

# Designing web pages

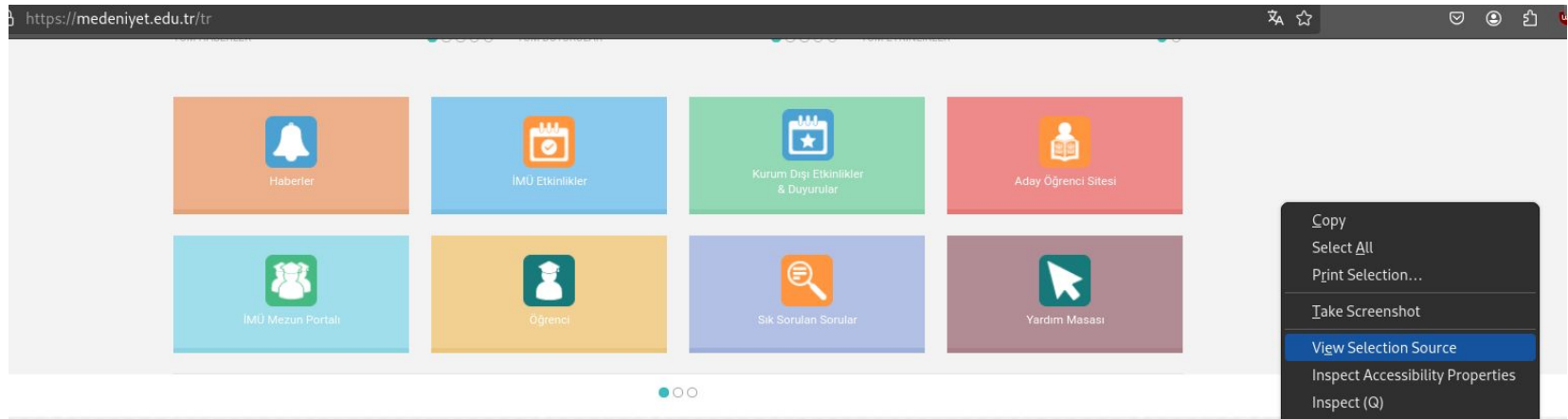
Html

Css

Javascript

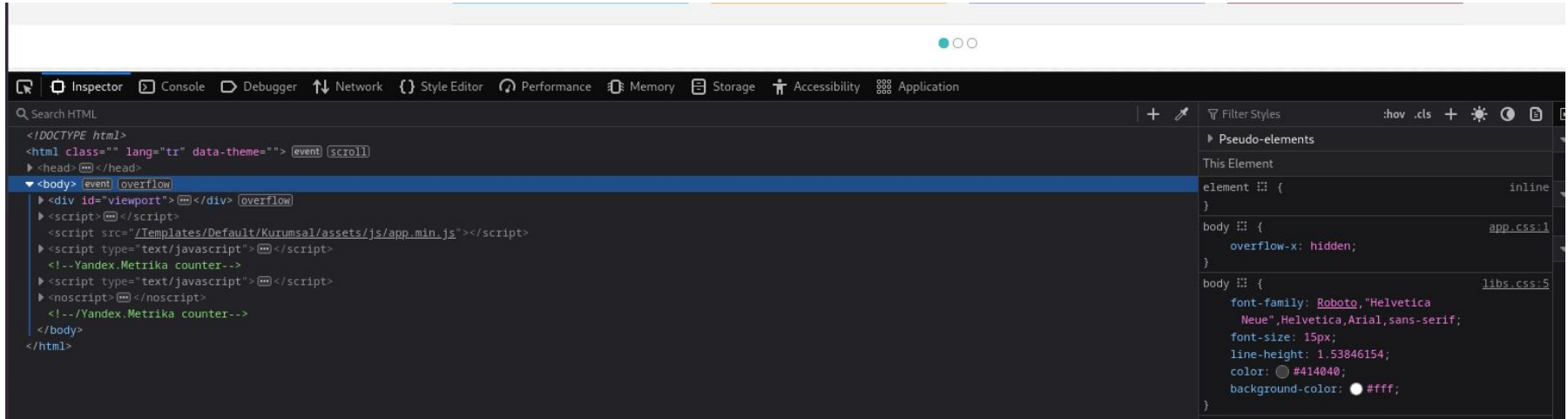
# A webpage

[www.medeniyet.edu.tr](https://www.medeniyet.edu.tr)



