

Dynamic HTML

CS 640, Lecture 8



What is Dynamic HTML?



- Generally defined as web pages reacting to user actions without server interaction
- There is no single standard
 - Cascading Style Sheets (CSS) allow fine control over how html elements are positioned on page
 - JavaScript code runs inside the browser
 - Triggered by events generated by the browser
 - Document Object Model (DOM) describes how the document object should be manipulated
- Widespread support, incompatibilities remain

Overview of lecture



- CSS
- DOM
- Dynamic web pages

How stylesheets work



- A style sheet is a collection of rules
- Rules consist of two parts
 - References to one or more HTML elements (selectors)
 - Rules also apply to elements inside those referenced
 - One or more style sheet attributes applied to them
 - For each attribute, the value assigned to it
- Multiple rules can refer to the same element, the cascading preference gives their priority

```
p {color:#0000ff;font-size:14pt}
a, h3 {color:green}
```

Further CSS selectors



Name	HTML	CSS
Class selector	<code><p class="big"></code> <code><h2 class="hot"></code>	<code>p.big{font-size:large}</code> <code>.hot{color:red}</code>
ID selector	<code><p id="special"></code>	<code>#special{border:ridge red}</code> or <code>p#special{...}</code>
Descendant sel.	<code><p>This</code>	<code>p em{background-color:blue}</code>
Pseudoclasses		<code>a:link,a:active,a:visited</code> <code>:hover</code>

- Many HTML elements can use the same class, but the value of the id attribute must be unique
- Pseudoclasses identify various states of elements: visited links, the mouse hovering over a paragraph

Adding style sheets to pages



- Rules inside the `<style></style>` tag
 - Place it inside `<head></head>`
 - For safety use `<!-- -->`
 - Can read external files
 - `@import url("coolstyle.css");`
 - Must precede other rules
- The style attribute of HTML tags
 - Selectors not needed, applies to HTML element
 - `<p style="color:yellow">`
 - Don't use it, mixes content and presentation

Some CSS attributes



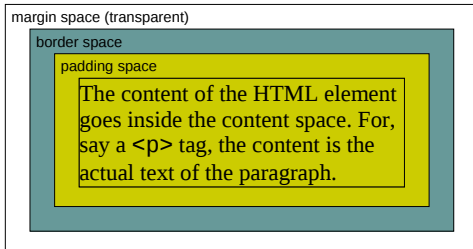
- **font-color** specifies the color of text
 - Six digit hexadecimal RGB representation
 - Names such as red, green, yellow, darkred, etc
- **font-size** specifies absolute or relative font size
 - In various units, percentages, symbolically
 - 12pt, x-small, medium, larger, 150%
- **font** can specify font size, color and other characteristics with a single attribute
- Other attributes: text-align, vertical-align, background-color, background-image

Size units used in CSS



Unit	Description
em	Element's font's height
ex	The height of a lower case x
px	Pixel (exact size depends on display)
in	Inch
cm	Centimeter
mm	Millimeter
pt	Point = inch/72
pc	Pica = 12 points

Borders, margins, padding



More CSS attributes



- height and width can specify the absolute or relative size for element (only content space)
- border-style can be solid, dashed, dotted, double, groove, ridge, inset, outset, none
- border-width, border-color
- border gives width, style, and color (in this order)
- margin and padding specify width only
- Other attributes: margin-bottom, border-top, border-right-color, border-left-style

Two useful HTML tags



- <div></div> and can enclose other HTML elements
 - Used as containers that pass on their CSS attributes to elements inside
 - <div> defines a paragraph-like rectangular area
 - can apply to any small piece of text

```
<head>
<title>A CSS Binge</title>
<style type="text/css"><!--
  p{color:#0000ff;font-size:14pt}
  a,h2{color:green}
  p.big{font-size:x-large}
  .halfpage{width:50%;border:double}
  #special{border:ridge red;width:40%}
  p em{background-color:red}
  p:hover{background-color:yellow}
--></style>
</head>
<body><h1>Style Sheet Binge</h1>
Normal text with a <a href="bucky.gif">link</a>.
<h2>Going crazy with <em>colors and fonts</em></h2>
<p>Text within a paragraph that includes a <a href="buckys.gif">link</a>.</p>
<p class="big">This text is <em>big</em> !</p>
<div class="halfpage"><p id="special">This is a special paragraph.</p>
<p>Another paragraph with a lot of text that shows us where text wraps.</p>
</div></body>
```



Some CSS positioning



- position identifies rule as giving position
 - absolute – position is absolute
 - fixed – similar but ignores scrolling
 - Other values possible
- top, left, right, bottom specify offset distance between element's positioning context and element's box (includes margins)

