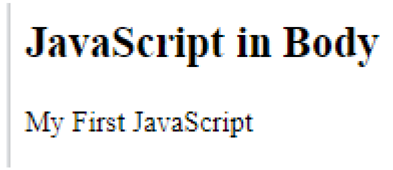
JavaScript Coding Examples- JavaScript Where To

- Just like CSS, the JS can also be included in the web pages in three ways.
- (1) Internal, (2) External, and inline.

In HTML, JavaScript code is inserted between <script> and </script> tags.



```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript in Body</h2>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "My First JavaScript";
</script>

</body>
</html>
```



JavaScript in Body

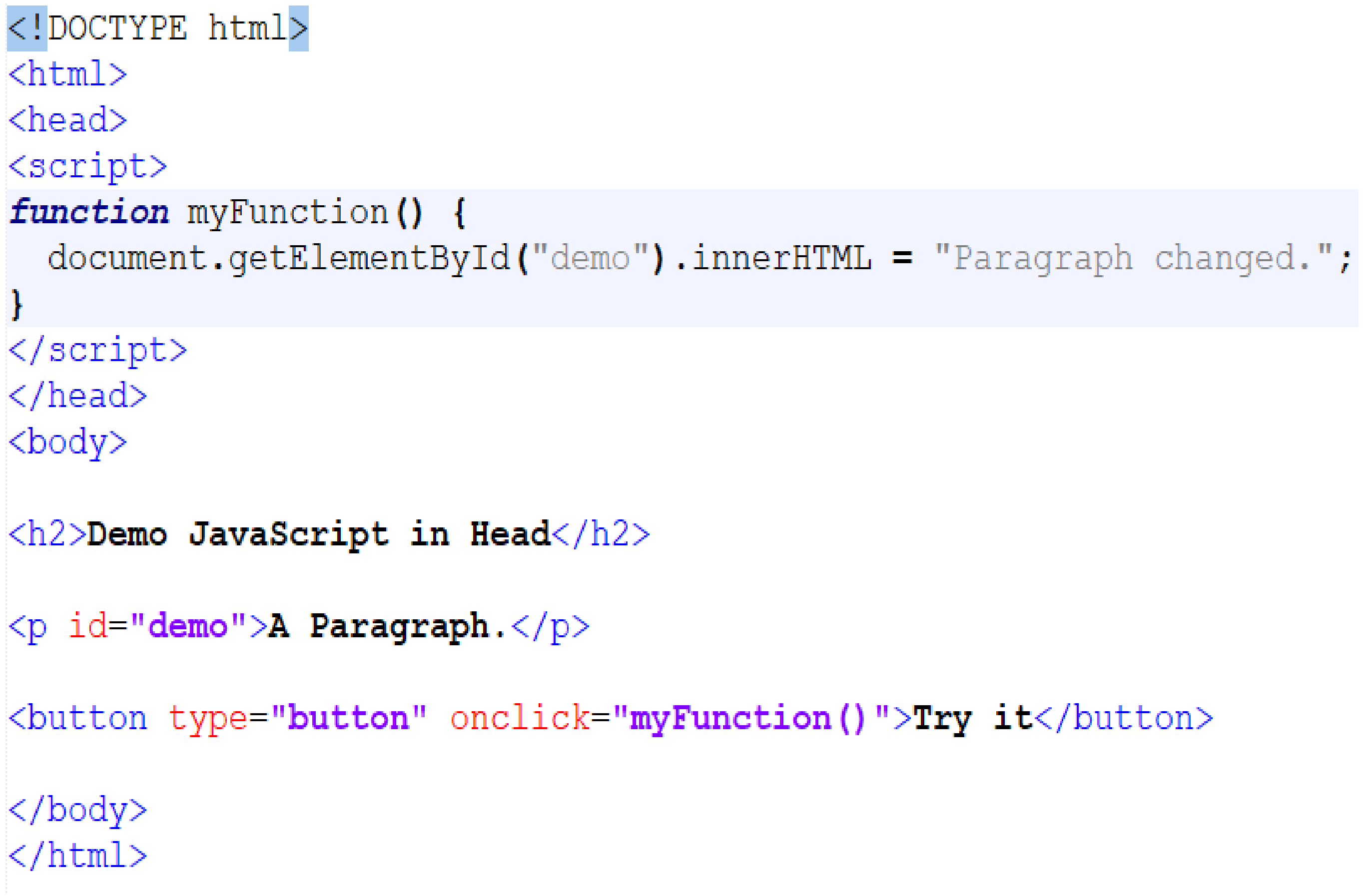
My First JavaScript

Q: What method is used to add JS in the above web page? Internal or inline?

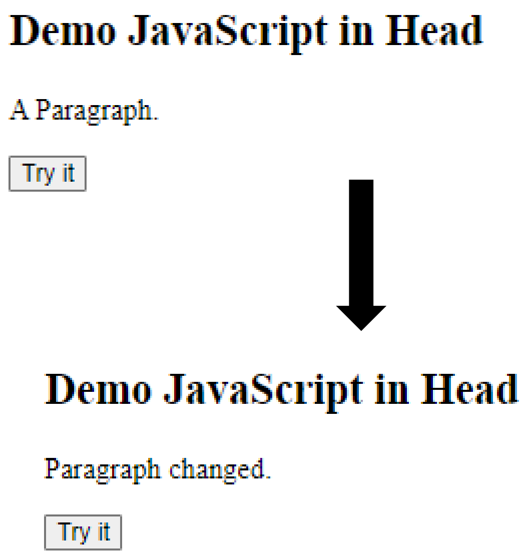
You can place any number of scripts in an HTML document. Scripts can be placed in the <body>, or in the <head> section of an HTML page, or in both.

JavaScript Coding Examples- JavaScript Where To

1. In this example, a JavaScript function is placed in the <head> section of an HTML page.
2. The function is invoked (called) when a button is clicked:



```
<!DOCTYPE html>
<html>
<head>
<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>
</head>
<body>

<h2>Demo JavaScript in Head</h2>

<p id="demo">A Paragraph.</p>

<button type="button" onclick="myFunction()">Try it</button>

</body>
</html>
```