Or ctrl+shift+i

# Source of the webpage



The image shows a browser's developer tools interface. The top bar includes icons for Inspector, Console, Debugger, Network, Style Editor, Performance, Memory, Storage, Accessibility, and Application. Below this is a search bar for HTML. The main area displays the HTML source code, with the `<body>` element selected. The right-hand pane shows the 'Filter Styles' section, displaying the styles for the selected element. The styles include 'overflow-x: hidden;' from 'app.css:1' and a font stack from 'libs.css:5'.

```
<!DOCTYPE html>
<html class="" lang="tr" data-theme=""> event | scroll |
  <head>
  <body> event | overflow |
    <div id="viewport">
    <script>
    <script src="/Templates/Default/Kurumsal/assets/js/app_min.js"></script>
    <script type="text/javascript">
    <!--Yandex.Metrika counter-->
    <script type="text/javascript">
    <noscript>
    <!--/Yandex.Metrika counter-->
  </body>
</html>
```

Filter Styles: .hov .cls + [Sun] [Moon] [Book] [Close]

Pseudo-elements

This Element

```
element :: { inline
}
body :: { app.css:1
  overflow-x: hidden;
}
body :: { libs.css:5
  font-family: Roboto, "Helvetica
    Neue", Helvetica, Arial, sans-serif;
  font-size: 15px;
  line-height: 1.53846154;
  color: #414040;
  background-color: #fff;
}
```

# html

The standard **markup language** for Web pages

Text is formatted by using **HTML elements**

- **<tagname>** Content goes here... **</tagname>**
  - from start tag to end tag defines an HTML element

# html

- The standard **markup language** for Web pages

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

The `<!DOCTYPE html>` declaration defines that this document is an HTML5 document

The `<html>` element is the root element of an HTML page

The `<head>` element contains meta information about the HTML page

The `<title>` element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)

The `<body>` element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.

The `<h1>` element defines a large heading

The `<p>` element defines a paragraph

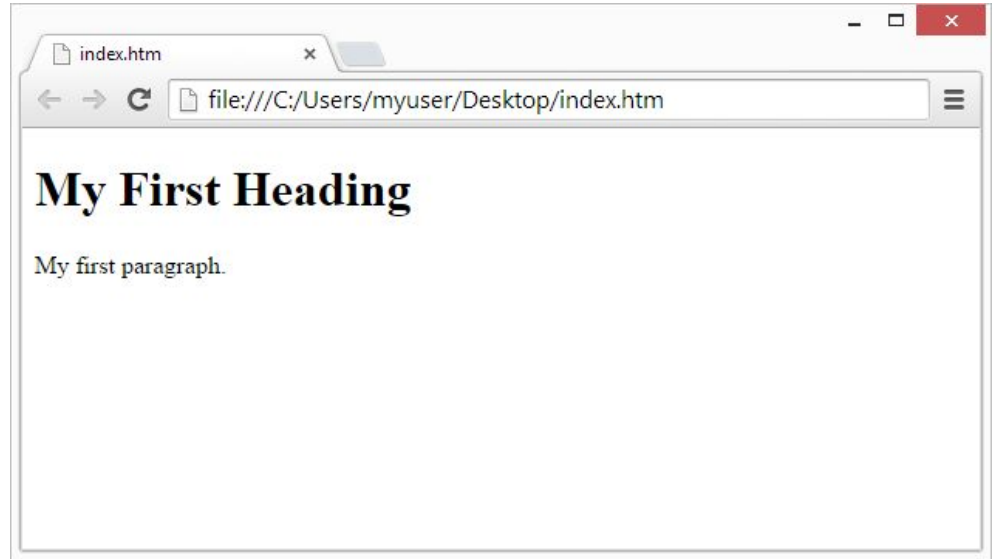
# Web browsers

a web browser (Chrome, Edge, Firefox, Safari) reads HTML documents and display them correctly.

