DOM Manipulation

By Duygu Gözen

1. What is the DOM?
2. Global Objects
3. Node & Elements
4. Querying Elements
5. Exercises
6. Event Listeners
7. Common Listeners
8. Writing Listeners
9. Exercises
10. Q&A

Index & Content

Java Script:

- is a hoisted language
- provides methods and functions to manipulate the html

Browser:

- Renders and Parses the code
- Exposes web API to allow JS to work with the parsed document and that is called the Document Object Model

What is the DOM?

Window:

- The active browser window / tab
- Acts as global storage for script, also provides access to window specific properties and methods (`window.innerWidth`)

Document:

- Root DOM Node
- Provides access to element querying and DOM content (`document.body`)

Global Objects

1. Open a website and go the inspector
2. Type in “window” What do you see?
3. Type in “document” What do you see?
4. Type in console.dir(document) What do you see?
5. Select and element and then type in console.log($0) This gives you the access to the last element you selected.

Example time!

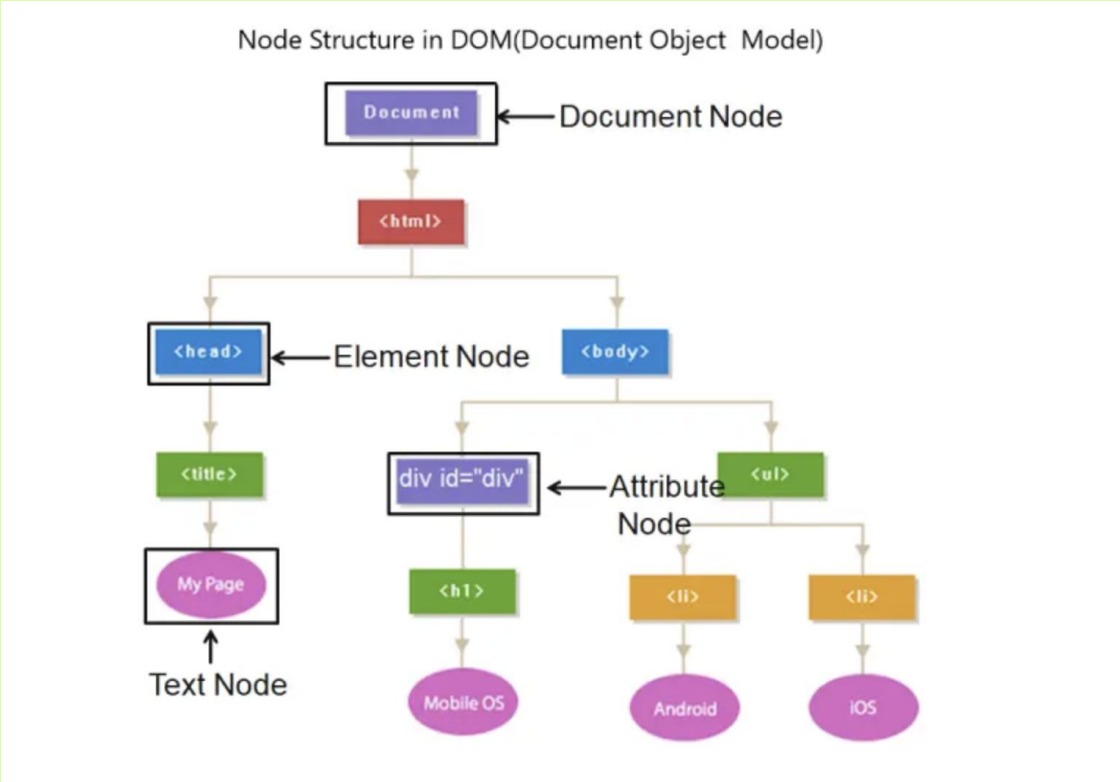
Nodes:

- Objects that make up the DOM
- Most HTML tags are element nodes.
- Text creates Text Nodes
- Attributes create Attribute Nodes

Elements:

- Elements are type of nodes!
- Special properties and methods to interact with the elements
- Available methods and properties depend on the kind of element
- Can be selected in various different ways with JS
- Can be created and removed via JavaScript

Node & Elements


Node Structure in DOM(Document Object Model)
Document
Document Node
<html>
<head>
Element Node
<body>
<title>
div id="div"
Attribute Node

My Page
<h1>

Text Node
Mobile OS
Android
iOS

Querying Elements

querySelector()
getElementbyId()

- Return single elements
- Way of querying elements by CSS Selector or ID
- Direct reference to the object that is returned

```
document.querySelector(".btn.primary")
document.getElementById("main-title")
```

querySelectorAll()
getElementsByTagName()
getElementsByClassName()

- Return collection of elements that are “array like” objects: NodeList
- Way of querying elements by CSS Selector, CSS tag name or class

```
document.querySelectorAll("p")
document.getElementsByClassName(".heading")
```

Open the boilerplate I provided and in the browser console:

1. Grab any element you want.
2. Grab any element you want and append class “highlight”.
3. Grab an element and change it’s style.
4. (HARD) Grab and list out all the li elements. Can you grab the innerText from these? HINT: use map() function with arrays
5. Come up with something of your own share it with us!

Exercise time!

Coffee Break

Event:
User actions or browser actions (click, keypress, scroll, load)

- JavaScript can “listen” and react to these.
- Make web pages interactive

Event Listeners

Mouse Events

- `click`: when an element is clicked

Keyboard Events

- `keydown`: when a key is pressed
- `keyup`: when a key is released

Form Events

- `input`: when a value is typed into a field
- `change`: when a field's value is changed and focus leaves it
- `submit`: when a form is submitted

Page / Window Events

- `load`: when the page finishes loading
- `scroll`: when the user scrolls the page
- `resize`: when the browser window size changes

Common listeners

```
element.addEventListener("eventType", function)
```

- `element`: the thing you're watching (button, input, etc.)
- `eventType`: a string like `"click"`, `"keydown"`, `"submit"`
- `function`: the code to run when it happens

Tip:

For certain actions the browser will have automatic behaviour. `preventDefault()` stops the browser's default action for an event.

For a form **submit**, the default is to reload the page or for a link **click**, the default is to go to another page. By calling `preventDefault()`, you stop that from happening so *your code* can take control instead.

Writing Listeners

Open up the boilerplate event listeners!

Exercise time!

Thank you...