

# Javascript in the browser

Thierry Sans

# Example

```
<html>
<body>
<script type="text/javascript">
    document.write("<h1>This is a heading</h1>");
    document.write("<p>This is a paragraph.</p>");
</script>
</body>
</html>
```

# Javascript: Inline, embedded or separate file?

## Inline

```
<button onclick="console.log('Hello World!');">Click me</button>
```

## Embedded

```
<script type="text/javascript">  
    console.log("Hello World!");  
</script>
```

## Separate file

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

# Javascript in the browser is restrictive

- ✓ You can access elements of the webpage and the browser
- ✓ You can track user actions on the webpage (events)
- ✓ You can create threads (web workers)
- ✓ You can open sockets (web sockets)
- ✓ ...
- ⦿ You cannot access the file system (only via the upload form)
- ⦿ You cannot access to other programs
- ⦿ You cannot access to other tabs in the browser
- ⦿ ...

# The Browser



# Pop-up Boxes

<code>alert("hello world!")</code>	dialog box with “ok” button
<code>confirm("are you sure?")</code>	dialog box with “ok” and “cancel” buttons
<code>prompt("Name?", "John")</code>	input box with prompt text and default value

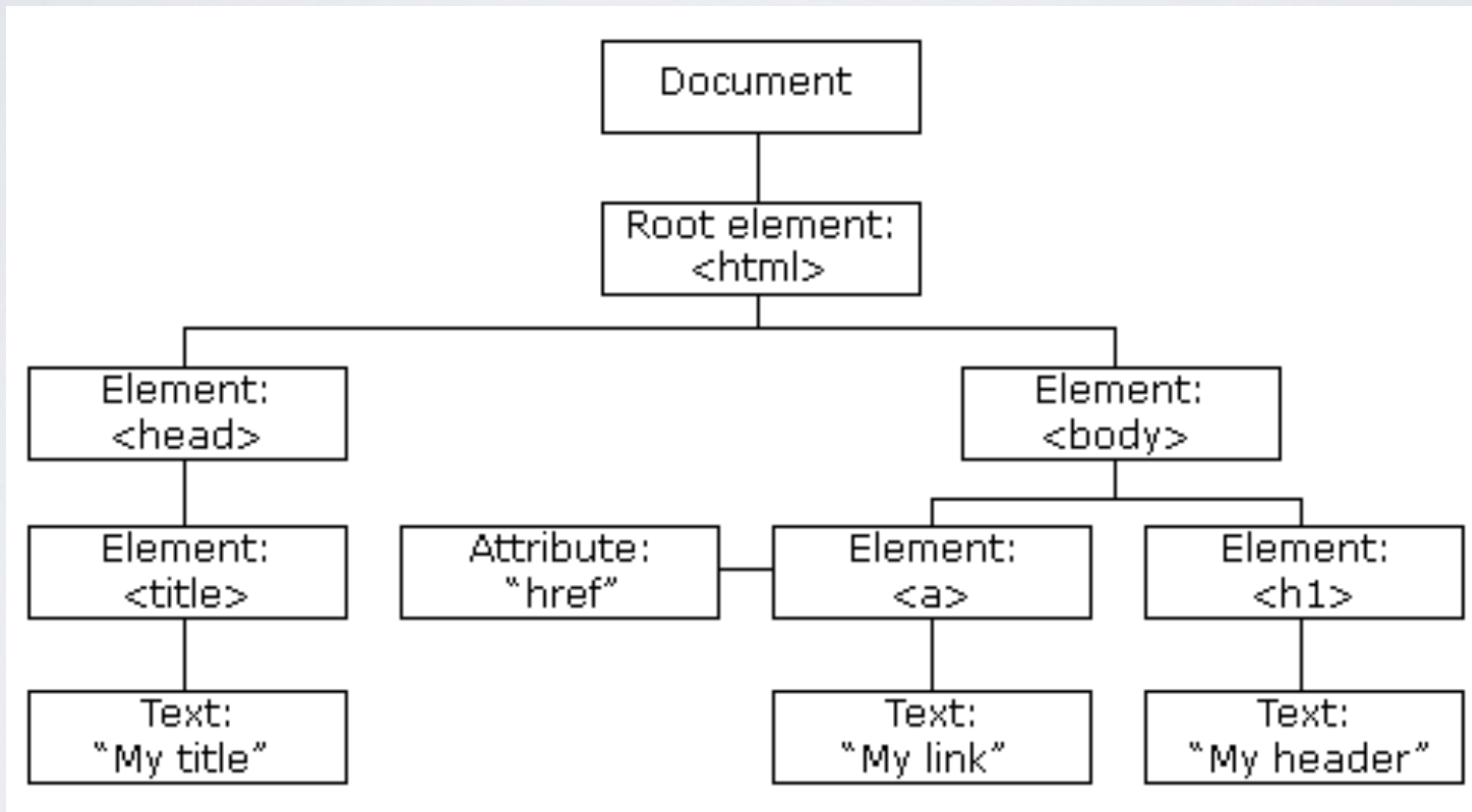
# The Browser

<b>screen</b>	the visitor's screen
<b>browser</b>	the browser itself
<b>window</b>	the current browser window
<b>url</b>	the current url
<b>history</b>	Back and forward URLs

# Document Object Model



# DOM - Document Object Model



# Node accessors

## The root node

`document`

## Accessors

```
document.getElementById("id")  
document.getElementsByTagName("p");  
document.getElementsByClassName("class");  
document.querySelector("#id .class p");  
document.querySelectorAll("#id .class p");
```

# DOM methods

<b>x.innerHTML</b>	the content of x
<b>x.attributes</b>	the attributes nodes of x
<b>x.style</b>	css of x
<b>x.parentNode</b>	the parent node of x
<b>x.children</b>	the child nodes of x
<b>x.appendChild</b>	insert a child node to x
<b>x.removeChild</b>	remove a child node from x
...	...

# Events

# DOM events and handlers

<b><code>e.onload</code></b>	when e is fully loaded
<b><code>e.onclick</code></b>	when e is clicked
<b><code>e.onSubmit</code></b>	when e is submitted
<b><code>e.onhover</code></b>	when the mouse is on top e
<b><code>e.onkeydown</code></b>	when a key is pressed while e is in focus
<b><code>...</code></b>	<b><code>...</code></b>

<https://developer.mozilla.org/en-US/docs/Web/Events>



# User-defined events and listeners

```
// Listen for the event
document.addEventListener( 'onSomething', function(e) {
    ...
});
```

```
// Dispatch the event
document.dispatchEvent(new Event( 'onSomething' ));
```

# Custom events

```
// Listen for the custom event
document.addEventListener('onSomething', function(e) {
    console.log(e.detail);
});
```

```
// Dispatch the custom event
document.dispatchEvent(new CustomEvent('onSomething',
{ e.detail: 'Hello World!' }));
```