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```



# Html formatting elements

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p>This text is normal.</p>
```

```
<p><b> - Bold text </b></p>
```

```
<p><strong> - Important text </strong></p>
```

```
<p><i> - Italic text </i></p>
```

```
<p><em> - Emphasized text</em></p>
```

```
<p><mark> - Marked text </mark></p>
```

```
<p><small> - Smaller text </small></p>
```

```
<p><del> - Deleted text </del></p>
```

```
<p><ins> - Inserted text </ins></p>
```

```
<p>- A<sub> 200 </sub></p>
```

```
<p>- B <sup> 200</sup></p>
```

```
</body>
```

```
</html>
```

This text is normal.

- **Bold text**

- **Important text**

- *Italic text*

- *Emphasized text*

- **Marked text**

- Smaller text

- ~~Deleted text~~

- Inserted text

- A<sub>200</sub>

- B<sup>200</sup>

# HTML attributes

Attributes usually come in name/value pairs like: **name="value"**

```
<p style="color:red;">This is a red paragraph.</p>
```

```

```

```

```

```
<a href="https://www.w3schools.com/html/">Visit our HTML tutorial</a>
```

- All HTML elements can have attributes
- Attributes provide additional information about elements
- Attributes are always specified in the start tag



# HTML styles

`<tagname style="property:value;">`

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1 style="text-align: center; font-size: 300%; color: yellow; background-color: blue;">Centered Heading</h1>
```

```
<p style="text-align: center; font-size: 100%; color: white; background-color: red;">Centered paragraph.</p>
```

```
</body>
```

```
</html>
```

**Centered Heading**

Centered paragraph.

[https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp)

# Html styles: CSS

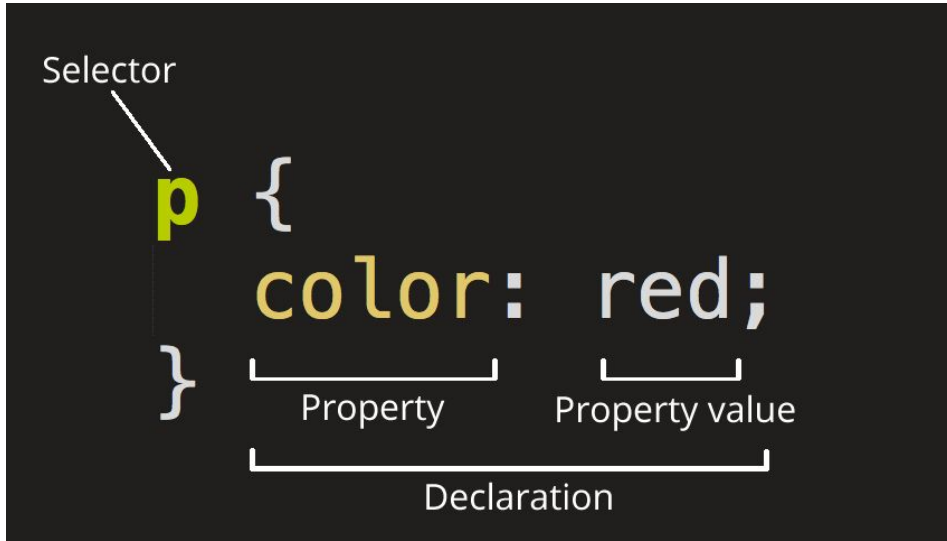
Cascading Style Sheets (CSS) is used to format the layout of a webpage.

CSS can be added to HTML documents in 3 ways:

- **Inline** - by using the `style` attribute inside HTML elements

```
<h1 style="color:blue;">A Blue Heading</h1>
```

# Html styles: CSS



- **Internal** - by using a `<style>` element in the `<head>` section

```
!DOCTYPE html>
<html>
<head>
<style>
  body {background-color: powderblue;}
  h1 {color: blue;}
  p {
    color: red;
    width: 500px;
    border: 1px solid black;
  }
</style>
</head>
<body>
```

```
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
```

```
</body>
</html>
```

[https://developer.mozilla.org/en-US/docs/Learn/Getting\\_started\\_with\\_the\\_web/CSS\\_basics](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics)

[https://www.w3schools.com/html/html\\_css.asp](https://www.w3schools.com/html/html_css.asp)

## "styles.css":

```
body {  
  background-color: powderblue;  
}  
h1 {  
  color: blue;  
}  
p {  
  color: red;  
}  
li,  
h2,  
h3 {  
  color: green;  
}
```

