# Introduction to HTML & CSS

Dr. Justin F. Brunelle
jbrunelle@cs.odu.edu
http://www.cs.odu.edu/~jbrunelle/cs518

# Topics

- Basic structure, DOCTYPEs, and semantics
- HTML Forms
- CSS Basics

- *Not comprehensive!*
  - W3C and Stack Overflow are excellent resources

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My Web Page</title>
    <link rel="stylesheet" type="text/css" href="style.css">
  </head>
  <body>
    <header>
     <h1>My Web Page</h1>
     <h2>Home to my stuff</h2>
    </header>
    <nav>
     <ul>
      <li><a href="#section1">Section 1</a></li>
      <li><a href="#section2">Section 2</a></li>
      <li><a href="#section3">Section 3</a></li>
     </ul>
    </nav>
    <section id="section1">
     <h2>Welcome to Section 1!</h2>
     <p>A section of content!</p>
    </section>
  </body>
</html>
```

```
 <head>
    <meta charset="utf-8">
    <title>My Web Page</title>
    <link rel="stylesheet" type="text/css"
href="style.css">
  </head>
```

- Defines page content's encoding
- Location to supply metadata about web page
- Reference to stylesheets (CSS), external or inline

```
<body>
  <header>
    ...
</body>
```

- Defines main content of the page
- Use semantic markup

# Web Standards & Semantic Markup

- Content/Structure (HTML), style (JavaScript), behavior (CSS)
- Use CSS for position, color, etc.
  - Only tabular data in tables!
- Align the purpose of the content/structure with an appropriate tag
  - e.g., <nav> for navigational items
- Validate your HTML/CSS

# Forms

- method - refers to HTTP method (RFC 2616)
- action - destination of submitted form
- name - variable name used for value of input
- type - UI element type

```
<form method="post" action="login.php">
    <label for="username">Username:</label>
    <input type="text" id="username" name="username" />
    <label for="password">Password:</label>
    <input type="password" id="password"
name="password" />
    <input type="submit" />
</form>
```

# Form elements

- <input type="text" id="username" name="username" />
- <select name="state" id="state">
  <option value="virginia">Virginia</option>
  <option value="maryland">Maryland</option>
  <option value="newyork">New York</option>
  </select>
- <input type="radio" name="sex" value="male" id="sex_male" checked>
  <input type="radio" name="sex" value="female" id="sex_female">



# Other form elements

- <input type="reset" value="Clear" />
  - Clears all set inputs in the form
- <input type="hidden" name="to" value="weigle" />
  - Used for hiding values to be passed from the user
- <input type="checkbox" name="fries" value="wantsFries" />Fries
  <input type="checkbox" name="coke" value="wantsCode" />Coke
  - For selecting multiple items
- textarea, button, select, etc.

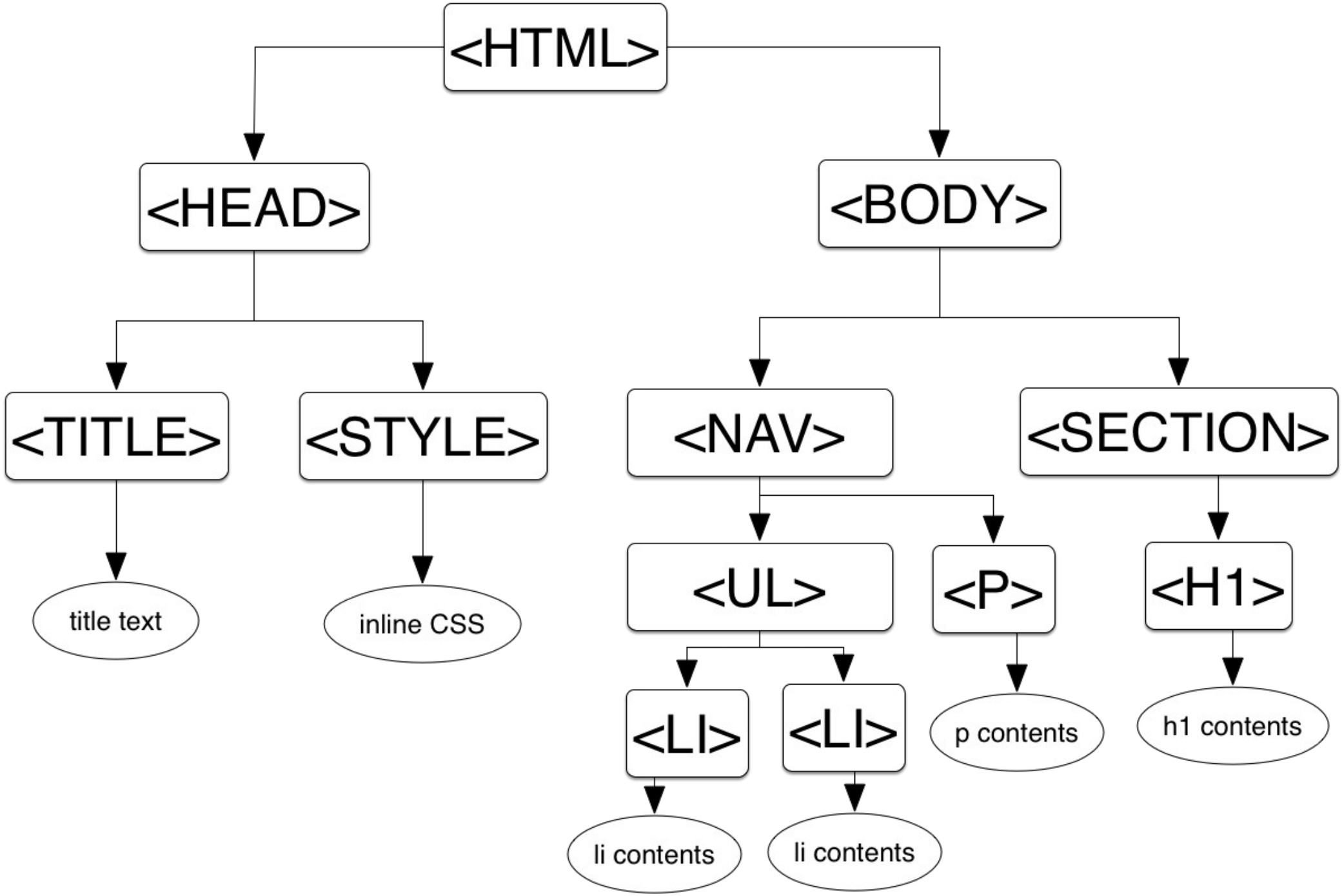
# Document Object Model (DOM)

- Defines STRUCTURE of a representation
- Tree-based
- Basis for specifying elements in CSS/JavaScript
- CSS:

