JavaScript, jQuery, and the Document Object Model

INFO/CS 2300:
Intermediate Web Design and Programming

2 sheets on half wall

Debugging

```php
<?php
//For scalar variables
echo "<p>Some Variable: $some_variable</p>";


//For arrays and objects
echo '<pre>' . print_r( $_POST, true) . '</pre>';
?>
```

FTP from off Campus

You need a VPN

Search engine: Cornell VPN

The DOM

Document Object Model

HTML

One (useful) way of thinking about an HTML document is as a "tree" of "nodes".

We also use family language such as parents, children, siblings.

```html
<html>
    <head>
        <title>The DOM</title>
    </head>
    <body>
        <div id="content">
            <div id="myimg">
                <img src="tree.gif" alt="HTML Tree" />
            </div>
            <p>We can think of HTML as organized like <a
                href="tree.html">a tree</a> or a family.
            </p>
        </div>
    </body>
</html>
```
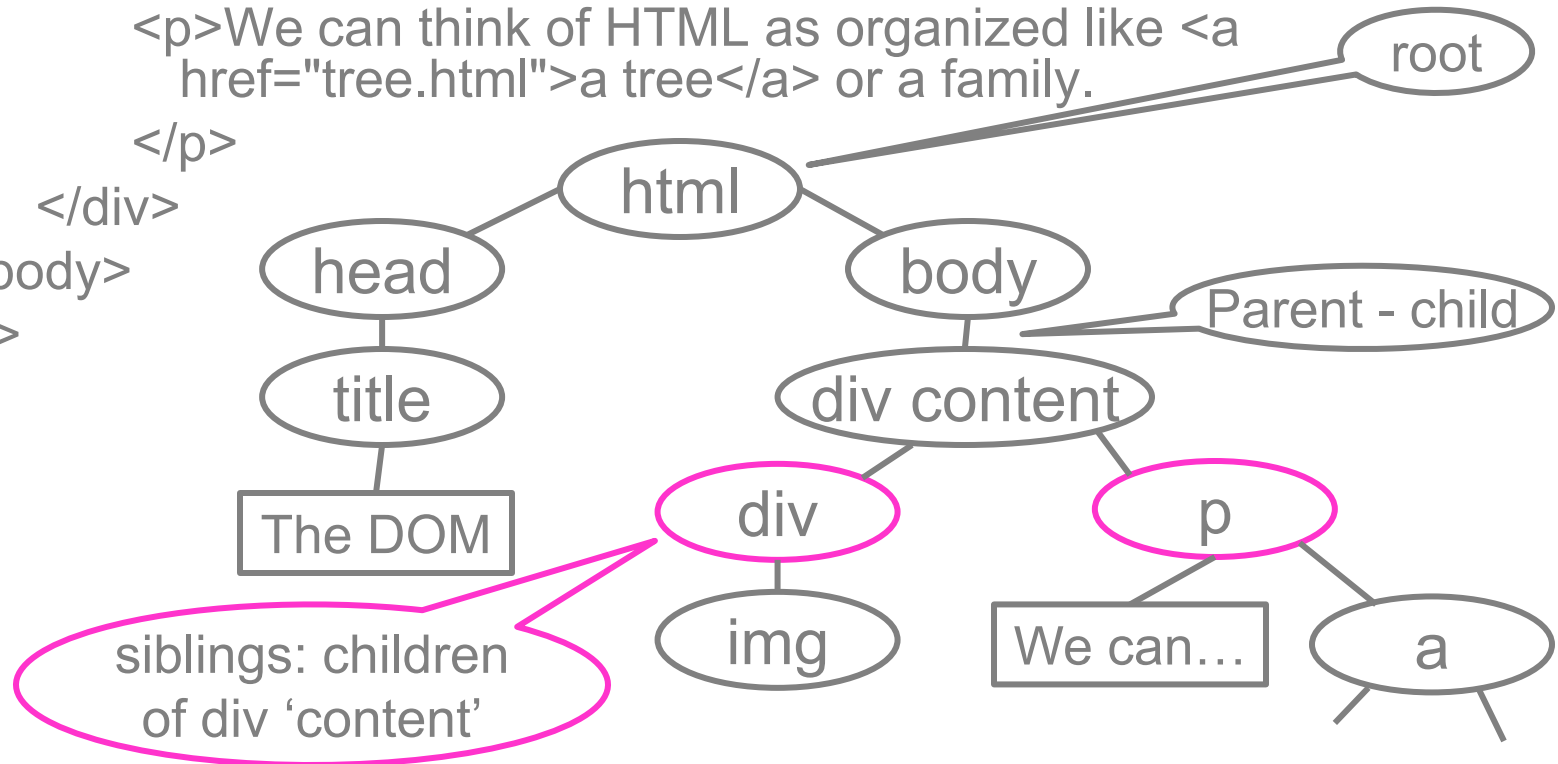
root