- **External** - by using a `<link>` element to link to an external CSS file

To use an external style sheet, add a link to it in the `<head>` section of each HTML page

```
<!DOCTYPE html>  
<html>  
<head>  
  <link rel="stylesheet" href="styles.css">  
</head>  
<body>
```

```
<h1>This is a heading</h1>
```

```
<p>This is a paragraph.</p>
```

```
</body>
```

```
</html>
```

# CSS example

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  color: blue;
  font-family: verdana;
  font-size: 300%;
}
p {
  color: red;
  font-family: courier;
  font-size: 160%;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

[https://www.w3schools.com/html/html\\_css.asp](https://www.w3schools.com/html/html_css.asp)

# CSS: selecting subset of elements

Class attribute is the list of classes of the element

HTML

```
<ul>
  <li>Item one</li>
  <li class="special">Item two</li>
  <li>Item <em>three</em></li>
</ul>
```

CSS

```
.special {
  color: orange;
  font-weight: bold;
}
```

CSS

```
li.special,
span.special {
  color: orange;
  font-weight: bold;
}
```

[https://developer.mozilla.org/en-US/docs/Learn/CSS/First\\_steps/Getting\\_started](https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/Getting_started)

# CSS: styling based on location

## HTML

```
<h1>I am a level one heading</h1>

<p>
  This is a paragraph of text.
  In the text is a <span>span element</span> and
  also a <a href="http://example.com">link</a>.
</p>
```

## CSS

```
li em {
  color: rebeccapurple;
}

h1 + p {
  font-size: 200%;
}
```

[https://developer.mozilla.org/en-US/docs/Learn/CSS/First\\_steps/Getting\\_started](https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/Getting_started)

# CSS: styling based on state

```
CSS

a:link {
  color: pink;
}

a:visited {
  color: green;
}

a:hover {
  text-decoration: none;
}
```

```
HTML

<h1>I am a level one heading</h1>

<p>
  This is a paragraph of text. In the text is a <span>span element</span> and
  also a <a href="http://example.com">link</a>.
</p>

<p>This is the second paragraph. It contains an <em>emphasized</em> element.</p>

<ul>
  <li>Item one</li>
  <li>Item two</li>
  <li>Item <em>three</em></li>
</ul>
```



# CSS: combining all

CSS

```
body h1 + p .special {  
  color: yellow;  
  background-color: black;  
  padding: 5px;  
}
```

This will style any element with a class of `special`, which is inside a `<p>`, which comes just after an `<h1>`, which is inside a `<body>`. Phew!

[https://developer.mozilla.org/en-US/docs/Learn/CSS/First\\_steps/How\\_CSS\\_is\\_structured](https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/How_CSS_is_structured)

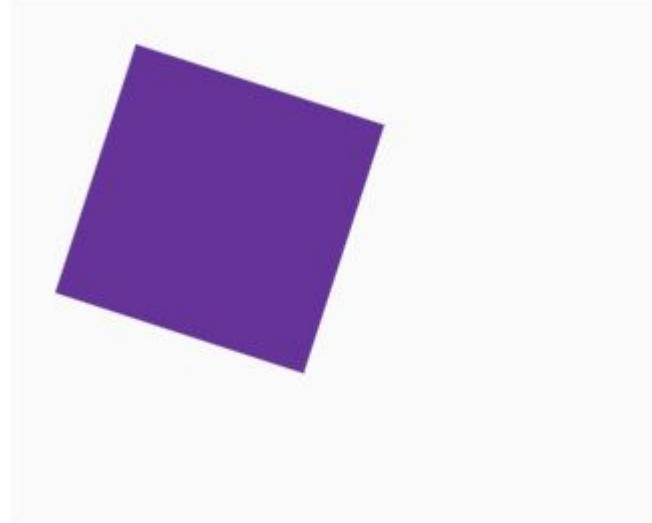
# CSS: functions

## HTML

```
<div class="box"></div>
```

## CSS

```
.box {  
  margin: 30px;  
  width: 100px;  
  height: 100px;  
  background-color: rebeccapurple;  
  transform: rotate(0.8turn);  
}
```



