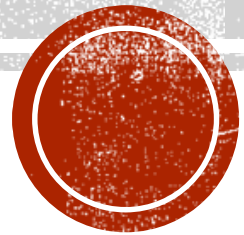
WEEK #8: THE DOCUMENT OBJECT MODEL



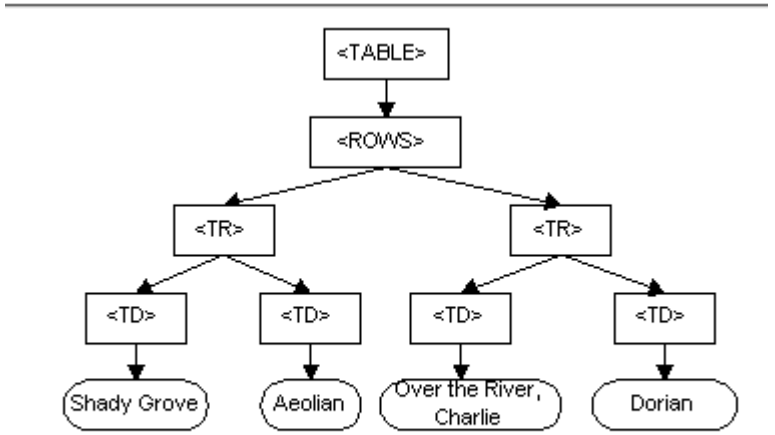
Maricel Medina-Mora

PROG 109 – Bellevue College

DEFINITION

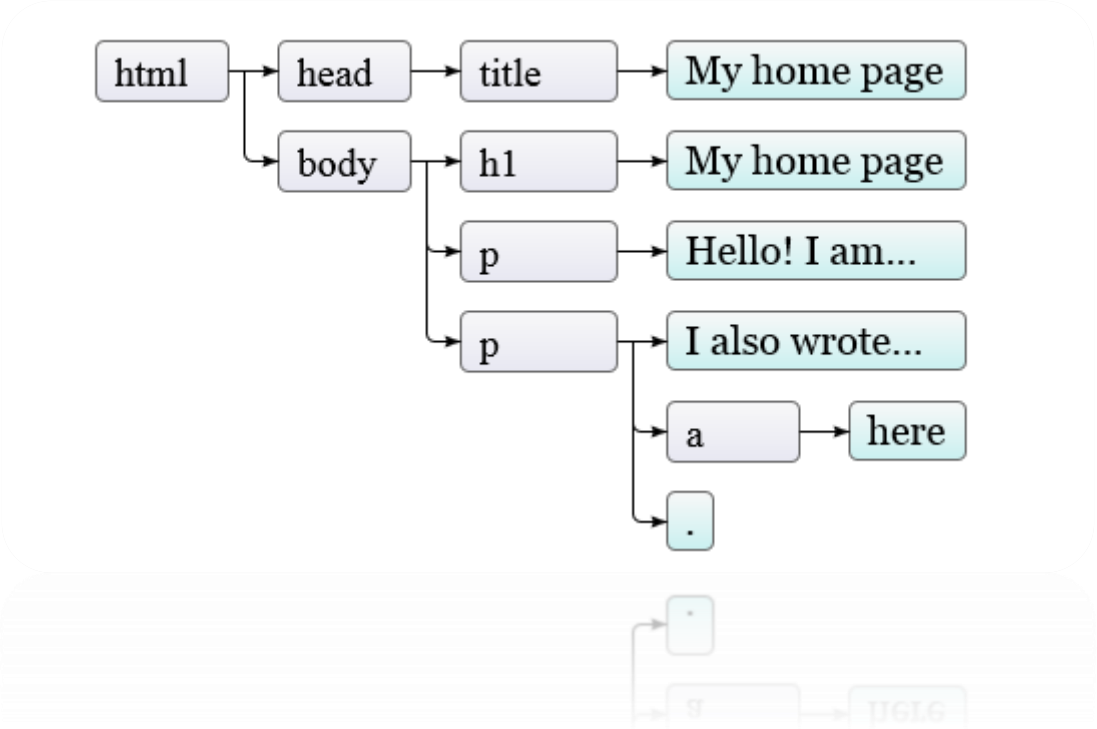
The Document Object Model (DOM) is a programming API for HTML and XML documents. It defines the logical structure of documents and the way a document is accessed and manipulated. The Document Object Model originated as a specification **to allow JavaScript scripts and Java programs to be portable among web browsers.**

