Objects and DOM

Web Authoring and Design

Benjamin Kenwright

Outline

- What do we mean by Objects and DOM?
- Summary
- Review/Discussion

DOM

- The DOM (Document Object Model) is an interface to the web document provided by the browser manufacturer
- Within this model, each element in the HTML document becomes an Object

Nodes

- In order to work with the browser and documents, JavaScript uses a hierarchical tree structure of parent and child Objects
- The main object is the Document Object, which in turn contains several other child objects
- Each Object or element in the document is called a Node in the DOM

Question

- What does DOM stand for?

Answer

- DOM (Document Object Model)

Tree Structure

- The DOM represents a document as a hierarchical tree of nodes, which can have parents, children, and siblings and this determines by its position in the tree structure

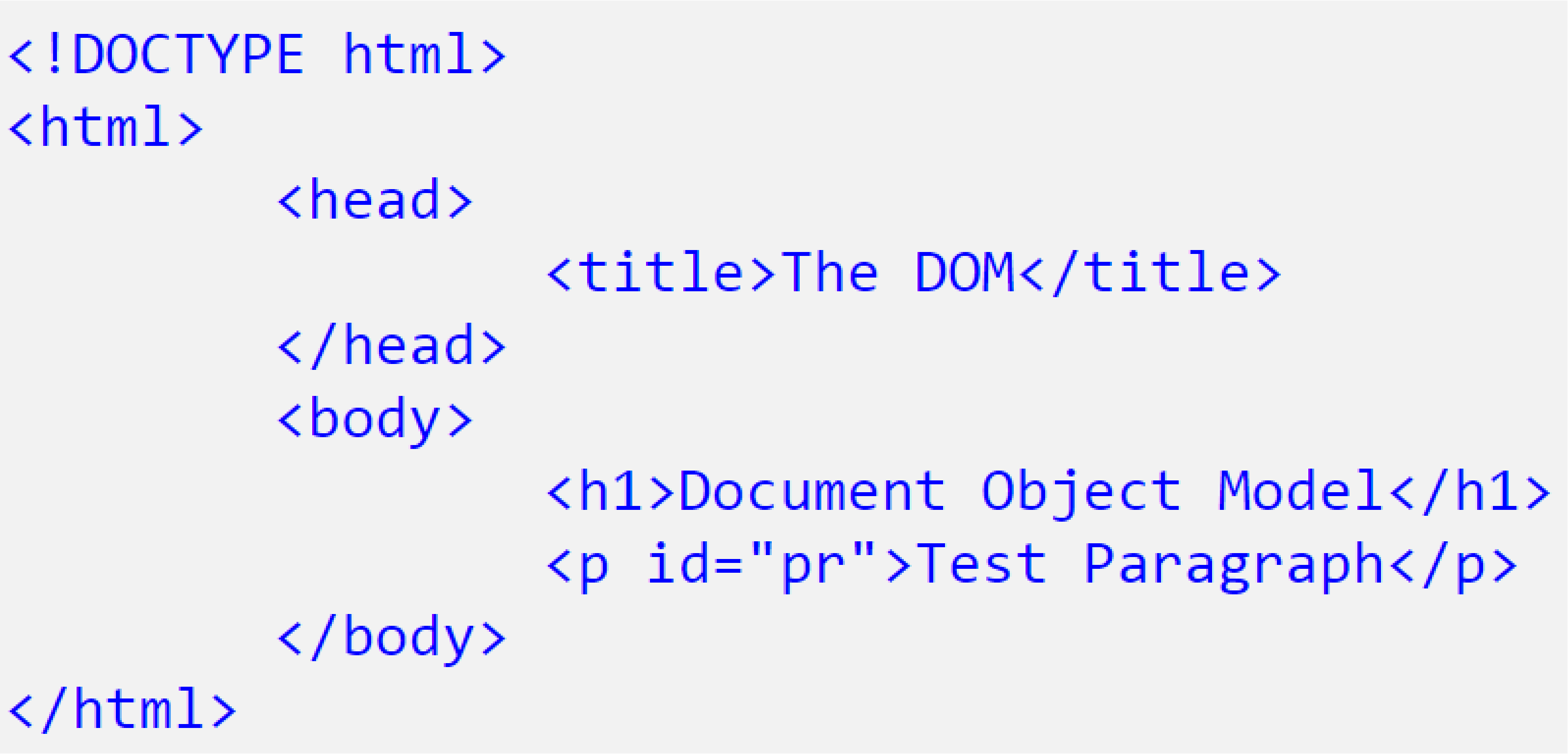
Node Types

- There are several node types in the tree, each representing different information or markup in the HTML document
- Each node type has different properties, methods, data, events, and each may have relationships with other nodes

HTML Document

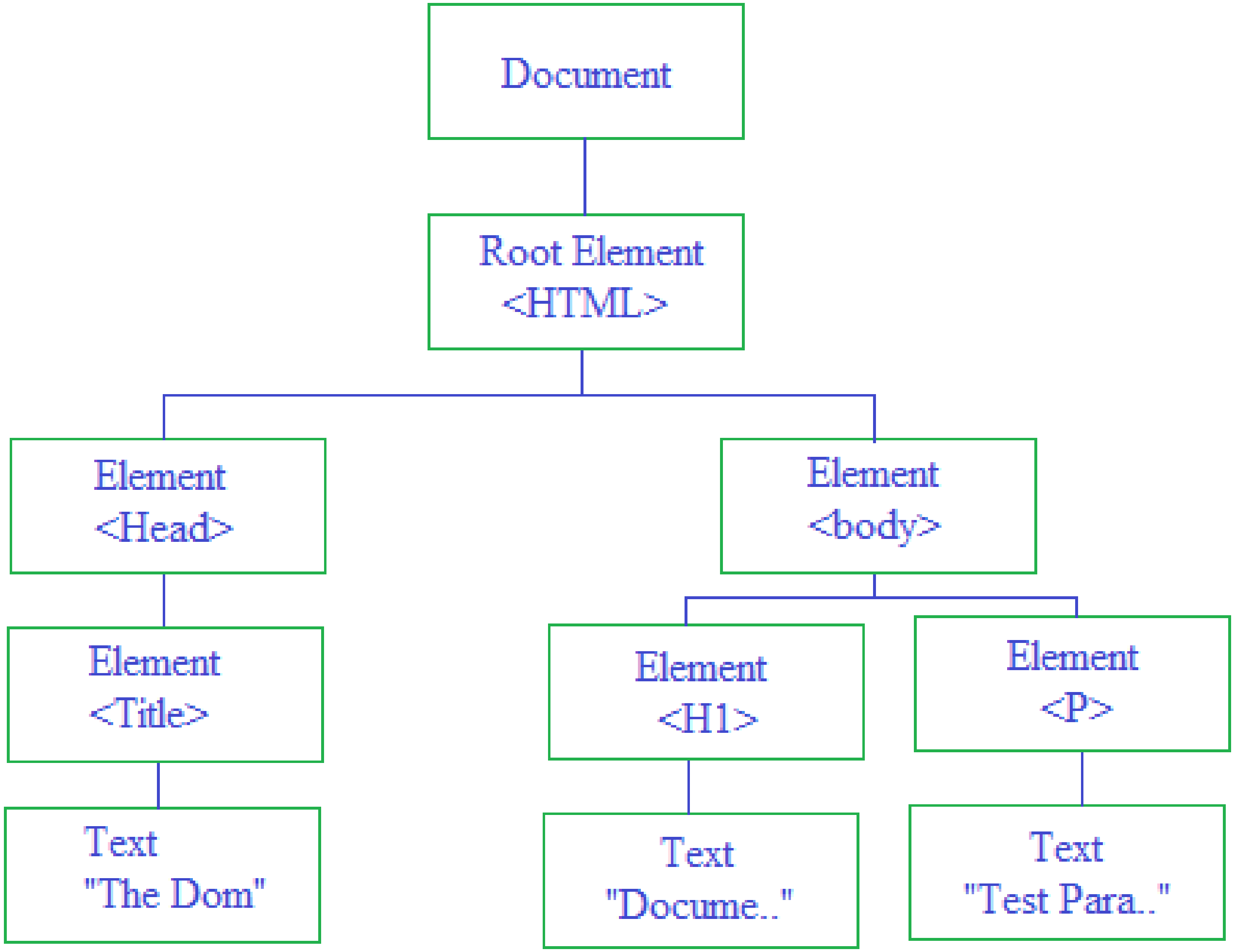
- The Document Object Model provides a uniform representation of the HTML document, and it achieves this by representing the entire HTML document as a tree structure
- When a web page is loaded in the browser, it creates a Document Object Model of the web page
- Each and every single element in the document will have a corresponding presence in the DOM