<https://developer.mozilla.org/en-US/docs/Web/CSS/transform>  
[https://developer.mozilla.org/en-US/docs/Learn/CSS/First\\_steps/How\\_CSS\\_is\\_structured](https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/How_CSS_is_structured)

# CSS:

## @rules (at rules)

[CSS statements](#) that instruct CSS how to behave

```
@import "styles2.css";
```

CSS

```
/* General structure */  
@identifier (RULE);  
  
/* Example: tells browser to use UTF-8 character set */  
@charset "utf-8";
```

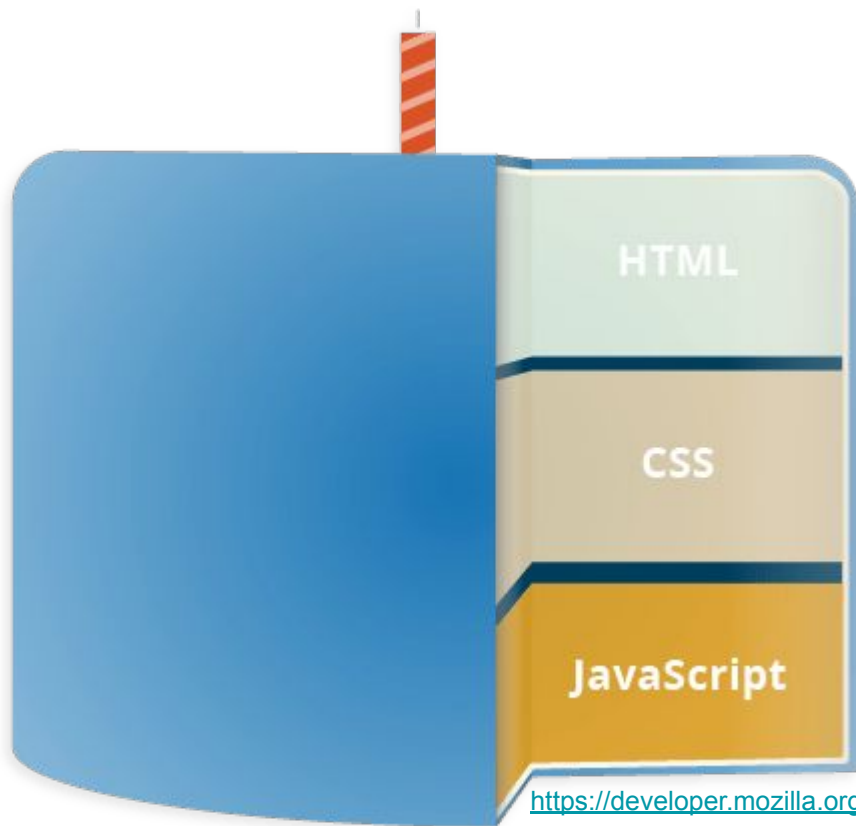
CSS

```
@identifier (RULE) {  
}
```

```
@media (min-width: 70em) {  
  /* Increase the global font size on larger screens or windows  
  for better readability */  
  body {  
    font-size: 130%;  
  }  
}
```

<https://developer.mozilla.org/en-US/docs/Web/CSS/At-rule>  
[https://developer.mozilla.org/en-US/docs/Learn/CSS/First\\_steps/How\\_CSS\\_is\\_structured](https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/How_CSS_is_structured)