Demo JavaScript in Head

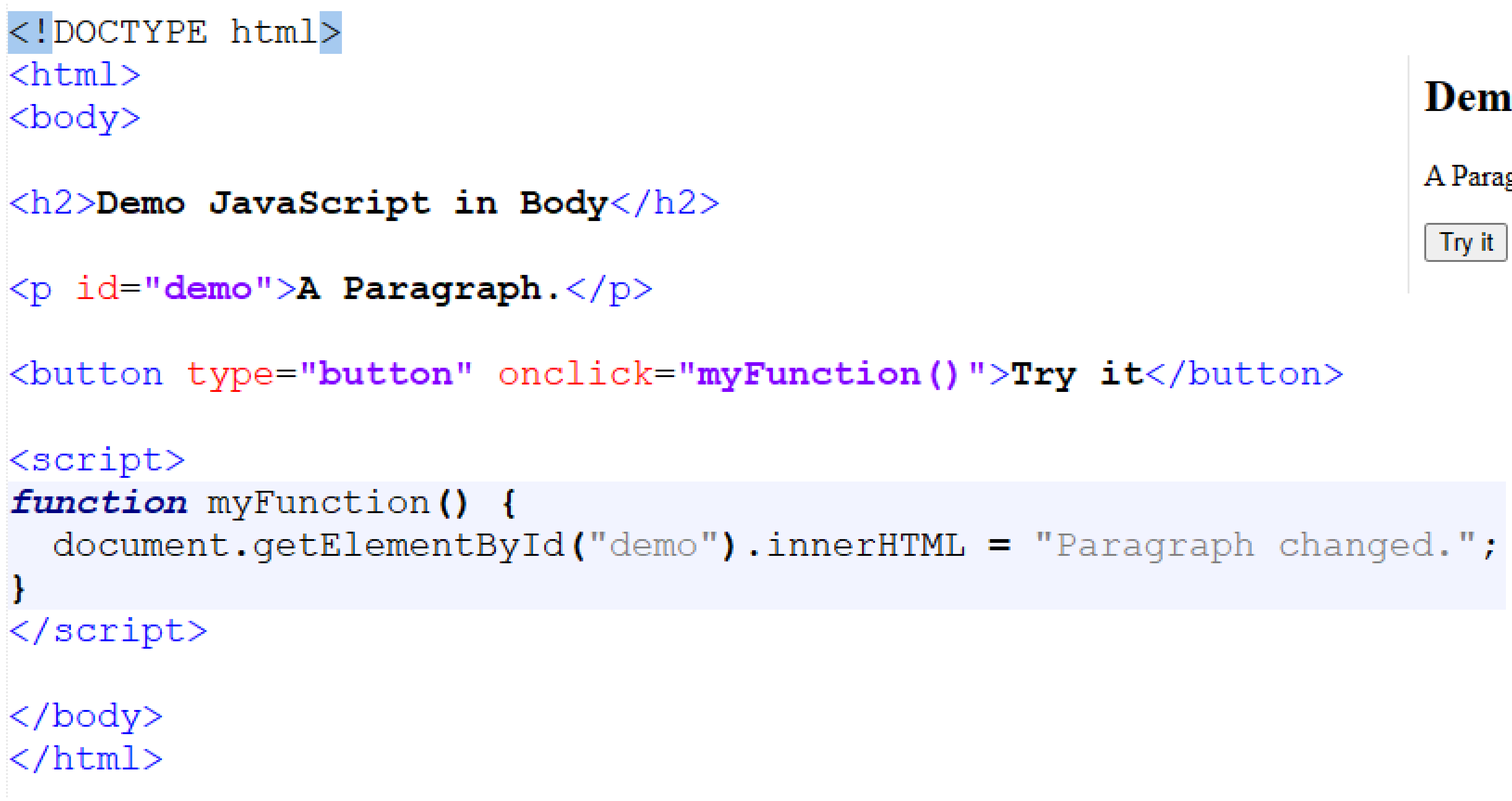
A Paragraph.

Demo JavaScript in Head

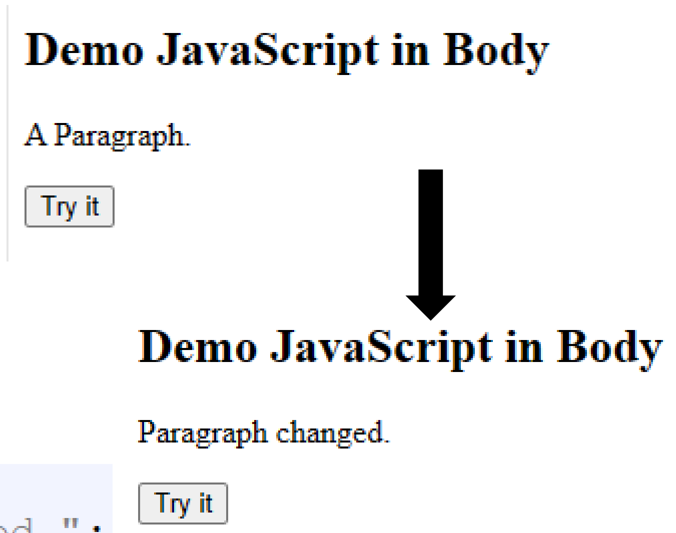
Paragraph changed.

JavaScript Coding Examples- JavaScript Where To

1. In this example, a JavaScript function is placed in the <body> section of an HTML page.
2. The function is invoked (called) when a button is clicked:



```
<!DOCTYPE html>
<html>
<body>

<h2>Demo JavaScript in Body</h2>

<p id="demo">A Paragraph.</p>

<button type="button" onclick="myFunction()">Try it</button>

<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>

</body>
</html>
```



Demo JavaScript in Body

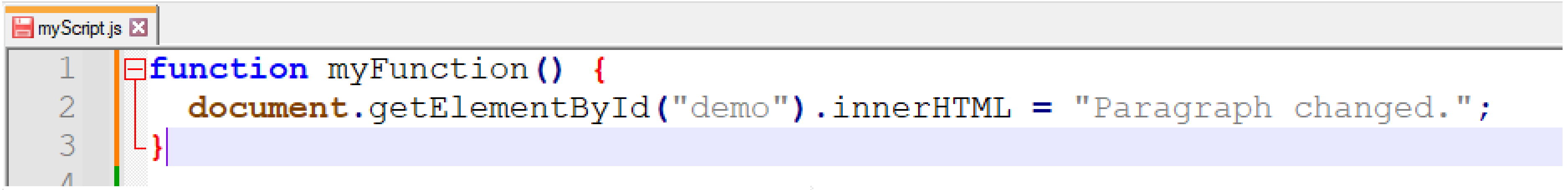
A Paragraph.

Demo JavaScript in Body

Paragraph changed.

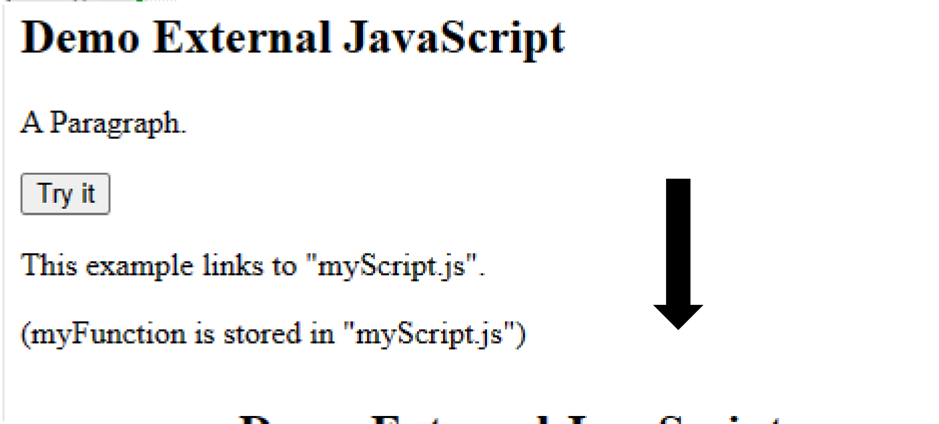
JavaScript Coding Examples- JavaScript Where To

1. Scripts can also be placed in external files



myScript.js

```
function myFunction() {
  document.getElementById("demo").innerHTML = "Paragraph changed.";
}
```



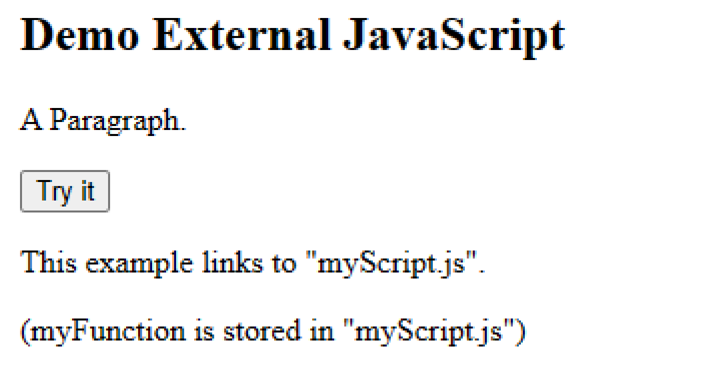
Demo External JavaScript

A Paragraph.