html

head   body

Parent - child

title   div content

The DOM   div   p

siblings: children
of div 'content'

img   We can…   a

Everything is an object

Your browser represents every node in the tree as a JavaScript object. The attributes of each tag are properties/fields of the object; the objects have methods that allow easy manipulation.

The top-level object is "document".

Accessing a node

The document object has methods "getElementById" and "getElementsByTagName".

E.g.
   var node = document.getElementById("photoid");
   var nodearray =
       document.getElementsByTagname("img");
   var nodearray =
       document.getElementsByClassName('someClass');

jQuery

jQuery is the most popular JavaScript library in use today

- So popular that the term "Vanilla JavaScript" emerged for plain JavaScript
- Simplifies node access
- Simplifies common tasks
- Unifies code across different browsers

Getting jQuery

http://jquery.com/download/

Current versions 1.12.0 or 2.2.0

Compressed (min) version is fine unless you want to read the code

2.x doesn't support IE 6-8 and that's fine

http://www.w3schools.com/browsers/browsers_explorer.asp

Using jQuery

```
<!DOCTYPE html>
<html>
  <head>
      <meta charset="UTF-8">
      <title>JavaScript Document Write</title>
      <script src="js/jquery-1.12.0.min.js" ></script>
  </head>
  <body>
  </body>
</html>
```

Using jQuery from a CDN

```
<script
    src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0
    /jquery.min.js"></script>
```

If you source from a Content Delivery Network such as Google, the browser may already have the file cached.

Accessing nodes via jQuery

Pretty much any kind of CSS selector will do.
 These select everything that matches

$("p")

$(".myclass")

$("div .header")

$(".mylist li")

$("#myid")

And more

$("[attribute='value']")

$(":checkbox")

$(":checked")

$(":even")

$(":first")

$("parent > child)

https://api.jquery.com/category/selectors/

Saving nodes in jQuery

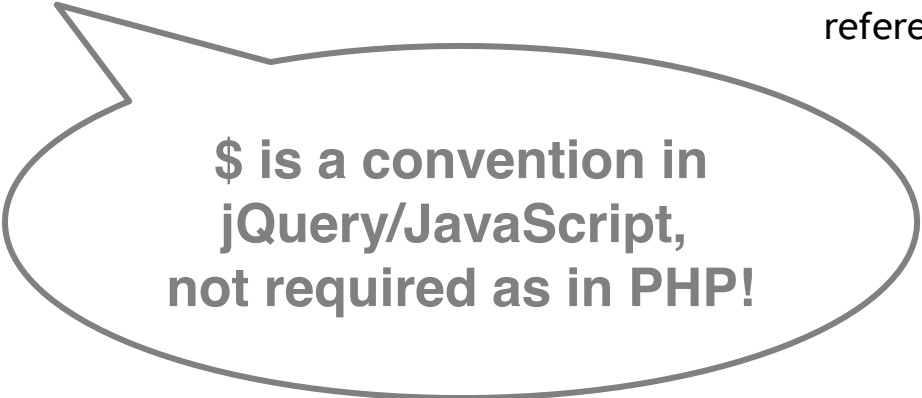You can save nodes accessed via jQuery in variables.

$ is part of the jQuery syntax

var $myNode = $("#tableid");

referencing a jQuery node use "$"

$ is a convention in jQuery/JavaScript, not required as in PHP!

