

# The DOM and some DIVs



# DOM - It's all about family???

DOM stands for document object model

DOM explains the way an HTML page is structured

While we do not explicitly use DOM it helps for us to understand why the term is important and how it explains our webpage

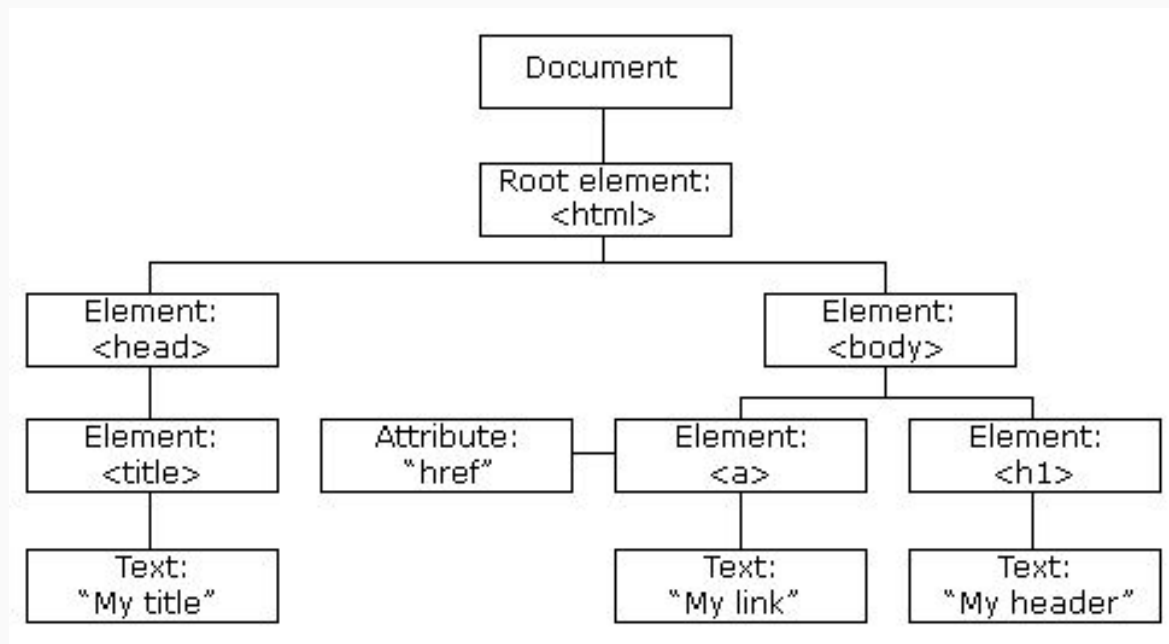
Let's start by reading:

[https://www.w3schools.com/whatis/whatis\\_htmlDOM.asp](https://www.w3schools.com/whatis/whatis_htmlDOM.asp)

- Definitions
- The HTML DOM
- Finding HTML Elements



# DOM Structure

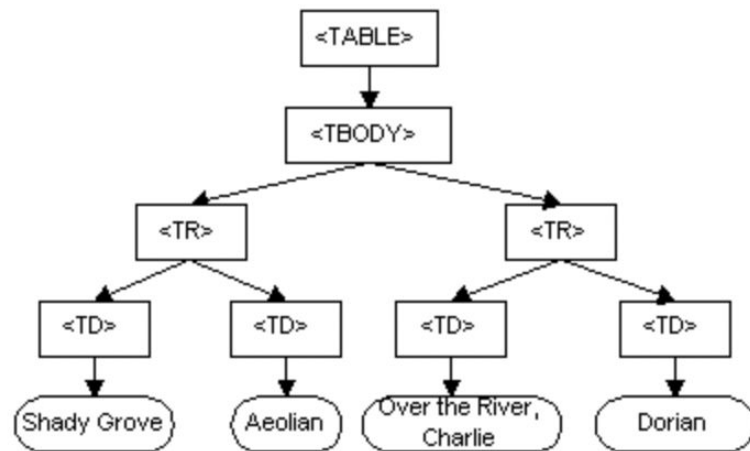


# DOM Explained (again)

## Code

```
<TABLE>
<TBODY>
<TR>
<TD>Shady Grove</TD>
<TD>Aeolian</TD>
</TR>
<TR>
<TD>Over the River, Charlie</TD>
<TD>Dorian</TD>
</TR>
</TBODY>
</TABLE>
```

## DOM



# DOM continued...like 11 times too many

Understanding DOM lets us define how we can manipulate individual elements on a page and how things are built on a page

# Div....really do we need to do this again?

A DIV is an empty container for your page

The benefit is it has no pre defined context or bias

DIVs are widely used across webpages

# Dance DIV Dance!!

A couple of cool things you can do with a div

- Show/Hide the div (can also be done with a javascript timer)
- Create a div in Javascript
- Add things to it (like content and child nodes)
- Make a loading screen
- ...and more!