Try it

This example links to "myScript.js".

(myFunction is stored in "myScript.js")



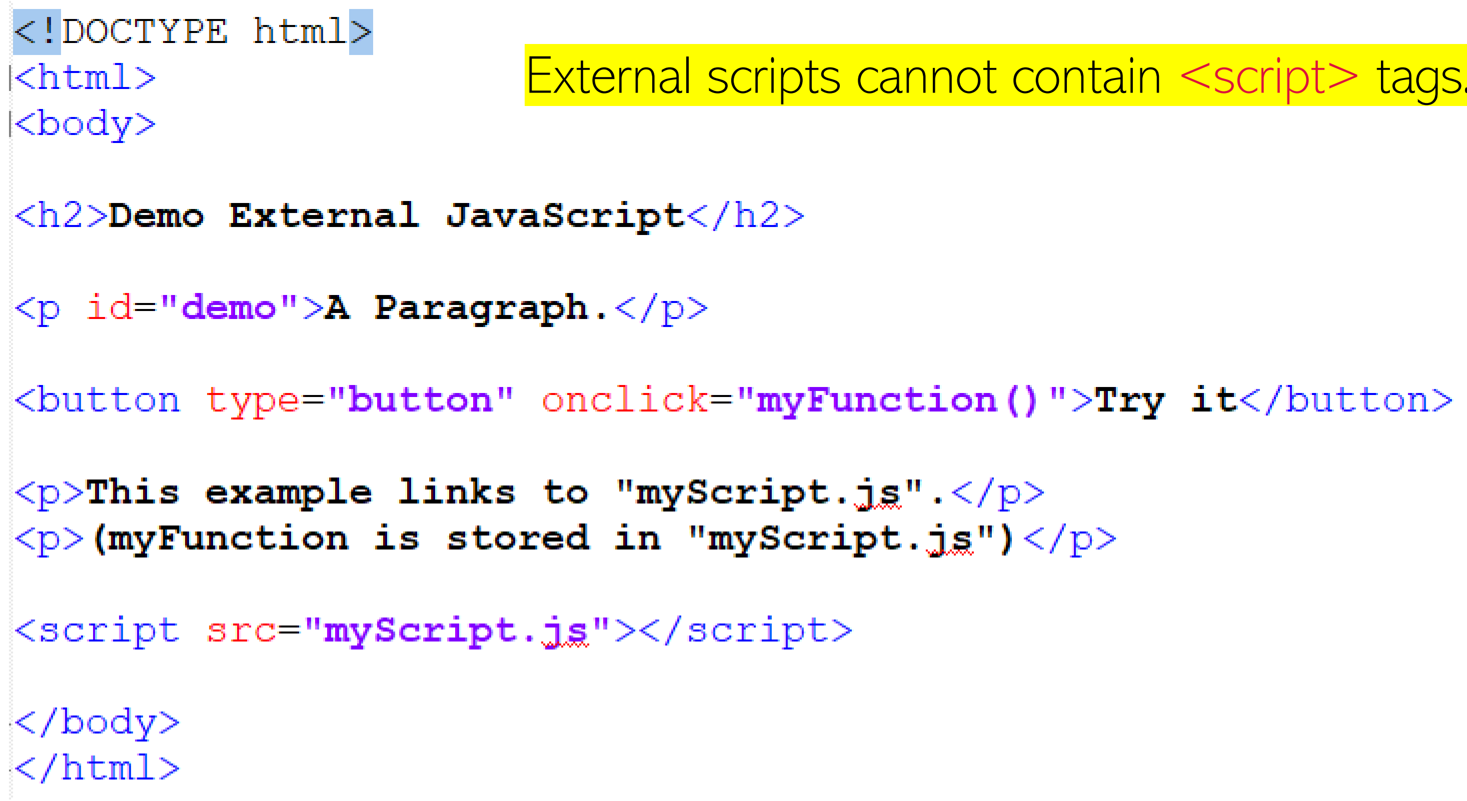
Demo External JavaScript

A Paragraph.

Try it

This example links to "myScript.js".

(myFunction is stored in "myScript.js")



```
<!DOCTYPE html>
<html>
<body>

<h2>Demo External JavaScript</h2>

<p id="demo">A Paragraph.</p>

<button type="button" onclick="myFunction()">Try it</button>

<p>This example links to "myScript.js".</p>
<p>(myFunction is stored in "myScript.js")</p>

<script src="myScript.js"></script>

</body>
</html>
```

External scripts cannot contain <script> tags.

JavaScript Coding Examples- JavaScript Where To

Placing scripts in external files has some advantages:
- It separates HTML and code
- It makes HTML and JavaScript easier to read and maintain
- Cached JavaScript files can speed up page loads

To add several script files to one page- use several script tags:

Example
```
<script src="myScript1.js"></script>
<script src="myScript2.js"></script>
```

An external script can be referenced in 3 different ways:
- With a full URL (a full web address)
- With a file path (like /js/)
- Without any path

This example uses a **full URL** to link to myScript.js:

Example
```
<script src="https://www.w3schools.com/js/myScript.js"></script>
```

This example uses a **file path** to link to myScript.js:

Example
```
<script src="/js/myScript.js"></script>
```

This example uses no path to link to myScript.js:

Example
```
<script src="myScript.js"></script>
```

Displaying Output in JavaScript

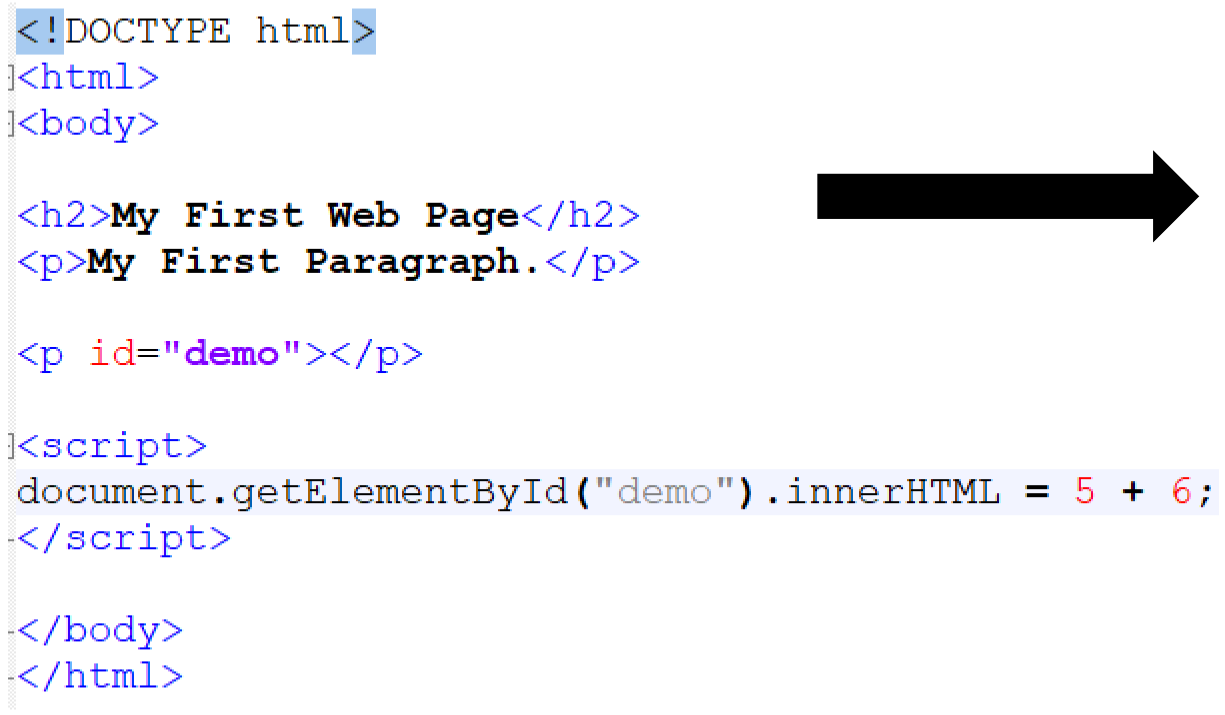
JavaScript Concepts-Displaying Outputs/Data

