

# JavaScript

## Lecture 3d (The DOM)

Waterford Institute of Technology

June 9, 2016

John Fitzgerald

# JavaScript Introduction

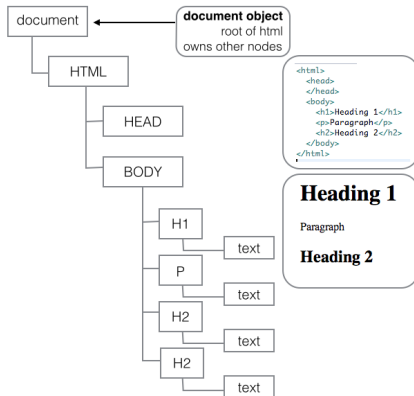
Topic discussed

- Document Object Model (DOM)

# JavaScript

## Document Object Model

### The DOM



# DOM

## Document Object Model

### The **DOM**

- A World Wide Web Consortium (W3C) Standard
- Defines standard for accessing web documents
- Represents the displayed web page
- Each element represented in the DOM by its own object
  - Access and modify individual elements
  - Add and delete elements

```
<script>  
  document.getElementById('demo').innerHTML = 'Hello JavaScript!';  
</script>
```

# HTML DOM Document Object

## HTML DOM Nodes

In the HTML DOM (Document Object Model), everything is a node:

- The document itself is a document node
- All HTML elements are element nodes
- All HTML attributes are attribute nodes
- Text inside HTML elements are text nodes
- Comments are comment nodes

The logo for w3schools.com, featuring the text "w3schools.com" in a bold, sans-serif font. The "w3" is in dark grey, "schools" is in dark grey, and ".com" is in green. The logo is set against a light grey rectangular background.

# DOM

## The Web Browser

On opening HTML document in browser:

- It becomes a **document object**
  - The **document object** is root node of HTML document
  - **document object** provides properties and methods to access node objects from within JavaScript.

```
<script>  
  let x = document.getElementsByName('map');  
  alert(map.length);  
  
</script>
```

# DOM Access

## Demo change text

Click the button to change this text.

Try it

Hi ICTSkills

Try it

# DOM Access

## Using DOM method

```
<!DOCTYPE html>
<html>
  <body>
    <p id="demo">Click the button to change this text.</p>
    <button onclick="domAccess()">Try it</button>
    <script src="dom.js"></script>
  </body>
</html>
```

```
//file: dom.js
function domAccess() {
  document.getElementById('demo').innerHTML = 'Hi ICTSkills';
}
```



# DOM Access

## Using jQuery

```
<!DOCTYPE html>
<html>
  <body>
    <p id="demo">Click the button to change this text.</p>
    <button onclick="jQueryAccess()">Try it</button>
    <script src="jquery-2.0.0.js"></script>
    <script src="jq.js"></script>
  </body>
</html>
```

```
//file: jq.js
function jQueryAccess() {
  $('#demo').html('Hi ICTSkills');
}
```

# HTML Tags

Attributes **name** and **id** are not interchangeable

**name:** Identifies value in form data

**id:** Uniquely identifies an element so you can access it

```
//View (Semantic UI)
```

```
<input id="paypal" name="methodDonated" value="paypal" type="radio">  
<label for="paypal">PayPal</label>  
<input id="direct" name="methodDonated" value="direct" type="radio">  
<label for="direct">Direct</label>
```

```
//Controller (Play): attribute name is methodDonated; content is value  
public static void donate(..., String methodDonated)
```

# HTML Nodes

## Methods to retrieve nodes

- `document.getElementById(id)`
  - **id** unique on a page hence *getElementById*
- `document.getElementsByName(name)`
  - returns array of elements with **name** attribute = *name*
  - **name** need not be unique hence *getElementsByName*
- `node.getElementsByTagName(tagName)`
  - returns array of elements with **tagName** attribute = *tagName*

# Get element by id

Simple demo `document.getElementById(id)`

Prints the height of image whose *id*="img1"

## Native JavaScript

```
//in html file

//in javascript file
let image = document.getElementById('img1');
alert('Image height is ' + image.height);
```