Access other nodes

$myNode.parent()
gives the parent of $myNode

$myNode.prev()
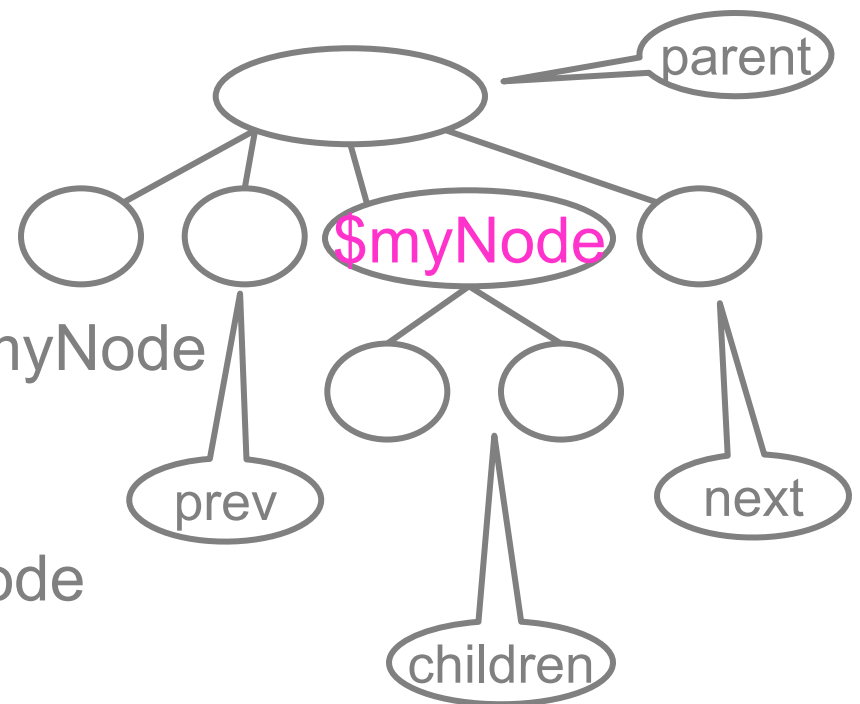gives the previous sibling of $myNode

$myNode.next()
gives the next sibling of $myNode

$myNode.children()
gives an array of all the children of $myNode (in order)

Nodes have properties

$myImgNode.attr('src')
the src atttribute of an image node

$myNode.text()
the string of all the text contained in $mynode

$myTextInput.val()
the string of text entered in a text input

$myNode.css('property_name')
The value of the nodes CSS property given by
    property_name

Changing values

$myImgNode.attr('src', newImgFile)
sets the src of $myImgNode to *newImgFile*.

$myTextNode.text(message)
sets the text of $myTextNode to *message*.

$myNode.html(newHtml)
sets the html of $myNode to *newHtml*.

Changing styling

$myNode.css('property', 'value')
sets the CSS *property* of $myNode to *value*.
Ex: $myNode.css('background-color', '#000000')

$myNode.addClass('className')
CSS in the stylesheet is instantly applied.

Click in!

Responding to Events

$("#myImgNode").click(myFunction);
$(".myImgNodes").click(myFunction);

$("#myImgNode").on("click", myFunction);

Also: .mouseover, .submit, .mouseout, etc.

Anonymous functions

You can write an event handler directly without using a named function by using *anonymous* functions.

E.g.

```
$( "#myNode" ).click( function () {
      alert( "Testing myNode.click" );
});
```

Debugging JavaScript

```
$( "#myNode" ).click( function () {
  console.log("Testing myNode.click");

  var $myNode = $( "#myNode" );
  console.log(  $myNode );
});
```

Chaining

Can string several method calls together.

E.g.

$("#p1").css("color", "red").slideUp(2000);

hide() and show()

```
function myFunction() {
    $("#p1").css( "display", "none" );
    $("#p1").css( "display", "" );
}
//Can be written as
function myFunction() {
    $("#p1").hide();
    $("#p1").show();
}
```

