



# Intro To DOM

time to start making some cool stuff

A presentation by Maysa Rouissi



# Agenda



**01** DOM: What?

**02** Accessing the DOM

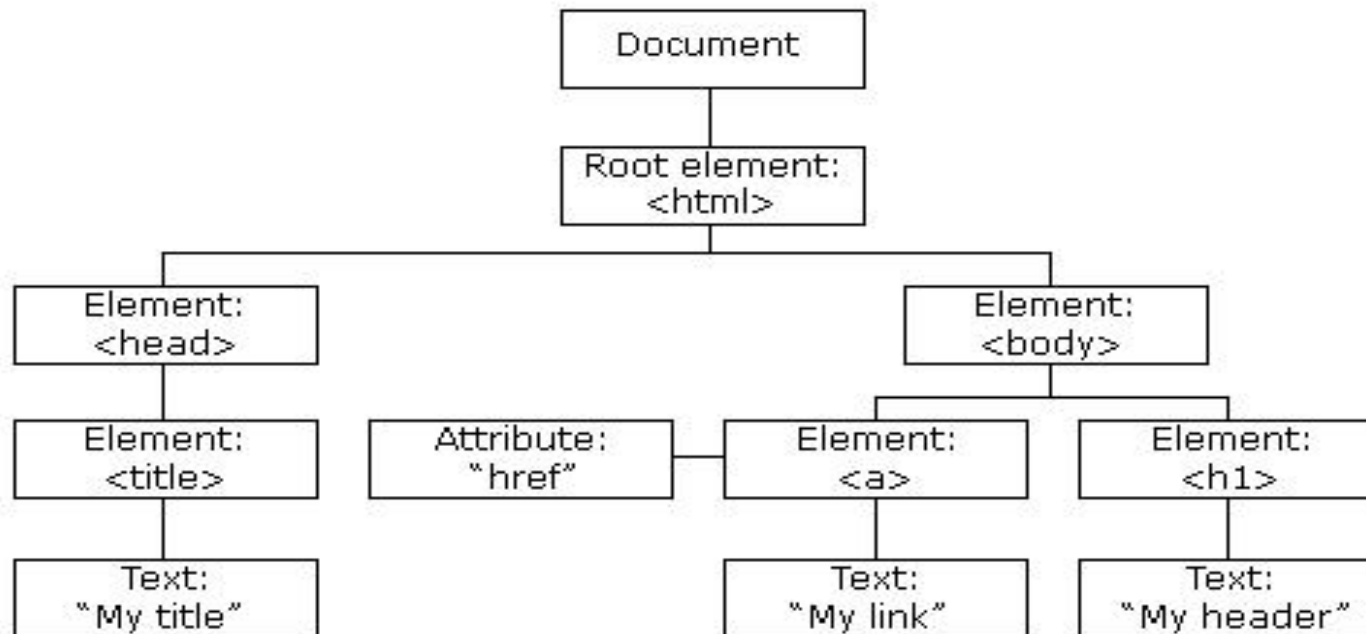
**03** DOM & CSS

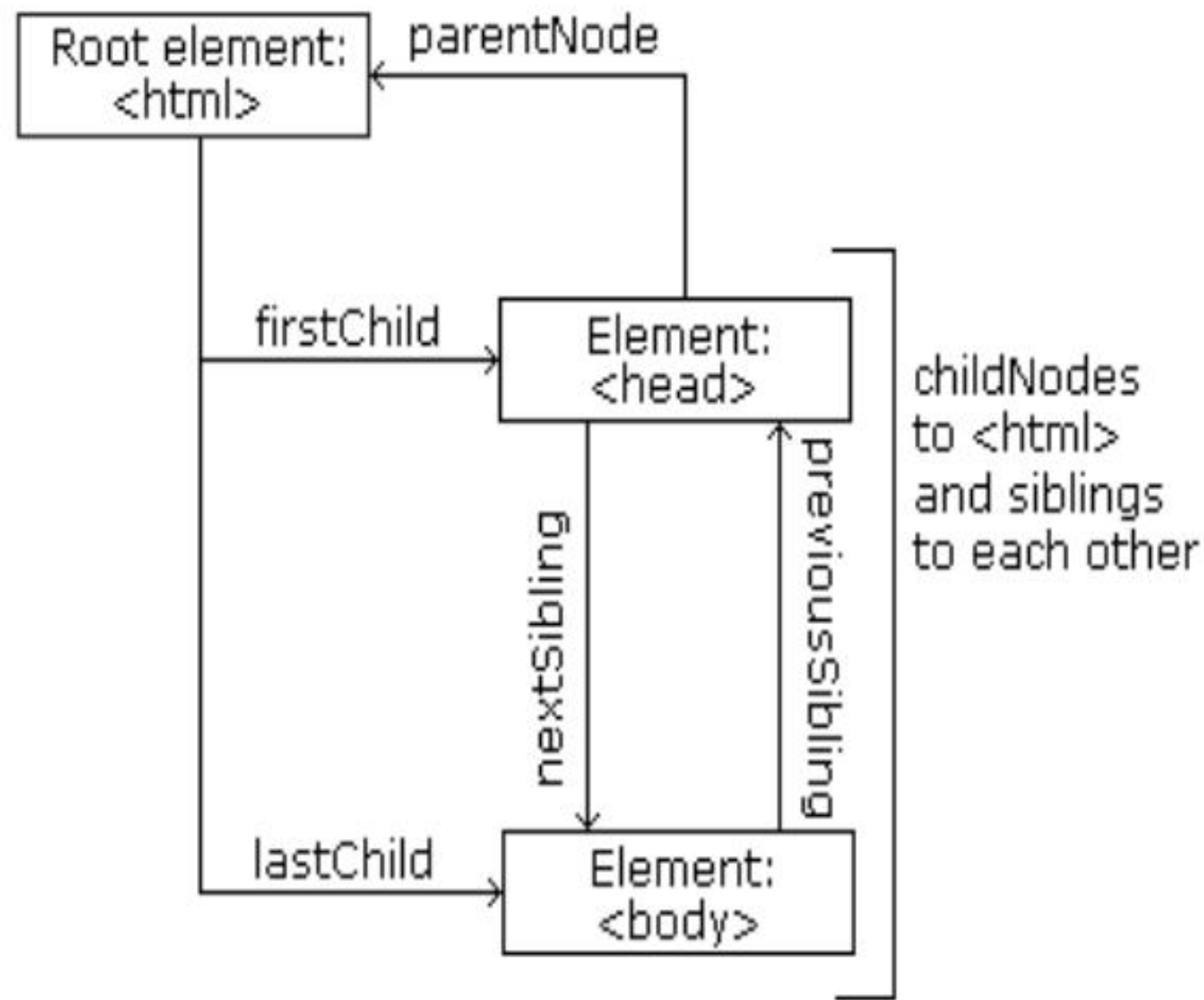
**04** DOM Events

# What is the DOM?

**The Document Object Model** (DOM) is a programming interface for HTML and XML documents. It represents the page so that programs can change the document structure, style, and content.

The HTML DOM model is constructed as a tree of Objects:





# Accessing the DOM

The elements in an HTML document are available directly as document properties.

If you want to access any element in an HTML page, you always start with accessing the document object.

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



**Finding HTML Elements**

<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



**Adding HTML elements**

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



**Changing HTML  
elements**

# Target Elements With Selectors

## `document.querySelector` —

returns the first element within the document that matches the specified or group of selector. If no matches are found null is returned..

## `document.querySelectorAll` —

This represents the list of all the document's elements that match the specified group of selectors.







**D**

**O**

**M**

---

**C**

**S**

**S**

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

```
<script>
```

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

```
</script>
```

# DOM Events



JavaScript code can be executed when an event occurs, like when a user clicks on an HTML element.



**onclick="JavaScript"**

```
<button onclick="displayDate()">Try it</button>
```



The HTML DOM allows you to assign events to HTML elements using JavaScript

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

# DOM EventListener

The `addEventListener()` method attaches an event handler to an element without overwriting existing event handlers.

You can add many event handlers of the same type to one element, i.e two "click" events.

You can easily remove an event listener by using the `removeEventListener()` method.

**Syntax :** `element.addEventListener(event, function);`

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

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



**THIS IS JAVASCRIPT**



Thank You