JavaScript can "display" data in different ways:

- ⌘ Writing into an HTML element, using innerHTML.
- ⌘ Writing into the HTML output using document.write().
- ⌘ Writing into an alert box, using window.alert().
- ⌘ Writing into the browser console, using console.log().

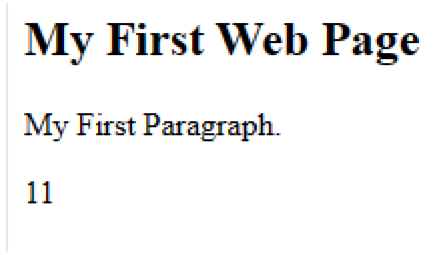
JavaScript Coding Examples-Displaying Outputs/Data- Inner HTML

- To access an HTML element, JavaScript can use the document.getElementById(id) method.
- The id attribute defines the HTML element. The innerHTML property defines the HTML content



```
<!DOCTYPE html>
<html>
<body>

<h2>My First Web Page</h2>
<p>My First Paragraph.</p>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = 5 + 6;
</script>

</body>
</html>
```

Output of the code

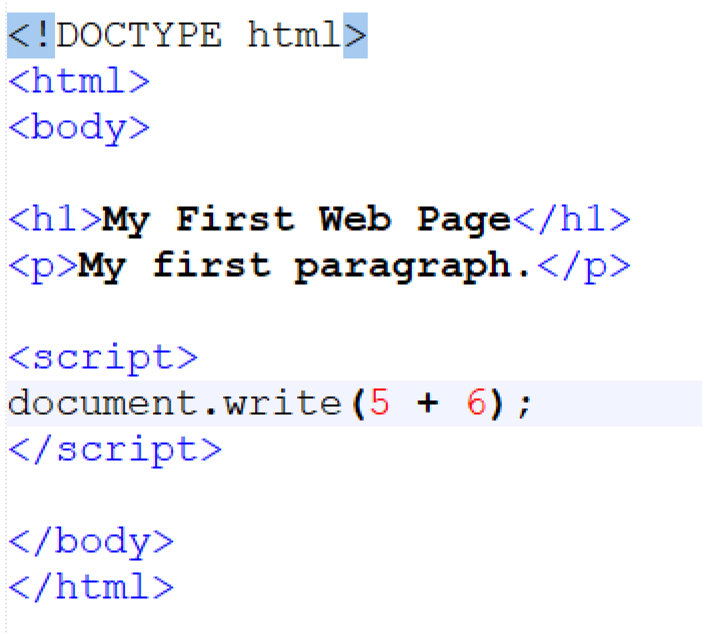


```
My First Web Page

My First Paragraph.

11
```

Changing the innerHTML property of an HTML element is a common way to display data in HTML.

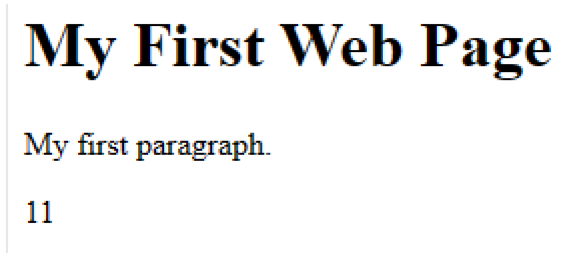
JavaScript Coding Examples-Displaying Outputs/Data- Document Write

⌘ For testing purposes, it is convenient to use document.write().



```
<!DOCTYPE html>
<html>
<body>

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<script>
document.write(5 + 6);
</script>

</body>
</html>
```

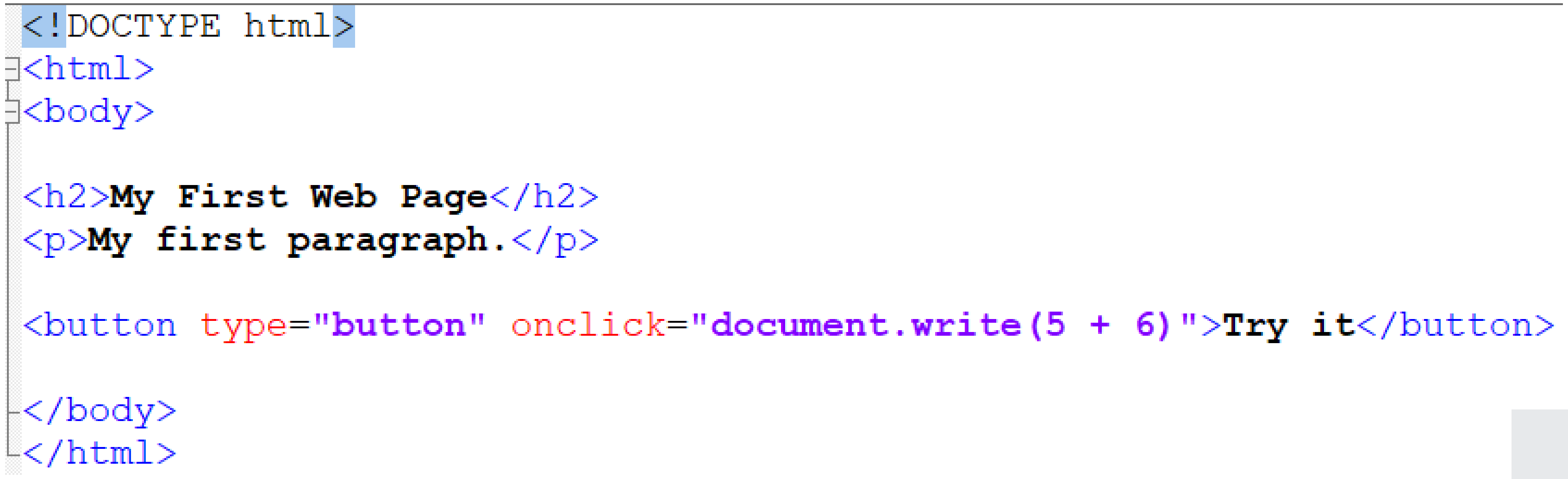
Output of the code



```
My First Web Page

My first paragraph.

11
```

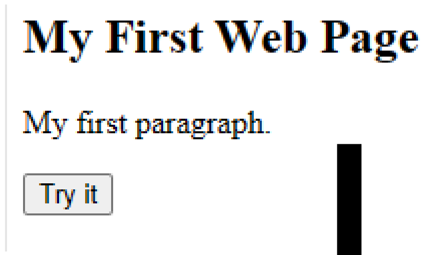
JavaScript Coding Examples-Displaying Outputs/Data- Document Write

For testing purposes, it is convenient to use document.write()



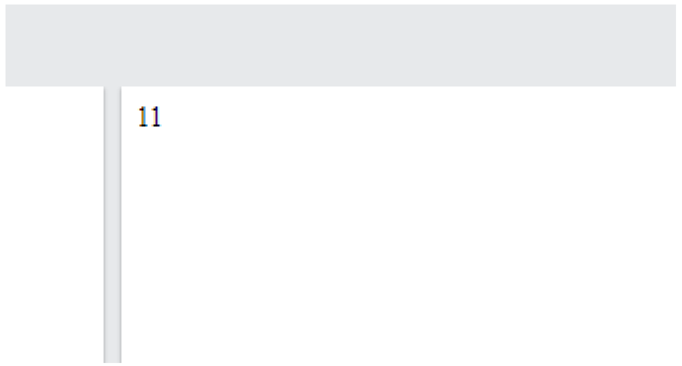
```
<!DOCTYPE html>
<html>
<body>

<h2>My First Web Page</h2>
<p>My first paragraph.</p>

<button type="button" onclick="document.write(5 + 6)">Try it</button>

</body>
</html>
```

Output of the code



My First Web Page

My first paragraph.