## jQuery

```
//in html file

//in javascript file
alert('Image height is ' + $('#img1').height());
```

# Get elements by name

Simple demo `document.getElementsByName(name)`

Discovers images with attribute *name="imgs"*

## Native JavaScript

```
let images = document.getElementsByName('imgs');  
for (let i = 0; i < images.length; i++) {  
    alert('Image height is ' + images[i].height);  
}
```

## jQuery

```
let $images = $('[name="imgs"]');  
images.each(function () {  
    alert('Image height is ' + $(this).height());  
});
```

# Get elements by tagName

Simple demo `node.getElementsByTagName(tagName)`

Can be used on a sub-tree, not just entire document

## Native JavaScript

```
let imgDiv = document.getElementById('ictskills-images');
let images = imgDiv.getElementsByTagName('img');
for (let i = 0; i < images.length; i++) {
  alert('Image height is ' + images[i].height);
}
```

## jQuery

```
//let images: only those contained in node <div id="ictskills-imgs">
//with attribute name="imgs", e.g.: 
let $images = $('#ictskills-imgs [name=\'imgs\']');
images.each(function () {
  alert('Image height is ' + $(this).height());
});
```

# Hide | Reveal Elements

## Using Native JavaScript

### HTML

```
<p id="text">Watch me appear and disappear</p>  
<button onclick="hide()">Hide</button>  
<button onclick="reveal()">Reveal</button>
```

### JavaScript

```
function hide() {  
    document.getElementById('text').style.visibility = 'hidden';  
}  
  
function reveal() {  
    document.getElementById('text').style.visibility = 'visible';  
}
```

# Hide | Reveal Elements

## Using jQuery

### HTML

```
<p id="text">Watch me appear and disappear</p>  
<button onclick="hide()">Hide</button>  
<button onclick="reveal()">Reveal</button>
```

### jQuery

```
function hide() {  
    $('#text').hide();  
}  
  
function reveal() {  
    $('#text').show();  
}
```



# Semantic UI

## Enable Dropdown Box using JQuery

### HTML

```
<div class="ui selection dropdown">  
  <input name="amountDonated" type="hidden">  
  <div class="default text">Amount</div>  
  <i class="dropdown icon"></i>  
  <div class="menu">  
    <div class="item" data-value="100">$100</div>  
    <div class="item" data-value="200">$200</div>  
    <div class="item" data-value="300">$300</div>  
  </div>  
</div>
```

### jQuery

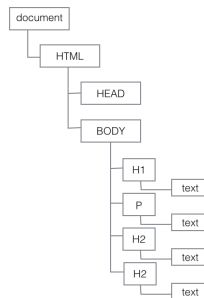
```
//Ensure you use latest version jQuery  
//http://code.jquery.com/jquery-latest.min.js  
<script>$('.ui.selection.dropdown').dropdown('enable');</script>
```

# The DOM

## Concluding with one expert's view

### DOM (Document Object Model)

- "A vast source of incompatibilities, pain and misery" –Douglas Crockford



# Summary

- Document Object Model (DOM)
  - HTML page underlying data structure.
  - Difficult development environment.
  - Better to use jQuery v native JavaScript.

# References

Applicable to JavaScript 3a, 3b, 3c, 3d

1. Simpson Kyle (2015). You Don't Know JS: ES6 & Beyond. O'Reilly Media

[http://shop.oreilly.com/product/0636920033769.do?  
sortby=publicationDate](http://shop.oreilly.com/product/0636920033769.do?sortby=publicationDate)

[Accessed 2016-05-09]

2. MDN: Mozilla Developer Network - Arrow functions

[https://developer.mozilla.org/en/docs/Web/JavaScript/  
Reference/Functions/Arrow\\_functions](https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Functions/Arrow_functions)

[Accessed 2016-06-16]

3. W3Schools JavaScript Tutorial

<http://www.w3schools.com/js/>

[Accessed 2016-06-16]



Except where otherwise noted, this content is licensed under a Creative Commons Attribution-NonCommercial 3.0 License.

For more information, please see <http://creativecommons.org/licenses/by-nc/3.0/>



Waterford Institute of Technology  
INSTITIÚD TEICNEOLAÍOCHTA PHORT LÁIRGE

