Document Object Model

Document Object Model

- DOM is a cross-platform and language independent mechanism for interacting with elements in HTML, XHTML, and XML
- Use DOM to dynamically manipulate elements in HTML documents (usually via JavaScript)
- DOM is not a set of data structures and does not define how information in a document is structured

DOM History - Legacy

- DOM began as a way for JavaScript scripts and Java programs to be shared among browsers
- JavaScript first available in 1996 with Netscape browser (v2) – at that time the #1 browser
- Microsoft's IE v3 browser featured its own JScript
- Both JavaScript and JScript let web designers create web pages having client interaction

e.g. `document.myForm.inputName` or `document.forms[0].elements[0]`

Level 0 DOM example

- In JavaScript confirm the current browser understands the term `document.images`
- Netscape Navigator v3 supports it but not Microsoft's Internet Explorer v3

```
if (document.images)
{
 // do something with document.images
 document.images['theName'].src = 'newImage.gif';
}
```

Navigator understands this term. Not MS IE v3.

DOM History - DHTML

- DHTML (dynamic HTML) became available in 1997 for both Netscape Navigator v4 and MS browsers
- DHTML provided enhanced changes to the loaded HTML document in the browser
- The two big browser makers worked separately on their own implementations of DOM extensions – led to incompatibility of web pages
- This cross-browser development required each browser to handle each DOM flavour separately

Intermediate DOM example

- Netscape v4 and IE v4 browsers handled DOM differently
- Note how the JavaScript used to handle it:

```
if (document.layers) {       // Netscape
 document.layers['layername'].top = 200;
} else if (document.all) {   // IE
 document.all['layername'].style.top = 200;
}
```


DOM History - Standards

- In 1994 W3C brought together Netscape Communications and Microsoft with other companies to agree on a cross-browser standard for scripting languages – ECMAScript was born in 1997
- After ECMAScript W3C standardized the DOM (Level 1 in 1998, Level 2 in 2000, Level 3 in 2004)
- By 2005 all common browsers support W3C standards

document tree

- In Level 1 DOM all HTML elements are part of the document 'tree'
- HTML documents have two 'branches' : head and body
- The head branch has at least one sub-branch: title
- The body branch will have sub-branches : any `p`, `div`, `h1`, etc

document tree

document.childNodes[1].childNodes[1].appendChild(document.childNodes[1].childNodes[3])

DOM enables walking the HTML document tree

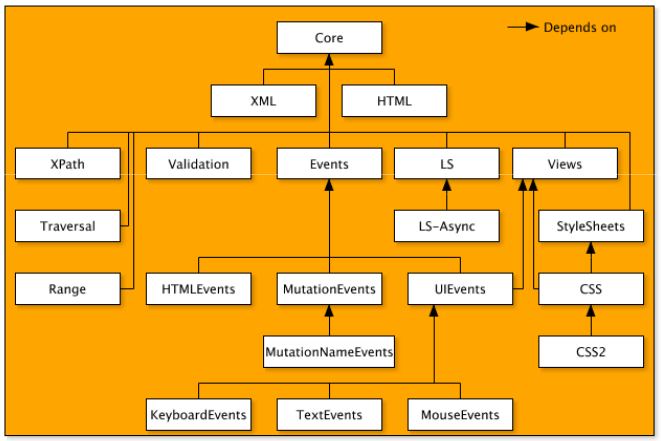
But going through the tree node by node is time-consuming … alternative is to use an element's ID

```
<p id="this_is_unique"> This HTML
  document has a paragraph.</p>
document.getElementById('this_is_unique')
```