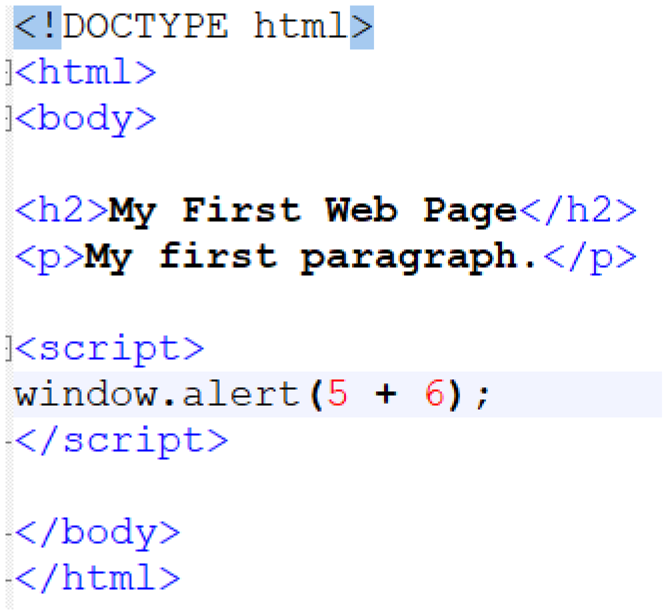
```
11
```

The document.write() method should only be used for testing.

Using document.write() after an HTML document is loaded, will **delete all existing HTML**:

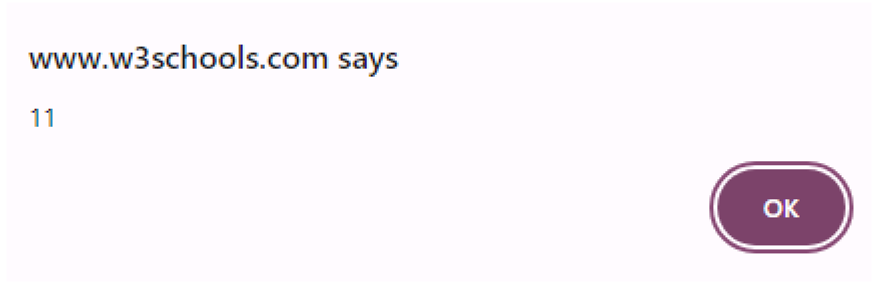
JavaScript Coding Examples-Displaying Outputs/Data- Window Alert

⌘ Alert box can be used to display JavaScript data.

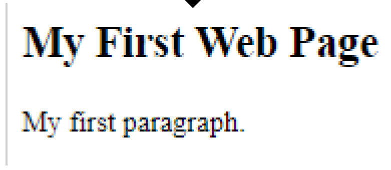


```
<!DOCTYPE html>
<html>
<body>

<h2>My First Web Page</h2>
<p>My first paragraph.</p>

<script>
window.alert(5 + 6);
</script>

</body>
</html>
```

Output of the code



```
www.w3schools.com says
11
```



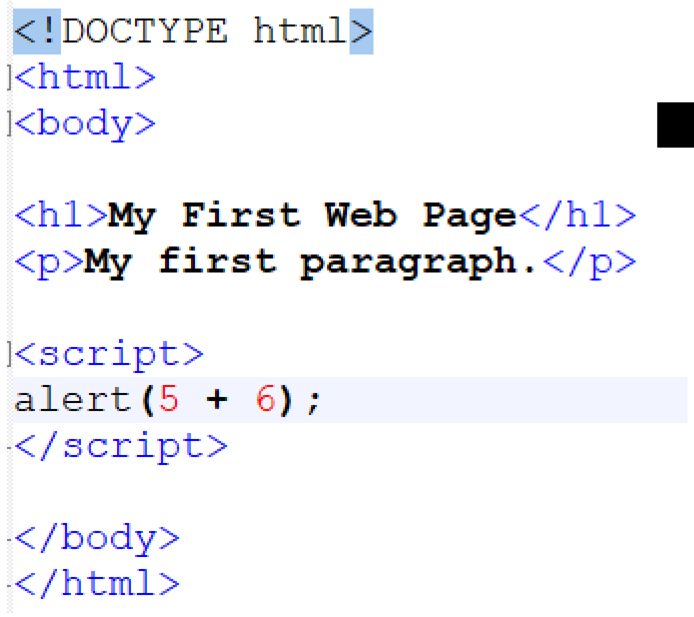
My First Web Page

My first paragraph.

You can skip the window keyword.

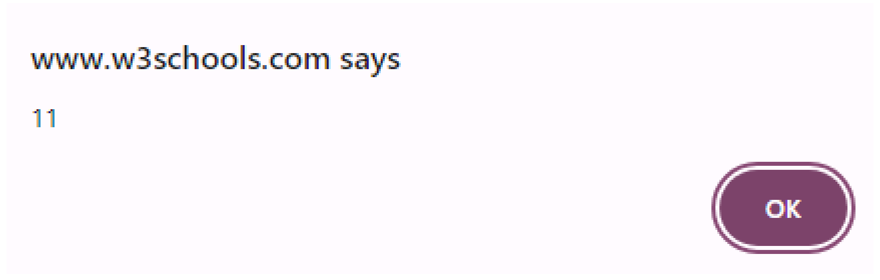
JavaScript Coding Examples-Displaying Outputs/Data- Skip Window Alert

- In JavaScript, the window object is the global scope object. This means that variables, properties, and methods by default belong to the window object. This also means that specifying the window keyword is optional:

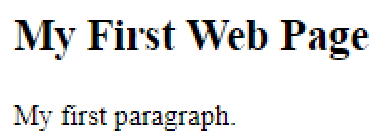


```
<!DOCTYPE html>
<html>
<body>

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<script>
alert(5 + 6);
</script>

</body>
</html>
```