Example



```html
<!DOCTYPE html>
<html>
        <head>
                <title>The DOM</title>
        </head>
        <body>
                <h1>Document Object Model</h1>
                <p id="pr">Test Paragraph</p>
        </body>
</html>
```

Example
Document Object Model



Document

Root Element
<HTML>

Element
<Head>

Element
<body>

Element
<Title>

Element
<H1>

Element
<P>

Text
"The Dom"

Text
"Docume.."

Text
"Test Para.."

Nodes

- Nodes within the DOM are represented by array-like node lists and the individual nodes themselves can be accessed via their index
- Using the above tree of nodes, you can access any element in the DOM

Example

- **document.childNodes[1]**
  - represents the HTMLElement, that is < html > tag.
- **document.childNodes[1].childNodes[1]**
  - represents HTMLBodyElement, that is < body > tag.
- **document.childNodes[1].childNodes[1]. childNodes[1]**
  - represents HTMLHeadingElement, that is < h1 > tag

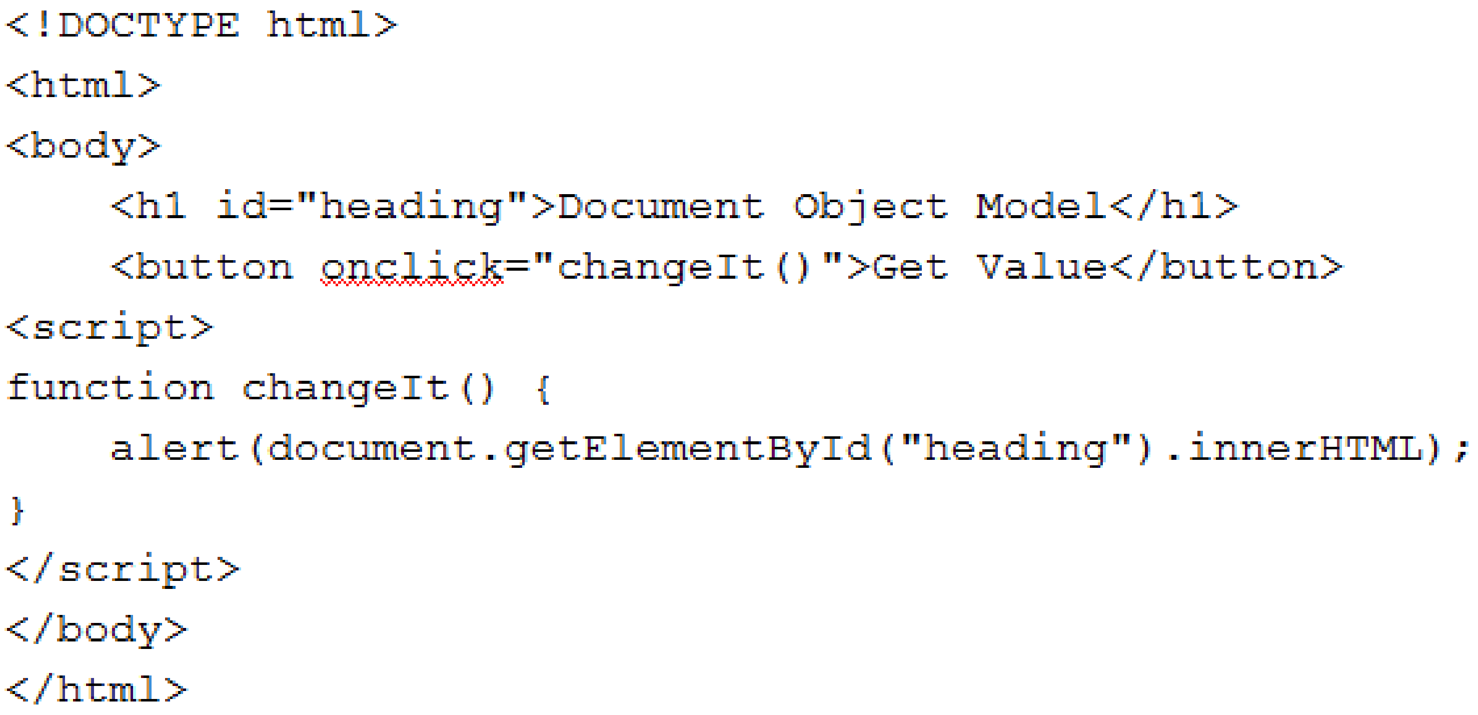
DOM Methods

- The getElementById() and getElementsByTagName() were the two methods from DOM standard
- HTML5 specification adds three new methods for accessing elements, getElementsByClassName(), querySelector(), and querySelectorAll()

getElementbyId()

- Typically you want to access an element within the DOM directly and try to do something with it
- Javascript provides a document.getElementById() method, which is the easiest way to access an element from the DOM tree structure
- It will return the element that has the ID attribute with the specified value.

