PIC 40A

Lecture 12: JS: DOM

What is DOM?

The Document Object Model (DOM)

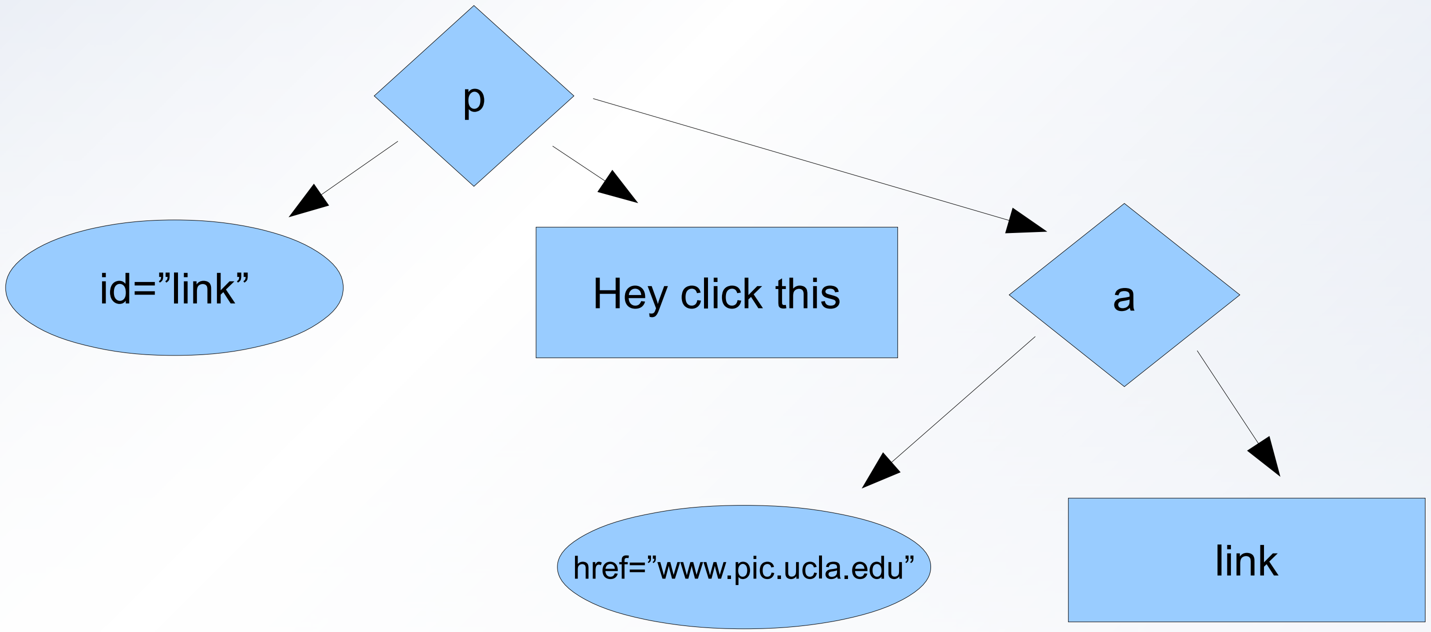
A browser's representation of its document in memory

Models an XML document as a tree of nodes

Defines an interface between XML docs and application programs such as JavaScript scripts

What is a node tree

```
<p id=”link”>
   Hey click this
   <a href=”www.pic.ucla.edu”>link</a>
</p>
```



Everything in an XHTML document is a node

In the DOM

- The entire document is a document node
- Every XHTML element is an element node
- The text in the XHTML elements are text nodes
- Every XHTML attribute is an attribute node
- Comments are comment nodes

Note: White space counts as text

Only text nodes contain text. Element nodes do not contain text!

DOM Example

To view a DOM tree on Firefox you need to download Firefox add on called DOM Inspector

For IE8 download Developer Toolbar

Terminology

The DOM tree structure is called a node-tree

Top node is called the **root**

If a node has another node immediately below it it is called a **parent node**

The node immediately below the parent node is a **child node**

parent node can have any number of children

children of the same parent are called **siblings**

a **leaf** is a node with no children

Observation: All nodes except the root node have exactly one parent

What does the DOM have to do with JS?

Everything in an XHTML document has a corresponding node in the DOM

Every node in an XHTML DOM can be retrieved as a JS object.

Once we have a node as a JS object we can manipulate it in almost any way.

Any changes we make to the element will be immediately rendered by the browser!!

How do I get a node object?

It is easy to get a node object of an XHTML element with an id attribute

```
<p id="intro">Hello! What's up?</p>
```

To get the corresponding node object the JS use document objects `getElementById` method.

```
var eltNode = document.getElementById("intro");
```

What if my element has no id?