Using the DOM: an example

The msg function example

In the form checker we showed last week, we used a function msg(idname, message) to alter the text in a table.

```
<table>
   <tr>

       <td>Your name:</td>
       <td><input type="text" id="name"
           onchange="validName( this.value );">
       <td id="namemsg">Required</td>
   </tr>
   <tr>

       <td>Your zip code:</td>
       <td><input type="text" id="zip"
               onchange="validZip( this.value );">
       <td id="zipmsg">Required</td>
   </tr>
</table>
```
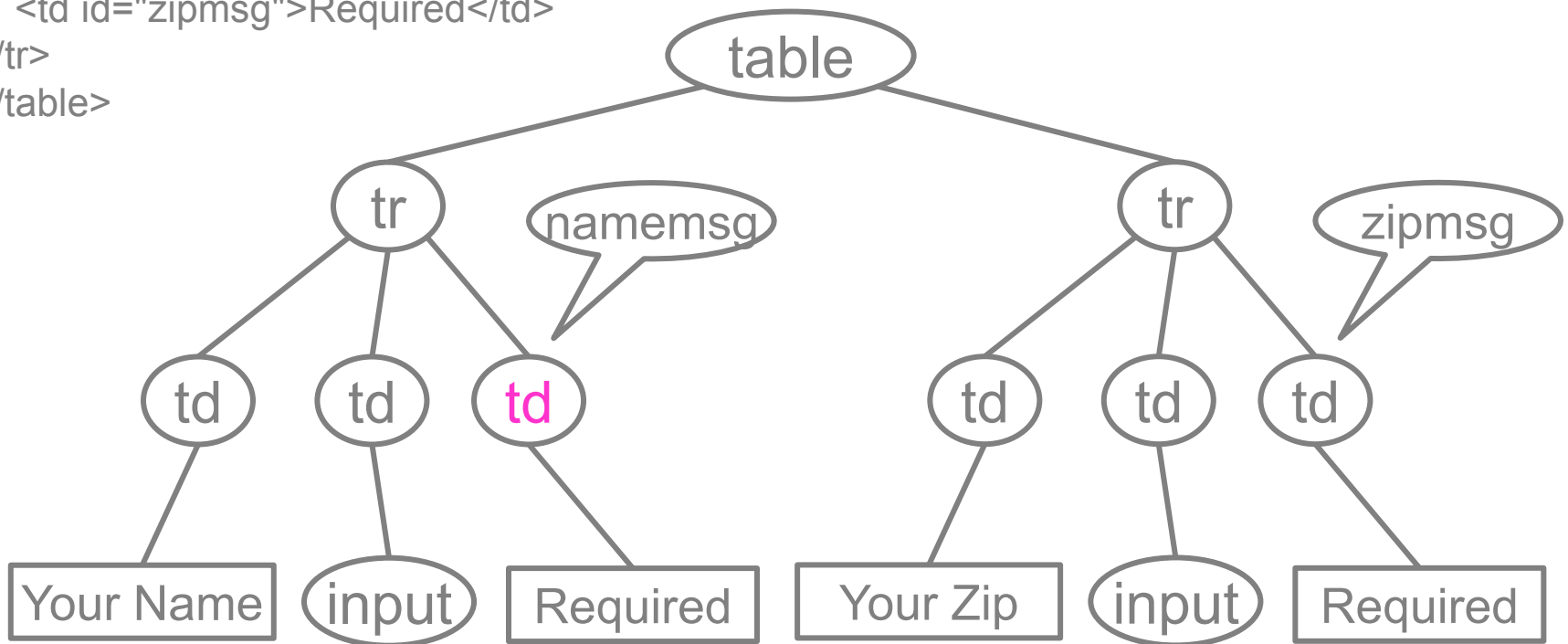
remember "this"

```
<table>
<tr>
    <td>Your name:</td>
    <td><input type="text" id="name" onchange="validName(this.value);" /></td>
    <td id="namemsg">Required</td>
</tr>
<tr>
    <td>Your zip code:</td>
    <td><input type="text" id="zip" onchange="validZip(this.value);" /></td>
    <td id="zipmsg">Required</td>
</tr>
</table>
```

Msg

Write a message to the node with the given id

Now write the code for the msg function.

function msg(idname, message) {

```
        if (trim(message) == "") {
            // Set message to non-breakable space
            message = String.fromCharCode(160);
        }
        $("#" + id).text(message);
```

}

Altering the document

Altering the document

The DOM also has functions that allow us to alter the markup by creating new nodes, moving nodes around, etc.

Putting new nodes in the tree

$node.append($othernode)

adds *othernode* as the last child below node. $othernode can also be HTML.

$node.prepend($othernode)

adds *othernode* as the first child below node. $othernode can also be HTML.