# Javascript

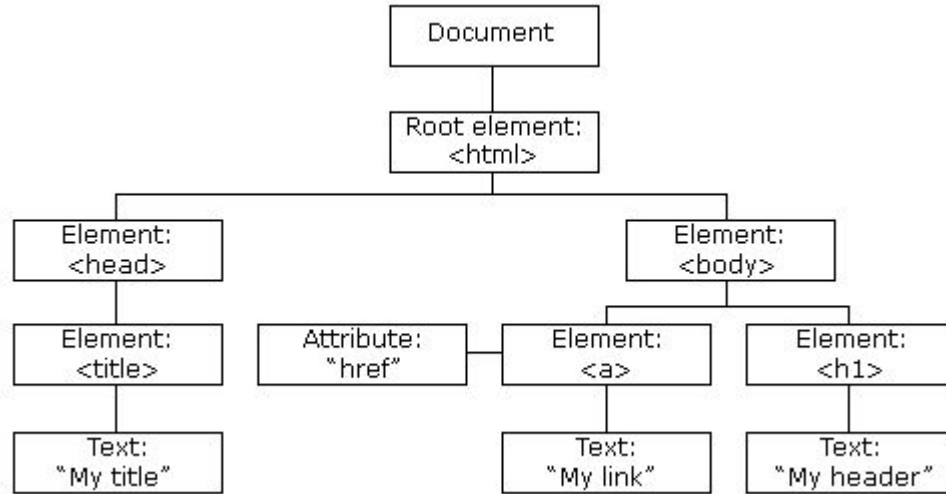


JavaScript (or "JS") is a scripting or programming language that allows you to implement complex features on web pages

- Used most often for dynamic client-side scripts on webpages,
- but it is also often used on the [server](#)-side,
  - using a runtime such as [Node.js](#), [Deno](#), and [Bun](#)
- How to use Javascript with HTML?

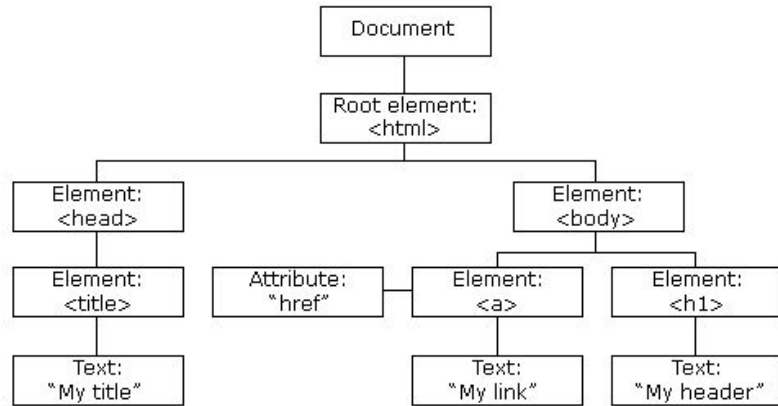
# The HTML DOM (Document Object Model)

When a web page is loaded, the browser creates a Document Object Model of the page.



# The HTML DOM (Document Object Model)

The HTML DOM is a standard for how to get, change, add, or delete HTML elements.



With the object model, JavaScript gets all the power it needs to create dynamic HTML:

With Javascript you can

- change all the HTML elements in the page
- change all the HTML attributes in the page
- change all the CSS styles in the page
- remove existing HTML elements and attributes
- add new HTML elements and attributes
- react to all existing HTML events in the page
- create new HTML events in the page

# The DOM Programming Interface

In the DOM, all HTML elements are defined as **objects**.

HTML DOM methods are **actions** you can perform (on HTML Elements).

HTML DOM properties are **values** (of HTML Elements) that you can set or change.

The HTML DOM can be accessed with JavaScript (and with other programming languages).

The programming interface is the properties and methods of each object.

- A **property** is a value that you can get or set
  - like changing the content of an HTML element.
- A **method** is an action you can do
  - like add or deleting an HTML element).

[https://www.w3schools.com/js/js\\_htmlDOM\\_methods.asp](https://www.w3schools.com/js/js_htmlDOM_methods.asp)

# Example

```
<html>
<body>

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

<script>
document.getElementById("demo").innerHTML = "Hello World!";
</script>

</body>
</html>
```

- **getElementById** is a **method**
  - The most common way to access an HTML element is to use the `id` of the element.
  - In the example above the `getElementById` method used `id="demo"` to find the element.
- **innerHTML** is a **property**.
  - The `innerHTML` property is useful for getting or replacing the content of HTML elements.
  - The `innerHTML` property can be used to get or change any HTML element, including `<html>` and `<body>`

[https://www.w3schools.com/js/js\\_html\\_dom\\_methods.asp](https://www.w3schools.com/js/js_html_dom_methods.asp)



# How to add JS

internal

External

Inline

HTML

```
<script>  
  // JavaScript goes here  
</script>
```

HTML

```
<script type="module" src="script.js"></script>
```

HTML

```
<button onclick="createParagraph()">Click me!</button>
```

# Finding HTML Elements from JS

## Method

## Example

**`document.getElementById(id)`**

```
const element = document.getElementById("intro");
```

**`document.getElementsByTagName(name)`**

```
const element = document.getElementsByTagName("p");
```

```
const x = document.getElementById("main");
```

```
const y = x.getElementsByTagName("p");
```

