

# JavaScript DOM

DOM

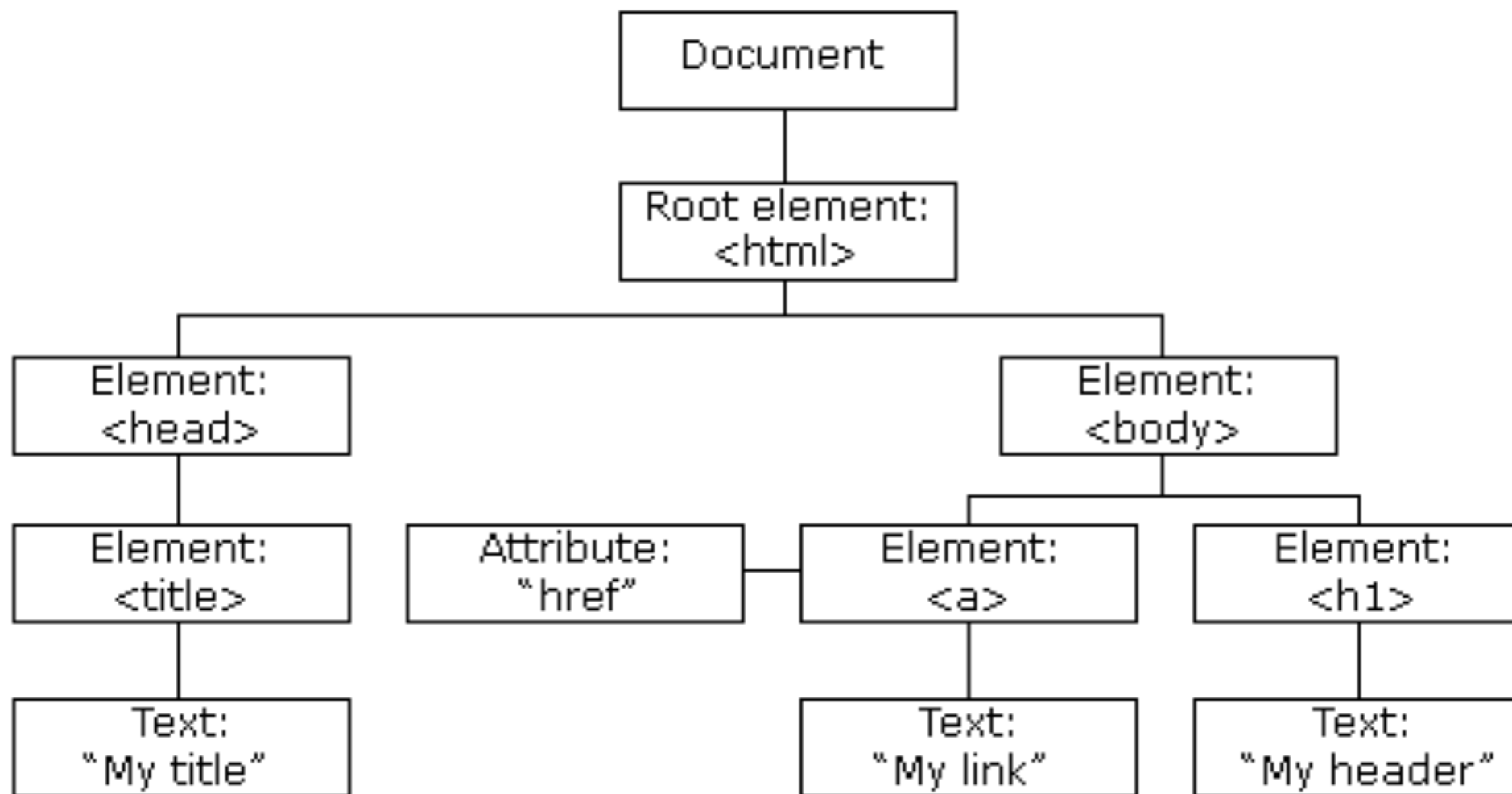
HTML Elements

CSS

Events

# Definition - DOM

## The HTML DOM Tree of Objects



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

# Definition - DOM

From the HTML above you can read:

- `<html>` is the root node
- `<html>` has no parents
- `<html>` is the parent of `<head>` and `<body>`
- `<head>` is the first child of `<html>`
- `<body>` is the last child of `<html>`

and:

- `<head>` has one child: `<title>`
- `<title>` has one child (a text node): "DOM Tutorial"
- `<body>` has two children: `<h1>` and `<p>`
- `<h1>` has one child: "DOM Lesson one"
- `<p>` has one child: "Hello world!"
- `<h1>` and `<p>` are siblings

```
<html>

  <head>
    <title>DOM Tutorial</title>
  </head>

  <body>
    <h1>DOM Lesson one</h1>
    <p>Hello world!</p>
  </body>

</html>
```

# DOM Programming Interface

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

# HTML Elements

## Finding HTML Elements

Method	Description
<code>document.getElementById(<i>id</i>)</code>	Find an element by element id
<code>document.getElementsByTagName(<i>name</i>)</code>	Find elements by tag name
<code>document.getElementsByClassName(<i>name</i>)</code>	Find elements by class name

## Changing HTML Elements

Property	Description
<code>element.innerHTML = <i>new html content</i></code>	Change the inner HTML of an element
<code>element.attribute = <i>new value</i></code>	Change the attribute value of an HTML element
<code>element.style.property = <i>new style</i></code>	Change the style of an HTML element
Method	Description
<code>element.setAttribute(<i>attribute</i>, <i>value</i>)</code>	Change the attribute value of an HTML element

# Example - Updating Element

```
<html>
<body>

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

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

</body>
</html>
```

[https://www.w3schools.com/js/tryit.asp?filename=tryjs\\_dom\\_method](https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_method)

# HTML Elements

## Adding and Deleting Elements

Method	Description
<code>document.createElement(<i>element</i>)</code>	Create an HTML element
<code>document.removeChild(<i>element</i>)</code>	Remove an HTML element
<code>document.appendChild(<i>element</i>)</code>	Add an HTML element
<code>document.replaceChild(<i>new</i>, <i>old</i>)</code>	Replace an HTML element
<code>document.write(<i>text</i>)</code>	Write into the HTML output stream

# Example - Adding element

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

<ul id="myList1"><li>Coffee</li><li>Tea</li></ul>
<ul id="myList2"><li>Water</li><li>Milk</li></ul>

<p>Click the button to move an item from one list to another.</p>

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

<script>
function myFunction() {
  var node = document.getElementById("myList2").lastChild;
  document.getElementById("myList1").appendChild(node);
}
</script>

</body>
</html>
```

[https://www.w3schools.com/jsref/tryit.asp?  
filename=tryjsref\\_node\\_appendchild2](https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_node_appendchild2)



# CSS Manipulation

```
document.getElementById(id).style.property = new style
```

```
<html>
```

```
<body>
```

```
<p id="p2">Hello World!</p>
```

```
<script>
```

```
document.getElementById("p2").style.color = "blue";
```

```
</script>
```

```
<p>The paragraph above was changed by a script.</p>
```

```
</body>
```

```
</html>
```

[https://www.w3schools.com/js/tryit.asp?  
filename=tryjs\\_change\\_style](https://www.w3schools.com/js/tryit.asp?filename=tryjs_change_style)

# Events

HTML events are **"things"** that happen to HTML elements.

When JavaScript is used in HTML pages, JavaScript can **"react"** on these events.

## Common HTML Events

Here is a list of some common HTML events:

Event	Description
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

# Event Based Manipulation

```
document.getElementById(id).style.property = new style
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1 id="id1">My Heading 1</h1>
```

```
<button type="button"
```

```
onclick="document.getElementById('id1').style.color = 'red'">
```

```
Click Me!</button>
```

```
</body>
```

```
</html>
```

[https://www.w3schools.com/js/tryit.asp?  
filename=tryjs\\_dom\\_color2](https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_color2)

# Event Listeners

```
element.addEventListener("click", function() { alert("Hello World!"); });
```

```
element.addEventListener("click", myFunction);
```

```
function myFunction() {  
    alert ("Hello World!");  
}
```

[https://www.w3schools.com/js/tryit.asp?filename=tryjs\\_dom\\_color2](https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_color2)