Example

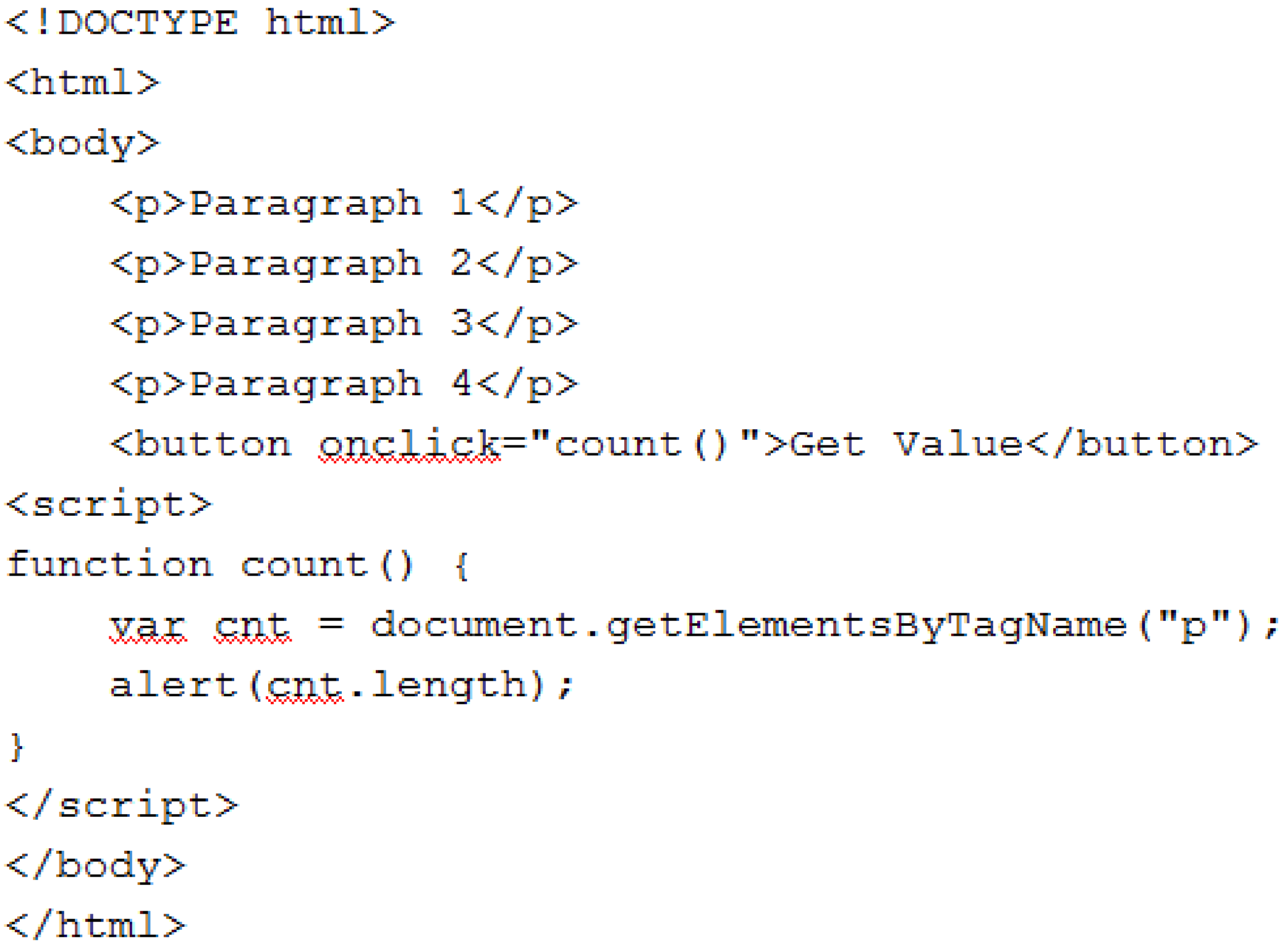


```
<!DOCTYPE html>
<html>
<body>
    <h1 id="heading">Document Object Model</h1>
    <button onclick="changeIt()">Get Value</button>
<script>
function changeIt() {
    alert(document.getElementById("heading").innerHTML);
}
</script>
</body>
</html>
```

getElementsByTagName()

