

# JavaScript II

---

CSCI 311

SPRING 2017

# Overview

---

Look at more JavaScript functionality

- DOM
- slide show
- preloading images
- pull down menus
- and more!

# Fun Tool for Sharing Ideas

---

## CodePen

Some examples:

- <https://codepen.io/ge1doot/pen/LkdOwj>
- <https://codepen.io/Mamboleoo/pen/eEvJKo>
- <https://codepen.io/BuddyLReno/pen/boGRPO>
- <https://codepen.io/ge1doot/pen/ZyLzjg> \*\*
- <https://codepen.io/Thibka/pen/BRzgOM>
- <https://codepen.io/gabriellewee/pen/ybKxgB>

# Document Object Model

---

DOM gives us ways to access elements of HTML from JavaScript (and other scripts)

- windows, documents, elements, attributes, text, events, style sheets and rules

## DOM tree

- elements in HTML are tree nodes
- organized into hierarchy
- from a node you can traverse down or up
- can add, modify, remove any elements from tree
- can get and set content attributes
- access, change, delete, add style elements

# DOM tree

---

Root is the Document node

- accessible via document or window.document
- for example: `document.getElementById("idname");`

HTML/XML markup elements:

- gives access to elements of the HTML and their attributes and children

innerHTML:

- for non-void elements, setting innerHTML changes the code in that element

Accessing an element's attributes:

- `element.attributes.xyz`
- `element.getAttribute('xyz');`
- `element.setAttribute('xyz', value);`

# Getting elements of the DOM

---

Method to get one element:

- `document.getElementById`

Methods to get arrays of elements:

- `document.getElementsByTagName`
- `document.getElementsByClassName`

# Ways to access the DOM

---

## Element Properties:

- attributes
- classList
- className
- id
- innerHTML
- tagName

# Example: TextNodes.html

---

```
var p=document.getElementById('para');
var tx=p.firstChild;
alert(tx.data);
alert(tx.wholeText);
```

```
<body style="margin: 50px" onload="showme()">
  <section>
    <h1>Text Nodes</h1>
    <p id="para">Here is an <strong>example</strong> of
      <span style="color: blue">marked up text</span>
      in a paragraph.</p>
  </section>
</body>
</html>
```



# Example: DomCalc.html

---

```
<section><h1>DOM Calculator</h1>
  <p>Simply type in a string such as
    <code>32 + 98 * 6</code> and click GO.</p>
  <p><strong>COMPUTE : </strong>
    <input required="" id="uin"
      placeholder="(20 - 12) * 6" maxlength="30" />
      &#xA0;&#xA0;
    <input id="go" value="GO" type="button"
      onclick="comp('uin')" />
  </p>
  <p id="par">And the answer is:
    <span id="ans" class="ans">48</span></p>
</section>
```

# document methods

---

write or writeln

createElement

createTextNode

getElementById

getElementsByTagName

getElementsByName

getElementsByClassName (can pass comma-separated list of classes)

# Generating New Content

---

Build a data-recording calculator

A smart Form

A sorting shopping cart

Demos:

- CalcSession
- DomSort

# Building elements dynamically

---

A couple more demos to build a list of selections with JavaScript

Demo:

- [DynamicList.html](#)
- [Selections.html](#)

# Preloading Images

---

Quick response to mouseover may require the image to be preloaded

To avoid delays, images may be able to be preloaded:

```
function loadImage(url)
{ if (document.images) // if browser supports images
  { img = new Image(); // obtains a new image object
    img.src = url;    // downloads the image
    return img;
  }
}
```

Script with above placed in head

triggered by the onload event in the body element

# Slideshow

---

We can easily make a slideshow using the click event

- Example: Slides.html, slide.css, slide.js
- Example: slideshow.html

# Example: Conversion Calculator

---

Two inputs of type number

- onfocus event triggers the method call: `reset()`

a button

- onclick event triggers `convert()` method call

`convert.js`

# Pull-down menus with JavaScript

---

Pull-down menu example will use:

- horizontal navbar
- html fragment file with one or more menus. Each is a nav element with child links
- style sheet for navbar and menus
- JavaScript file with pull-down and pull-up actions

Table-styled page layout

Demos:

- Pulldown.html
- MyMenu.html



# Pull-down Menu

---

## Horizontal navbar

- lines 19-27
- 5 links:
  - Main, Service, and Contact are conventional anchor elements <a>
  - Activity and Location are spans with onclick events, and onmouseover events that call functions in menuup.js

## The JavaScript part:

- menus only work as mouseovers if activated
- to be activated, they must be clicked
- to deactivate, click an activated menu item
- only one pull-down shows at a time
  - in down method, any current menu is pulled up first

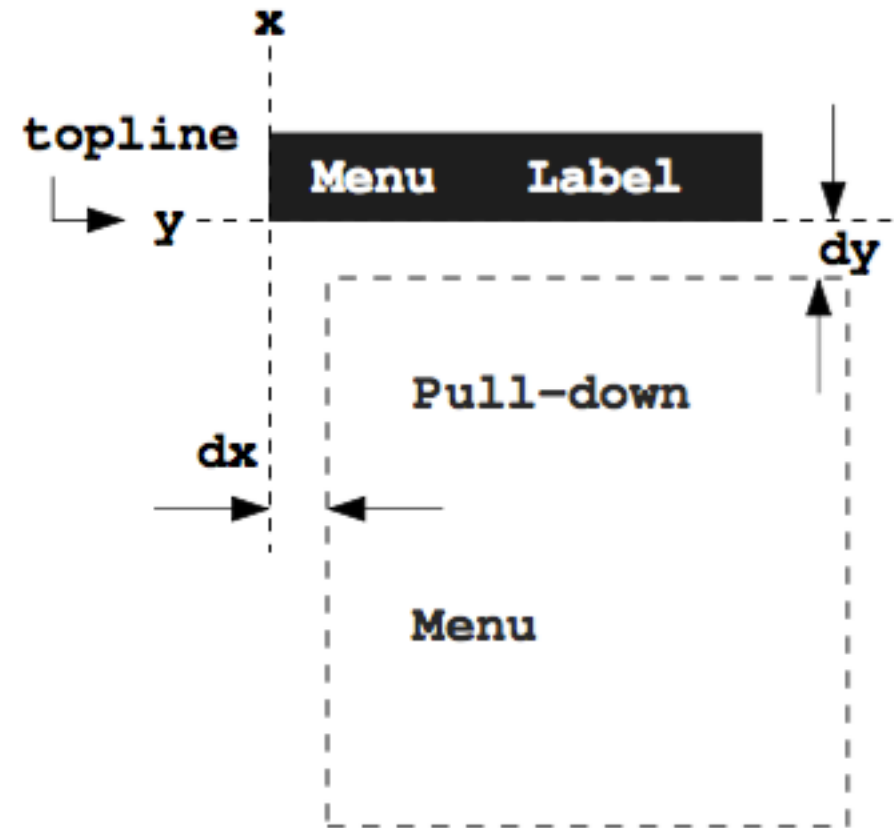
# Pull-down menu

---

## Offsets:

- y offset is set relative to context box below
  - set in init method
- x position is from the menu item itself
  - xPosition method goes up the *offset parent* chain
  - sums the offsets
  - in this case it is the x value of the menu + the offset of the parent (body with offset of 0)
- in our case dx and dy are both 0

**FIGURE 6.7:** Pull-Down Menu Positioning



# Summary

---

Seen more advanced JavaScript that can be used to implement

- user interaction
- dynamic pages
- dropdown menus
- slide shows