# Show/Hide

```
<script>
function abraKadabra() {
    var divOnPage = document.getElementById("magic");
    if (divOnPage.style.display == "none") {
        divOnPage.style.display = "block";
    } else {
        divOnPage.style.display = "none";
    }
}
</script>
```

```
<button onclick="abraKadabra()">Click
Me</button>

<div id="magic">
    Hey you can see me now!
</div>
```

# Creating a DIV in Javascript

```
var newDiv = document.createElement('div');  
newDiv.innerHTML = "<text>This is text inside my new DIV<text>";  
document.body.appendChild(newDiv);
```

# Using a DIV as an empty container

Javascript:

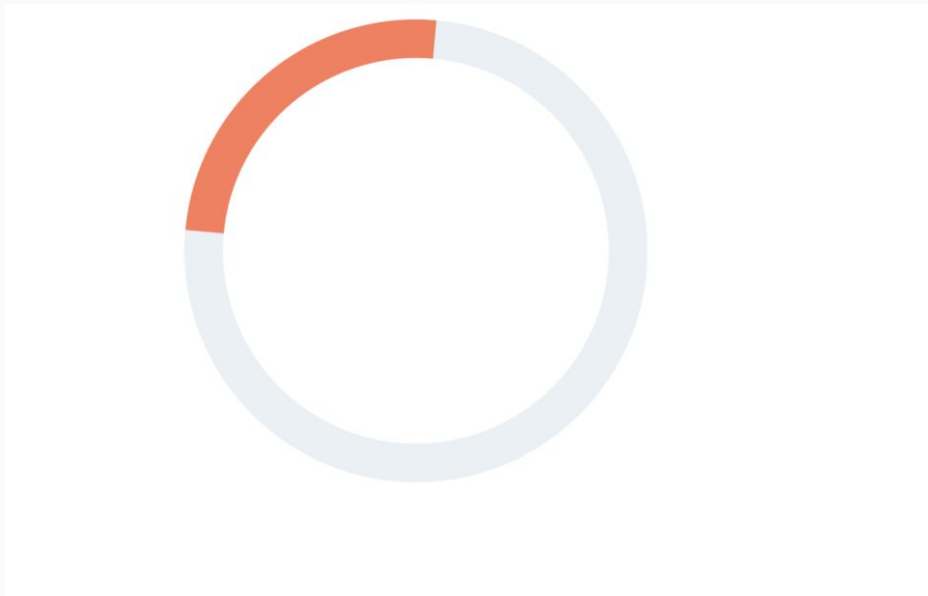
```
document.getElementById("emptyDiv").innerHTML = "<p>Adding text to the  
page</p>";
```

HTML:

```
<div id="emptyDiv"></div>
```

# Loading screen sample

```
<style>
.loader {
  margin: auto;
  border: 20px solid #EAF0F6;
  border-radius: 50%;
  border-top: 20px solid #FF7A59;
  width: 200px;
  height: 200px;
  animation: spinner 4s linear infinite;
}
@keyframes spinner {
  0% { transform: rotate(0deg); }
  100% { transform: rotate(360deg); }
}
</style>
<body>
  <div class="loader"></div>
</body>
```



# Using a timer to show/hide a DIV

```
<div id="loadingAnimation" class="loader"></div>
```

```
<script>
```

```
  setTimeout(() => {
```

```
    document.getElementById('loadingAnimation').style.display = 'none';
```

```
  }, 1000);
```

```
</script>
```

# Homework

Read and complete the exercises in the JS HTML DOM section of W3 Schools:

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

W3schools | Tutorials | Exercises | Get Certified | Services

HTML | CSS | **JAVASCRIPT** | SQL | PYTHON | JAVA | PHP | BOOTSTRAP | HOW TO | W3.CSS | C | C++ | C# | REACT | R | JQUERY | DJANGO | TYPESCRIPT

JS Callbacks  
JS Asynchronous  
JS Promises  
JS Async/Await

**JS HTML DOM**  
DOM Intro  
DOM Methods  
DOM Document  
DOM Elements  
DOM HTML  
DOM Forms  
DOM CSS  
DOM Animations  
DOM Events  
DOM Event Listener  
DOM Navigation  
DOM Nodes  
DOM Collections  
DOM Node Lists

**JavaScript HTML DOM**

← Previous

With the HTML DOM, JavaScript can access and change all the elements of an HTML document.

**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** model is constructed as a tree of **Objects**:

**The HTML DOM Tree of Objects**

```
graph TD
    Document[Document] --> RootElement["Root element: <html>"]
    RootElement --> HeadElement["Element: <head>"]
    RootElement --> BodyElement["Element: <body>"]
```