**`document.getElementsByClassName(name)`**

```
const x = document.getElementsByClassName("intro");
```

**`document.querySelectorAll(name)`**

```
//return a list of all <p> elements with class="intro"
```

```
const x = document.querySelectorAll("p.intro");
```

[https://www.w3schools.com/js/js\\_html\\_dom\\_document.asp](https://www.w3schools.com/js/js_html_dom_document.asp)

# Changing HTML elements

## Property

## Description

*`element.innerHTML = new html content`*

Change the inner HTML of an element

*`element.attribute = new value`*

Change the attribute value of an HTML element

*`element.style.property = new style`*

Change the style of an HTML element

## Method

## Description

*`element.setAttribute(attribute, value)`*

Change the attribute value of an HTML element

[https://www.w3schools.com/js/js\\_htmlDOM\\_document.asp](https://www.w3schools.com/js/js_htmlDOM_document.asp)

# DOM Events

## Reacting to events

**onclick=JavaScript**

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

<h1 onclick="this.innerHTML = 'Oops!'">Click on this text!</h1>

</body>
</html>
```

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

<h1 onclick="changeText(this)">Click on this text!</h1>

<script>
function changeText(id) {
  id.innerHTML = "Oops!";
}
</script>

</body>
</html>
```

[https://www.w3schools.com/js/js\\_html\\_dom\\_events.asp](https://www.w3schools.com/js/js_html_dom_events.asp)

[https://www.w3schools.com/js/js\\_events.asp](https://www.w3schools.com/js/js_events.asp)

See events for each HTML element

[https://www.w3schools.com/tags/ref\\_eventattributes.asp](https://www.w3schools.com/tags/ref_eventattributes.asp)

[https://www.w3schools.com/tags/ref\\_attributes.asp](https://www.w3schools.com/tags/ref_attributes.asp)

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<button onclick="displayDate()">The time is?</button>
```

```
<script>
```

```
function displayDate() {
```

```
    document.getElementById("demo").innerHTML=Date();
```

```
}
```

```
</script>
```

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

```
</body>
```

```
</html>
```

[https://www.w3schools.com/js/js\\_htmlDOM\\_events.asp](https://www.w3schools.com/js/js_htmlDOM_events.asp)

# Assigning events using HTML DOM

```
<script>  
document.getElementById("myBtn").onclick = displayDate;  
</script>
```

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript HTML Events</h1>  
<h2>The onclick Events</h2>  
  
<p>Click "Try it" to execute the displayDate() function.</p>  
<button id="myBtn">Try it</button>  
  
<p id="demo"></p>  
  
<script>  
document.getElementById("myBtn").onclick = displayDate;  
function displayDate() {  
    document.getElementById("demo").innerHTML = Date();  
}  
</script>  
</body>  
</html>
```

# Event listeners

When an event occurs, it is executed!

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript addEventListener()</h2>
<p>This example uses the addEventListener() method to attach a click event to a
button.</p>

<button id="myBtn">Try it</button>

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

<script>
document.getElementById("myBtn").addEventListener("click", displayDate);

function displayDate() {
  document.getElementById("demo").innerHTML = Date();
}
</script>

</body>
</html>
```

[https://www.w3schools.com/js/js\\_htmlDOM\\_eventlistener.asp](https://www.w3schools.com/js/js_htmlDOM_eventlistener.asp)

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript addEventListener()</h2>
<p>This example uses the addEventListener() method to attach a click event to a button.</p>
<button id="myBtn">Try it</button>
<p id="demo"></p>

<script>
document.getElementById("myBtn").addEventListener("click", displayDate);
document.getElementById("myBtn").addEventListener("click", changeColor);

function displayDate() {
    this.innerHTML = Date();
    document.getElementById("demo").innerHTML = Date();
}
function changeColor() {
    this.style = "color:red;background-color:white;";
}
</script>
</body>
</html>
```

[https://www.w3schools.com/js/js\\_htmlDOM\\_eventlistener.asp](https://www.w3schools.com/js/js_htmlDOM_eventlistener.asp)