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



DOM representation of the example table

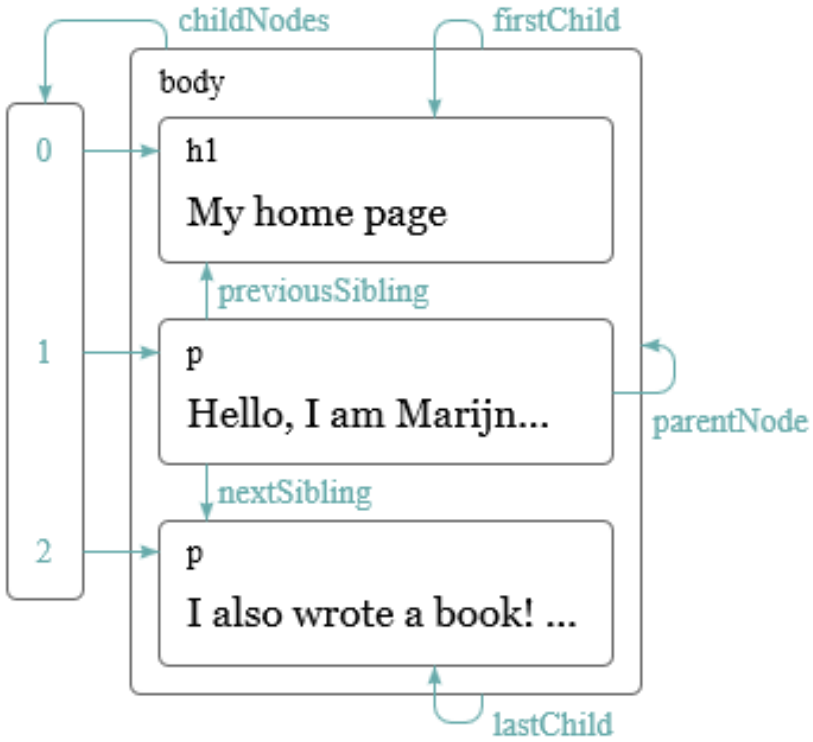
DOM TREE STRUCTURE



Haverbeke, http://eloquentjavascript.net/13_dom.html

- **document.documentElement** serves as the root.
- Nodes for regular elements, **which represent HTML tags**, determine the structure of the document. Example:
  - **document.body**
- Each DOM node object has a **nodeType** property, which contains a numeric code that identifies the type of node (see W3Schools for codes)
- Regular elements have the value 1, which is also defined as the constant property **document.ELEMENT_NODE**

DOM TREE STRUCTURE



DOM nodes contain a wealth of links to other nearby nodes

- **parentNode**
- **childNodes**
- **firstChild**
- **lastChild**
- **previousSibling**
- **nextSibling**

Haverbeke, http://eloquentjavascript.net/13_dom.html

FINDING ELEMENTS

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>JS Bin</title>
</head>
<body>
<p id="p1">My ostrich Gertrude:</p>
<p><img id="gertrude"
src="http://wfiles.brothersoft.com/o/ostrich-face_113665-
1440x900.jpg" height=75 ></p>

<p id="p2">Here comes the result of JS: </p>
  <div>
  </div>
</body>
</html>
```

- **getElementsByTagName:** All element nodes have a method, which collects all elements with the given tag name that are descendants (direct or indirect children) of the given node and **returns them as an array-like object (i.e. HTMLParagraphElement)**
- **document.getElementById:** Finds a specific node
- **getElementsByClassName:** Finds a specific class

Haverbeke, http://eloquentjavascript.net/13_dom.html

MODIFYING THE DOM

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>JS Bin</title>
</head>
<body>
<p>One</p>
<p>Two</p>
<p>Three</p>
<div>
</div>
<script>
  var paragraphs =
document.body.getElementsByTagName("p");
document.body.insertBefore(paragraphs[2],paragraphs[0]);
</script>

</body>
</html>
```

- **document.write:** Modifies the DOM in the position it has been called
- ***Element*.innerHTML:** Modifies the content of the element
- **removeChild:** Removes the given child node from the document.
- **appendChild:** Add a child at the end of the list of children
- **insertBefore:** Which inserts the node given as the first argument before the node given as the second argument.

Haverbeke, http://eloquentjavascript.net/13_dom.html

MORE ABOUT MODIFYING THE DOM

```
var paragraphs = document.getElementsByTagName("p");
for (var i=0; i < paragraphs.length; i++) {
  console.log("Paragraph " + paragraphs[i].id);
}


var ostrich = document.getElementById("gertrude");
  console.log(ostrich.src);

var myDiv = document.querySelector( "div" );
myDiv.innerHTML = "Foo";
```

MORE ABOUT MODIFYING THE DOM

```
<script>
  var paragraphs = document.body.getElementsByTagName("p");
document.body.insertBefore(paragraphs[2],paragraphs[0]);
  paragraphs[0].style.color ="red";
  paragraphs[1].style.color ="blue";
  paragraphs[2].style.color ="green";
</script>
```

RESOURCES

- DOM:

http://www.w3schools.com/jsref/dom_obj_document.asp

http://www.w3.org/TR/WD-DOM/introduction.html