An example

The booklist example files are on your course server accounts under lecture06

Making this work

We'll walk through the effects a step at a time.

```html
<h2> My Cart </h2>
<p id="cartcontent">Your cart is currently empty.</p>

<table id="carttable">
<thead>
<tr id="carthead"><th class="pict"></th><th class="title">Title</th><th class="author">Author</th><th class="price">Price</th>
</tr>
</thead>
<tbody id="cartbody">
</tbody>
</table>

<h2> Books </h2>
<p> Click on a book to add it to your cart. </p>

<table id="availtable">
<thead>
<tr id="availhead"><th class="pict"></th><th class="title">Title</th><th class="author">Author</th><th class="price">Price</th>
</tr>
</thead>
<tbody id="availbody">
<tr class="book">
          <td class="pict"><img id="php" src="php.jpg" /></td>
          <td class="title">Learning PHP 5</td>
          <td class="author">David Sklar</td>
          <td class="price">$29.95</td>
</tr>
<tr class="book">
          <td class="pict"><img id="bpa" src="bpa.jpg" /></td>
          <td class="title">Bulletproof Ajax</td>
          <td class="author">Jeremy Keith</td>
          <td class="price">$34.99</td>
</tr>
```
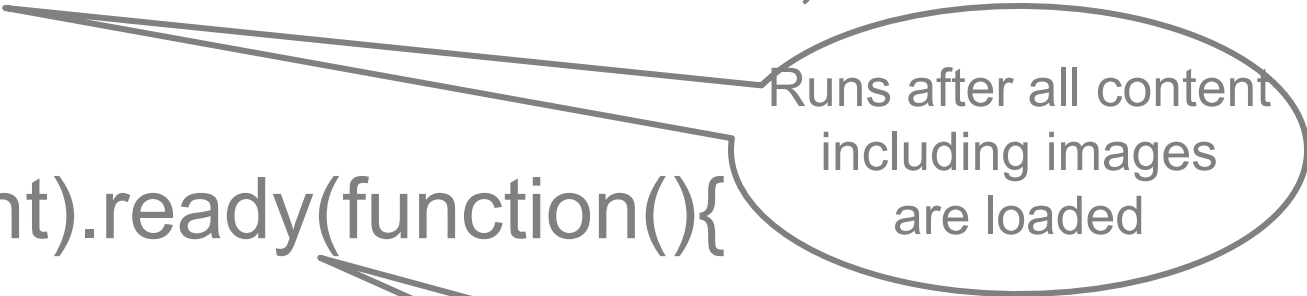
Adding the event handlers

```
$(document).ready(function(){
    //When the mouse is over a row with the book class an
    anonymous function adds a "highlighted" class

    //When the mouse leaves any book rows, an anonymous
    function removes the "highlighted" class

    //Set the moveRow function for all rows so that when the
    row is clicked, the function runs

    //Initialize the empty cart
    cartitems = 0;
    $("#carthead").hide();
});
```
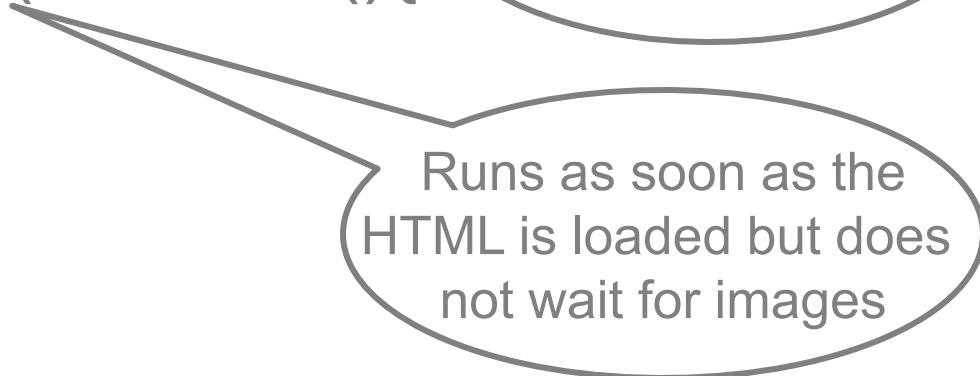
How do we get the js to run?

Use one of these in the books.js file

window.onload = someFunction;

$(document).ready(function(){
   //do stuff
});

Runs after all content including images are loaded

Runs as soon as the HTML is loaded but does not wait for images

Review

- Everything on your webpage is an object. You can use the properties and methods of the objects to:
  - Change properties (styling, text, event handlers)
  - Add new nodes
  - Move nodes

- Project 2 due Tuesday Feb 23 at 5 pm