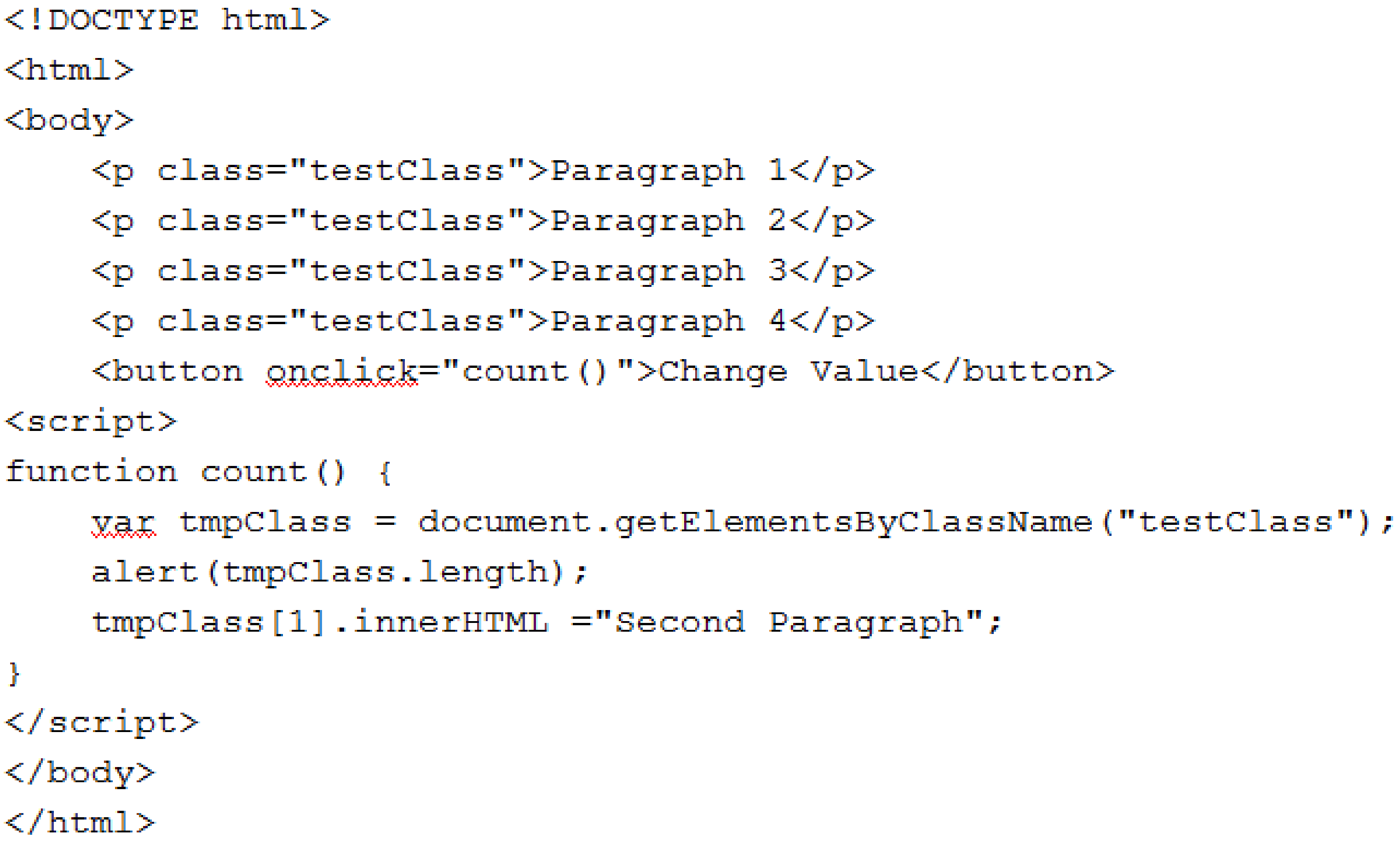
- The getElementsByTagName() is one of the method exposes for accessing nodes directly
- This method takes a tag name as argument and returns a collection of all the nodes it finds in the document that are a sort of tag

Example



```
<!DOCTYPE html>
<html>
<body>
    <p>Paragraph 1</p>
    <p>Paragraph 2</p>
    <p>Paragraph 3</p>
    <p>Paragraph 4</p>
    <button onclick="count()">Get Value</button>
<script>
function count() {
    var cnt = document.getElementsByTagName("p");
    alert(cnt.length);
}
</script>
</body>
</html>
```

getElementsByClassName()



```
<!DOCTYPE html>
<html>
<body>
    <p class="testClass">Paragraph 1</p>
    <p class="testClass">Paragraph 2</p>
    <p class="testClass">Paragraph 3</p>
    <p class="testClass">Paragraph 4</p>
    <button onclick="count()">Change Value</button>
<script>
function count() {
    var tmpClass = document.getElementsByClassName("testClass");
    alert(tmpClass.length);
    tmpClass[1].innerHTML ="Second Paragraph";
}
</script>
</body>
</html>
```

Summary

- Overview of Objects and DOM
- Structure of Website
- Examples

This Week

- Review Slides
- Read Associated Chapters
- Work through Examples
  - Setup GitHub Account/Webpage
- **Exam**
- **Group Project**

Questions/Discussion