Output of the code



```
www.w3schools.com says
11
```



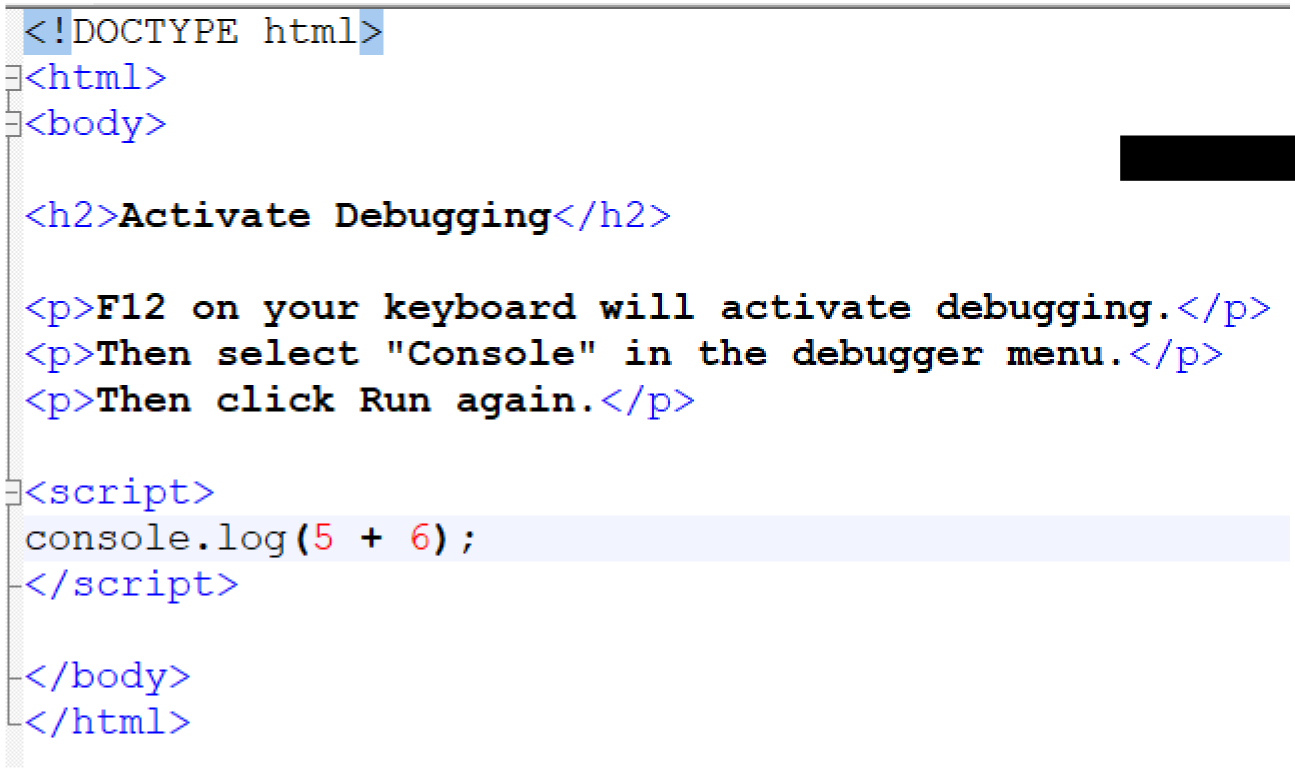
My First Web Page

My first paragraph.

You can skip the window keyword.

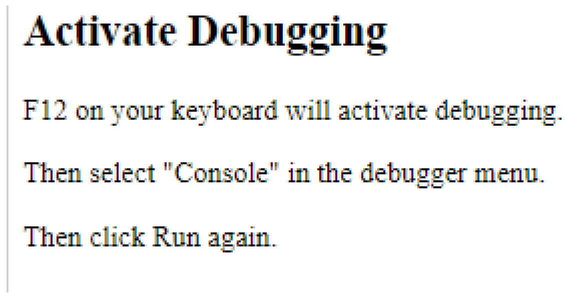
JavaScript Coding Examples-Displaying Outputs/Data- Console Log

⌘ For debugging purposes, you can call the console.log() method in the browser to display data



```
<!DOCTYPE html>
<html>
<body>

<h2>Activate Debugging</h2>

<p>F12 on your keyboard will activate debugging.</p>
<p>Then select "Console" in the debugger menu.</p>
<p>Then click Run again.</p>

<script>
console.log(5 + 6);
</script>

</body>
</html>
```

Output of the code



Activate Debugging

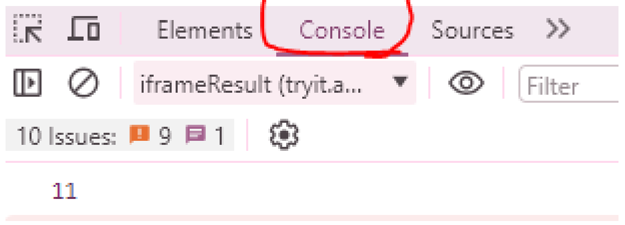
F12 on your keyboard will activate debugging.

Then select "Console" in the debugger menu.

Then click Run again.

Inspect ①

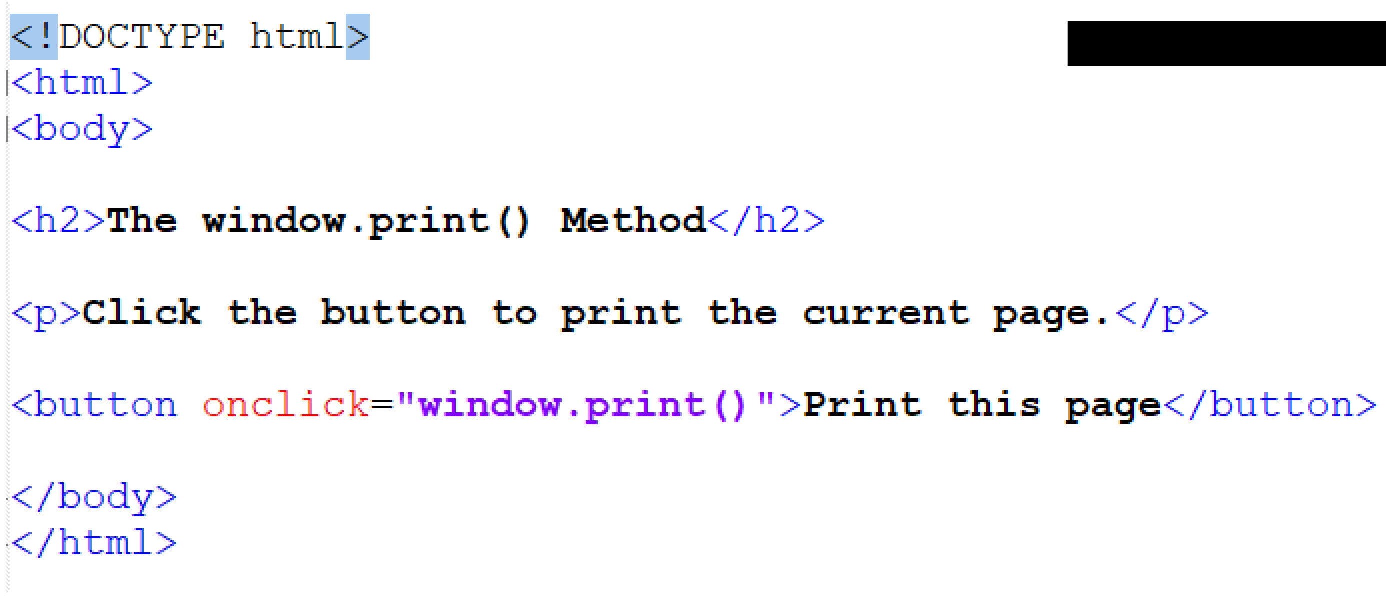
②



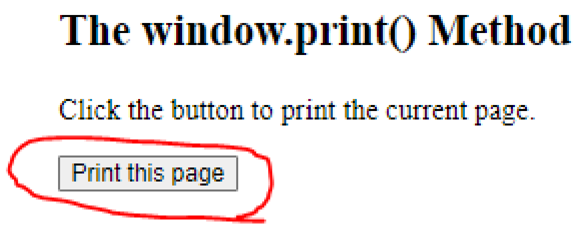
```
11
```

JavaScript Coding Examples-Displaying Outputs/Data- JS Print

- JavaScript does not have any print object or print methods.
- You cannot access output devices from JavaScript.
- The only exception is that you can call the window.print() method in the browser to print the content of the current window.