HTML DOM Objects

1. DOM Document
2. DOM Events
3. DOM Elements

DOM Architecture

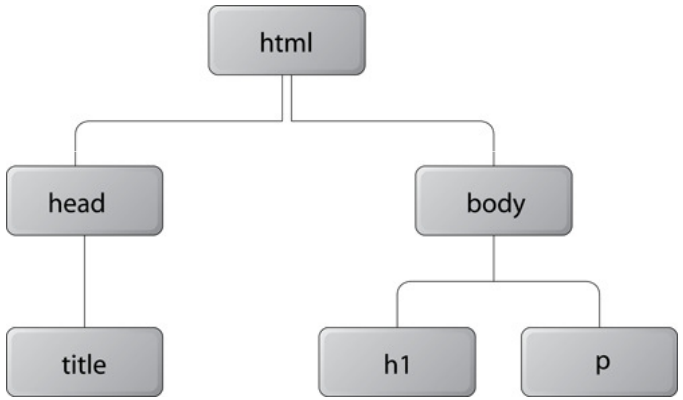


http://www.w3.org/TR/2004/PR-DOM-Level-3-Core-20040205/introduction.html

Example HTML test.html

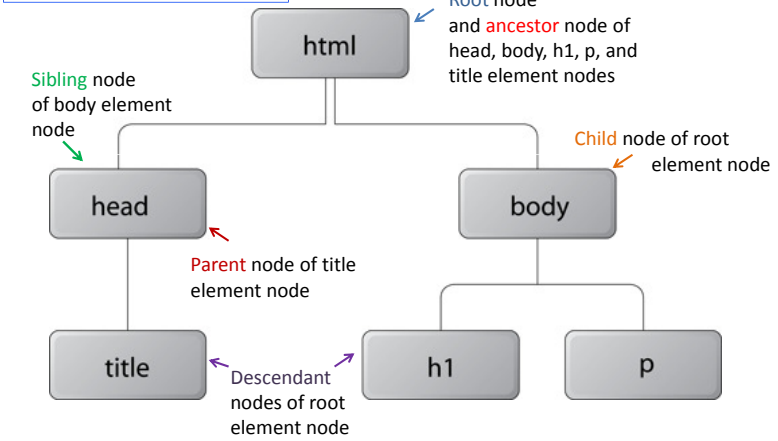
```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title> Gadgets </title>
  </head>
  <body>
    <h1>Gadgets</h1>
    <p>Welcome to Gadgets, the number one company
       anywhere for selling gadgets!
    </p>
  </body>
</html>
```

DOM tree from test.html



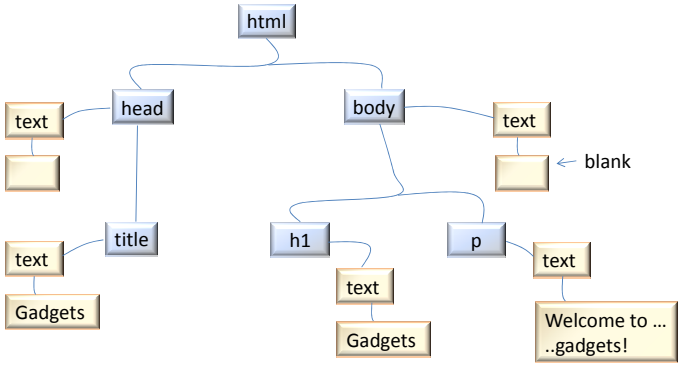
DOM tree from test.html

Text nodes and attribute nodes not shown for clarity.



DOM tree from test.html

Text nodes shown:



Example XHTML with JavaScript

```
<html lang="en">
<head>
 <title>Example DOM</title>
 <script type="text/javascript">
    function url_Load() {
       document.getElementById("thisURL").innerHTML = document.URL;
    }
 </script>
</head>
<body onload="url_Load();">
 <h1>Average Chapter</h1>
 <p>The <em>average</em> or central tendency of a <i>data set</i>
    is a measure of the <span>middle</span> value of the data set.
 </p>
 <div id="thisURL">
 </div>
 </body>
</html>
```

Current HTML document's URL

Execute the JavaScript function.

Reference the ID of the division.

JavaScript function `url_Load` places the URL inside this div.

1.0 DOM Document

- Any HTML document loaded into a browser is a document object
- Each element of that HTML document is accessible via DOM
- The document object is also part of the Window object as in `window.document` property

1.1 DOM Document Properties

- cookie
- documentMode
- domain
- lastModified
- readyState
- referrer
- title
- URL

1.2 DOM Document Methods

- close()
- getElementById()
- getElementsByName()
- getElementsByTagName()
- open()
- write()
- writeln()

1.3 DOM Document Collections

- anchors[]
- forms[]
- images[]
- links[]

2.0 DOM Events

- onblur – this event fires when the element loses focus (i.e. user tabs out of the element)
- onchange – element content text is modified
- onclick – element has been clicked by mouse
- onerror – triggered when an error occurs loading a document or an image
- onfocus – element receives focus (i.e. user tabs into the element)
- onkeydown – a keyboard key is pressed
- onkeypress – a keyboard key is pressed or held down
- onkeyup – a keyboard key is released

2.1 DOM Events

- onmousedown – a mouse button is pressed
- onmousemove – the mouse is moved
- onmouseout – the cursor is moved off an element
- onmouseover – the cursor is moved over an element
- onmouseup – the mouse button is released
- onresize – a window or a frame is resized
- onselect – text is selected
- onunload – user exits the page

Example onblur event

```
<html>
<head>
  <script type="text/javascript">
    function upperCase() {
      var x=document.getElementById("fname").value;
      document.getElementById("fname").value=x.toUpperCase();
    }
  </script>
</head>

<body>
  Enter your name:  <input type="text" id="fname" onblur="upperCase();" />
</body>
</html>
```

When this element loses focus, the JavaScript function named upperCase will run.

2.2 DOM Events – Mouse/Keyboard

- altKey
- button
- clientX
- clientY
- ctrlKey
- metaKey
- relatedTarget
- screenX
- screenY
- shiftKey

3.0 DOM Elements - Properties

- accessKey
- className
- disabled
- id
- innerHTML
- clientHeight
- clientWidth
- style
- title
- width

3.1 DOM Elements - Methods

- blur()
- click()
- focus()
- getElementsByTagName()
- appendChild()
- removeChild()
- replaceChild()

- http://www.w3.org/TR/2004/REC-DOM-Level-3-Core-20040407/core.html