Get all the elements of a given type as an array using `getElementsByTagName`, then look for the one you want. For example to get all the objects corresponding to anchor elements in a document

```
var anchor_array = document.getElementsByTagName("a");
```

`anchor_array[0]` is the first anchor in the document

`document.getElementsByTagName("*")` gets all document elements

Modifying a DOM element node

- change its content (only if element is XHTML)

```
eltNode.innerHTML="Hello";
```

- change its attribute or add a new one

```
eltNode.setAttribute("class", "intro");
```

- get an existing attribute's value

```
var attValue = eltNode.getAttribute("class");
```

  - get an attribute node

```
var attNode = eltNode.getAttributeNode("class");
```

Example

See example

When I change the DOM structure, am I changing the document's file?

No, by changing the DOM structure, you are only changing the browser's model for rendering the document.

The changes are not made to the document's file, just the browser's DOM representation of it in memory.

Example

We can use our ability to change elements attributes in conjunction with existing style sheet to create some nice effects.

See example.

DOM node properties

These are all DOM nodes:

parentNode – Get the parent object

firstChild – Get first child object

lastChild – Get the last child of the element

nextSibling – Get the next sibling object

previousSibling – Get the previous sibling object

Syntax:

Given a node object elt

```
var node_obj = elt.property
```

Example

```
<div>
  <p id=”some_text”>Some text<p>
</div>

var eltobj = getElementById(“some_text”);
var parentobj = eltobj.parentNode;
```

Some more properties

childNodes property

Gives an array of child nodes.

```
elt.childNodes
```

Example:

```
var third_child=elt.childNodes[2];
```

nodeValue Property

The nodeValue property specifies the value of a node.

- nodeValue for element nodes is undefined
- nodeValue for text nodes is the text itself
- nodeValue for attribute nodes is the attribute value

DOM node methods

Syntax:

nodeobj.nodemethod(parameters are usually other node objects);

```
appendChild(node)
```

- Adds a new child element to the end of the list of children of the element

Example:

```
parent_node.appendChild(newchild) ;
```

More node methods

`replaceChild(newnode,oldnode)`

Replaces a child node with a new child node

Example:

```
parent_node.replaceChild(newChild, oldChild);
```

`insertBefore(newnode,beforethis)`

Inserts a new node before an existing node

Example:

```
elementNode.insertBefore(new_node,existing_node;)
```

More node methods

`removeChild(node)`

Removes a child node from a parent.

Example:

```
parent_node.removeChild(child_node);
```

`createElement(elementname)`

Creates a new node

Example:

```
var newNode = document.createElement("div");
```

`createTextNode(text)`

Creates a text node

Example:

```
var newNode = document.createTextNode("Some text");
```

How do I create my own nodes in the DOM tree?

Step 1.

Create a node object

Step 2.

Place the node object in the DOM by making it a child node of another node by using `appendChild` or by `insertBefore`

Example:

Lets say I have something like this in my xhtml document.

```
    <div id="my_div_node"></div>
```

In my head section I might have a script in some function

```
//Create an element but it is not in the DOM yet
var my_link_node = document.createElement("a");

//Set the attribute of the node
my_link_node.setAttribute("href", "www.math.ucla.edu");

//Create a new text node
var my_link_text = document.createTextNode("Go here");

//Set the text content of the link
my_link_node.appendChild(my_link_text);

//get our div element so we can add the link to it

var my_div_node = document.getElementById("my_div_node");

//Finally add the link node to the div element
my_div_node.appendChild(my_link_node);
```

Big DOM Example

See the examples page.

JS dynamic style

1. Use a combination of CSS and dynamically setting class or id attributes

2. Set style property of a node object

1. Using existing CSS and dynamic class or id attributes

Suppose you have an element

```
<p id="example"> Some text </p>
```

and suppose that you already have an existing CSS rule:

```
.red {color: red}
```

We can dynamically change our paragraph to belong to the red class

```
pnode = getElementById("example");
pnode.setAttribute("class", "red");
```

Similarly we could change the id attribute

2. Setting style directly

syntax:

```
objnode.style.property = “value”
```

Example:

```
pnode = document.getElementById(“my_paragraph”);
pnode.style.color=”red”;
```

- Most properties you are used to from CSS are available
- Properties with hyphen are written without the hyphen and with the word after the hyphen capitalized.

```
eg. z-index is zIndex
```

Some available style properties

```
style.border
style.margin
style.padding
style.overflow
style.display
style.verticalAlign
style.visibility
style.listStyle
style.borderCollapse
style.textAlign
style.textDecoration
```

```
style.textIndent
style.textTransform
style.whitespace
style.color
style.font
style.width
style.top
style.left
style.right
style.background
style.zIndex
style.height
style.bottom
```

Animation example

You will have a lab on animation where you will learn more techniques.

Example: Scrolling text for midterm review