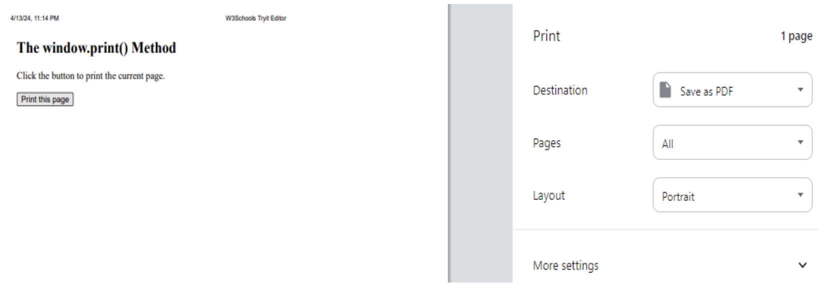


```
<!DOCTYPE html>
<html>
<body>

<h2>The window.print() Method</h2>

<p>Click the button to print the current page.</p>

<button onclick="window.print()">Print this page</button>

</body>
</html>
```



The window.print() Method

Click the button to print the current page.



4/13/24, 11:14 PM W3Schools Tryit Editor

The window.print() Method

Click the button to print the current page.

Print 1 page

Destination: Save as PDF

Pages: All

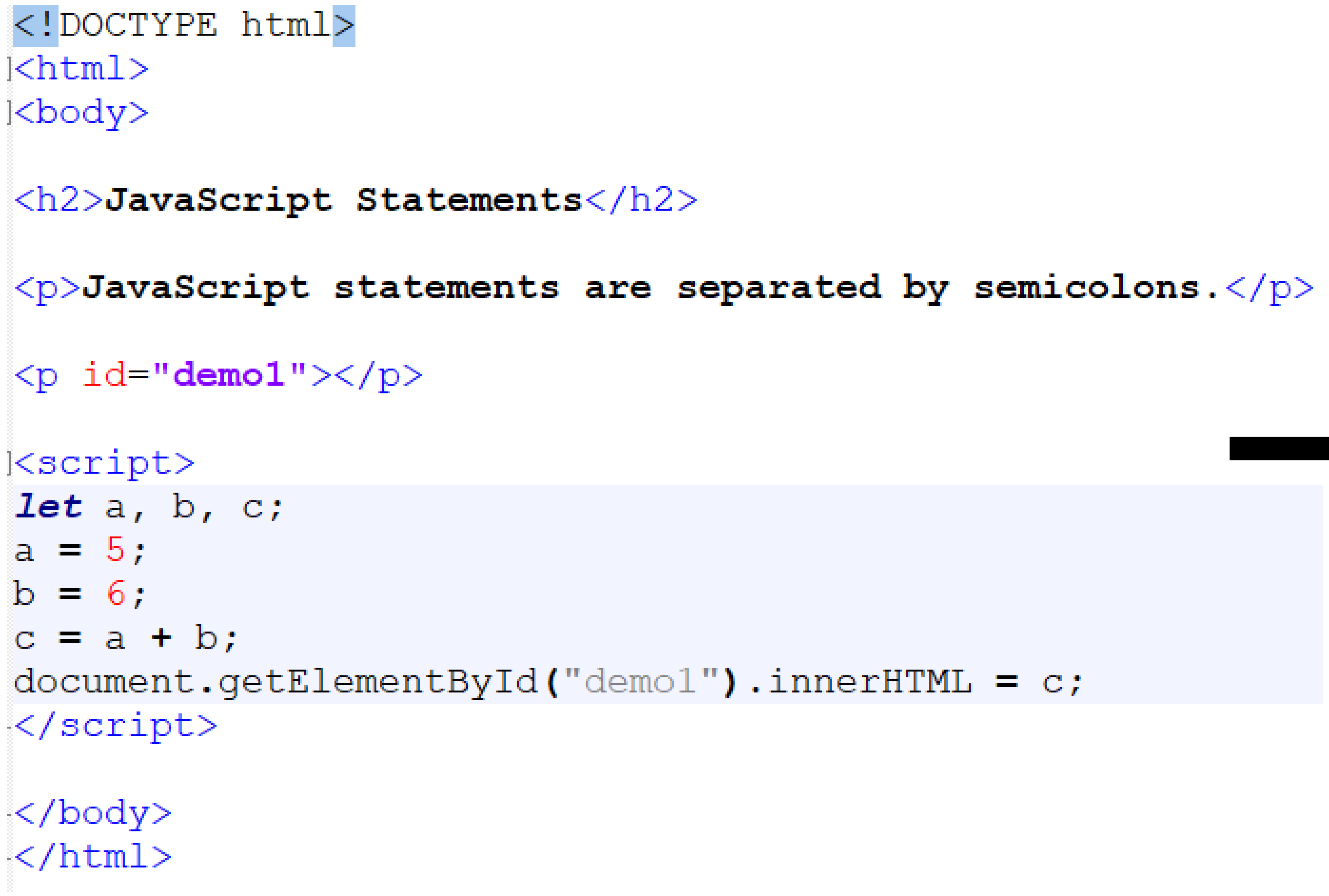
Layout: Portrait

JavaScript Coding Examples- Displaying output

- ✓ Writing into an HTML element, using innerHTML.
- ✓ Writing into the HTML output using document.write().
- ✓ Writing into an alert box, using window.alert().
- ✓ Writing into the browser console, using console.log().
- ✓ You can call the window.print() method in the browser to print the content of the current window.

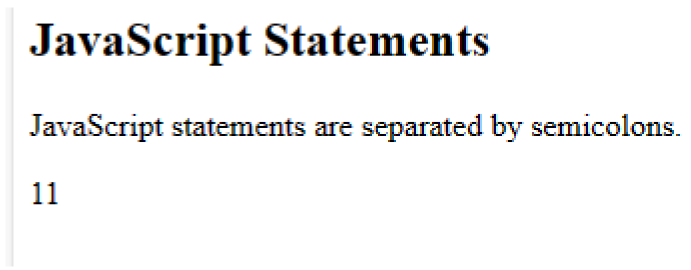
JavaScript Coding Examples- Simple Example

- Semicolons separate JavaScript statements.
- Add a semicolon at the end of each executable statement:



```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Statements</h2>

<p>JavaScript statements are separated by semicolons.</p>

<p id="demo1"></p>

<script>
let a, b, c;
a = 5;
b = 6;
c = a + b;
document.getElementById("demo1").innerHTML = c;
</script>

</body>
</html>
```