Overview of lecture



- CSS
- DOM
- Dynamic web pages

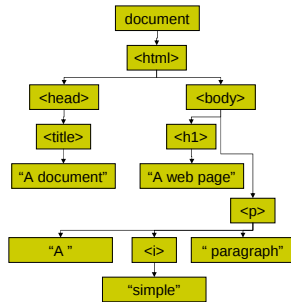
Document Object Model



- Describes how the document object can be traversed and modified
 - Represented as tree structure
- Two approaches in use
 - IE-specific more convenient for HTML
 - W3C more verbose, but also applies to XML
- DOM has levels 0-3 and many sub-standards
- The DOM interface used in other contexts with other languages (C++, Java, python, etc.)

The document as a tree

```
<html>
<head>
  <title>A Document</title>
</head>
<body>
  <h1>A web page</h1>
  <p>A <i>simple</i>
    paragraph</p>
</body>
</html>
```



Manipulating nodes

- Traversing the element tree
 - Each node has `childNodes` array
 - Can use properties `firstChild`, `lastChild`, `nextSibling`, `previousSibling`
 - Firefox's DOMInspector visualizes the DOM tree
- `nodeType` property can be 1 (element), 2 (attribute), 3 (text), 8 (comment), 9 (document)
- Can change structure using `appendChild()`, `removeChild()`, `replaceChild()`, `insertBefore()`

Tag attributes

- Attribute nodes are ignored during traversal
- Elements have properties for attributes
 - Words capitalized – e.g. the `body` element has a `bgColor` property corresponding to the HTML attribute `bgcolor`
 - Can assign strings to these properties
 - Can also treat `style` attribute as an object with properties of its own
- Elements have methods `getAttribute()`, `setAttribute()`, `removeAttribute()`

More DOM manipulation



- The document object (and element objects) have methods for finding specific elements
 - `getElementsByTagName()` returns an array with all elements with the given tag name
 - `getElementsByTagName()` returns an array with all elements with given name
 - `getElementById()` returns element with given ID
- To build new nodes, use the document object's methods `createElement(tagName)` and `createTextNode(text)`
- Text node have `appendData()`, `insertData()`, `deleteData()`, `replaceData()` methods

Overview of lecture



- CSS
- DOM
- Dynamic web pages

JavaScript timers



- Used extensively in dynamic pages
- `setTimeout(code, delay)` tells browser to execute `code` in `delay` milliseconds
- If you save the return value, you can cancel using `clearTimeout(timeoutID)`
- `setInterval()` and `clearInterval()` work similarly, but code is run periodically instead of just once

Dyn

```
function changeBGColor(color){
    var p=document.getElementById("para1");
    p.style.backgroundColor=color;
}
function checkColor(){
    var s=document.getElementById("textfield1").value;
    if (s.length!=6){
        alert("Must enter six hex digits");
        return;
    }
    for (var i=0;i<6;i++){
        if(!((s[i]>="A" && s[i]<="F")||
            (s[i]>="a" && s[i]<="f")||
            (s[i]>="0" && s[i]<="9")))){
            alert(" Character '"+s[i]+' is not valid");
            return;
        }
    }
    changeBGColor("#"+s);
}
```



Ani

```
var pos=0;
function runAway(){
    var image=document.getElementById("bucky");
    if(pos==0){
        image.style.left="250px";
        image.style.top="50px";
        pos=1,vpos=50;
    } else {
        image.style.left="50px";
        image.style.top="50px";
        pos=0,vpos=50;
    }
    setTimeout("shiftImage()",50);
}
var vpos=0;
function shiftImage(){
    var image=document.getElementById("bucky");
    if(vpos<250){
        vpos+=2;
        image.style.top=vpos+"px";
        setTimeout("shiftImage()",50);
    }
}
```



Cha

```
function addItalic(){
    var i=document.createElement("i");
    i.appendChild(document.createTextNode("italic"));
    addParagraph(i);
}
function addBold(){
    var b=document.createElement("b");
    b.appendChild(document.createTextNode("bold"));
    addParagraph(b);
}
function addParagraph(node){
    var p=document.createElement("p");
    p.appendChild(document.createTextNode("Some "));
    p.appendChild(node);
    p.appendChild(document.createTextNode(" text. "));
    document.getElementById("playground").appendChild(p);
}
function clearAll(){
    var d=document.getElementById("playground");
    while(d.childNodes.length>0)
        d.removeChild(d.childNodes[0]);
}
```