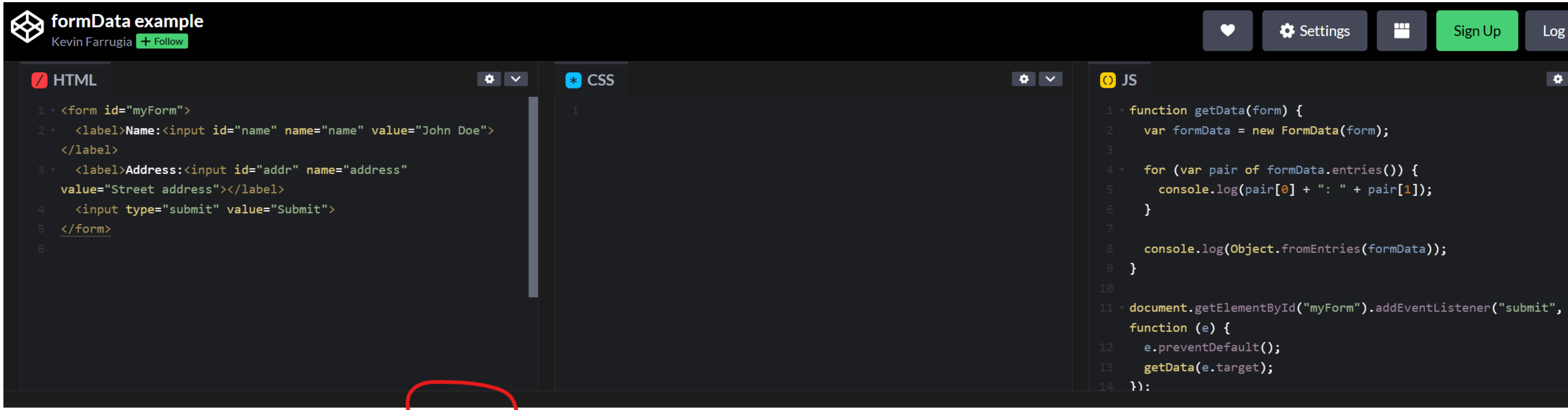
Output of the code



```
JavaScript Statements

JavaScript statements are separated by semicolons.

11
```

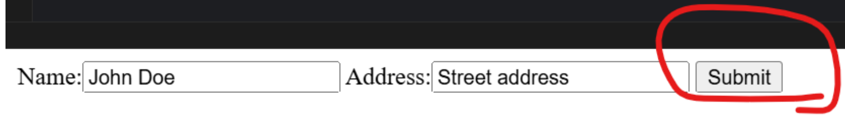
Use of JavaScript- An Example

- Getting form data

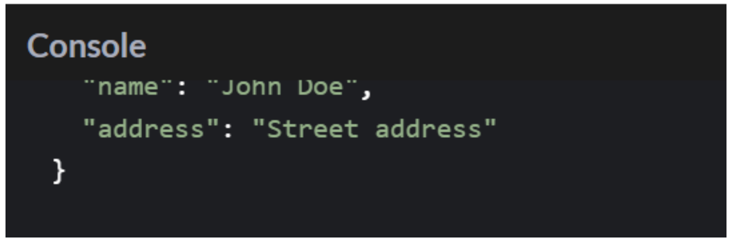


```html
<form id="myForm">
  <label>Name:<input id="name" name="name" value="John Doe">
</label>
  <label>Address:<input id="addr" name="address"
value="Street address"></label>
  <input type="submit" value="Submit">
</form>
```

```javascript
function getData(form) {
  var formData = new FormData(form);

  for (var pair of formData.entries()) {
    console.log(pair[0] + ": " + pair[1]);
  }

  console.log(Object.fromEntries(formData));
}

document.getElementById("myForm").addEventListener("submit",
function (e) {
  e.preventDefault();
  getData(e.target);
});
```



Output



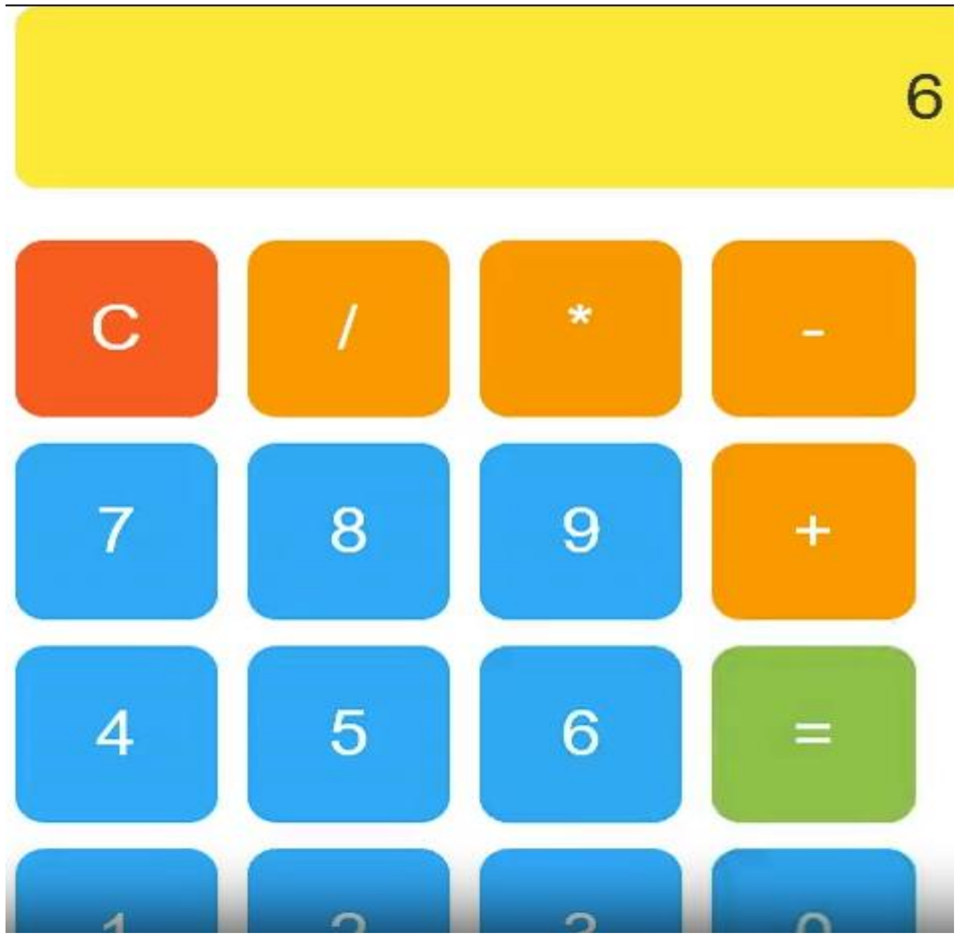
```
Console
  "name": "John Doe",
  "address": "Street address"
}
```

https://codepen.io/kevinfarrugia/pen/Wommgd?editors=1111

JavaScript in Action-Development of the Calculator APP

- Please try the calculator example available at the link below on your own systems.

https://www.geeksforgeeks.org/javascript-calculator/



Summary of the Today's Lesson

Part I

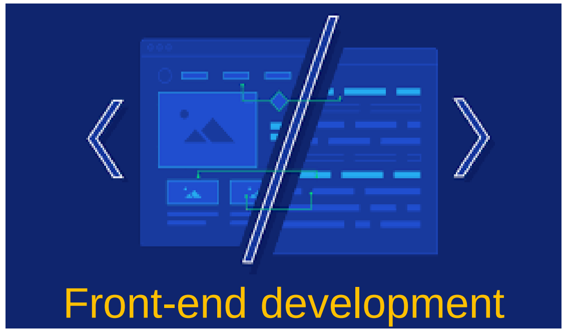
Working with forms

- ❑ Simple forms
- ❑ Elements of forms
- ❑ Input element
- ❑ Etc.

Part II

Working with JS Basics

- ❑ External
- ❑ Internal
- ❑ Inline



Part III

Displaying JS Outputs

- ❑ Window alert
- ❑ Inner HTML
- ❑ Console.log
- ❑ Print
- ❑ Alert

Part IV

JS in practice

- ❑ Calculator development

Note: Please execute all codes at least once on your computers.