```
p {background-color: blue;}
section > h1 {font-size: 24px;}
```

- JavaScript:

```
var navs = document.getElementsByTagName("nav");
var mainNav = nav[0];
var secondItem = mainNav.childNodes[1];
```


<HTML>
<HEAD>
<BODY>
<TITLE>
<STYLE>
<NAV>
<SECTION>
title text
inline CSS
<UL>
<P>
<H1>
<LI>
<LI>
p contents
h1 contents
li contents
li contents

# DOM Attributes

- HTML definition -- id

  ```
  <p id="myParagraph">Lorem Ipsum</p>
  ```

- CSS association

  ```
  #myParagraph {color: red;}
  ```

- JavaScript reference

  ```
  var myP = document.getElementById('myParagraph');
  ```

# Grouping by class

- HTML definition

  ```
  <p class="specialElements">Lorem Ipsum</p>
  <div class="anotherGroup">Another group's contents</div>
  <span class="specialElements">Dolor Sit Amet</span>
  ```

- CSS association

  ```
  .specialElements {text-decoration: underline;}
  ```

- JavaScript reference

  ```
  var mySpecialElements =
  document.getElementsByClassName('specialElements');
  var secondSpecialElement = specialElements[1]; //note the array
  syntax
  ```

# Cascading Style Sheets (CSS)

- Provides style to documents
- Defined remotely or inline
  - Remotely in a separate file

```
<link rel="stylesheet" type="text/css" href="myStyle.css" />
```

  - Inline within the HTML document

```
<style type="text/css">
    p {color: red;}
    p#myUniqueParagraph {font-family: 'Times', 'Georgia', sans-serif;}
    .myGroupedElements {color: green;}
    p.myGroupedElements {color: red;}
</style>
```

# Order of CSS Operations

inline style > id > class > element

| | | |
|---|---|---|
| `<p>Hello</p>` | `p {color: red;}` | Hello |
| `<p class="test">Hello</p>` | `p {color: red;}`<br>`p.test {color blue; background-color: yellow;}` | Hello |
| `<p class="test" id="myP">Hello</p>` | `p {color: red;}`<br>`p.test {color blue; background-color: yellow;}`<br>`p#myP { background-color: orange;}` | Hello |
| `<p class="test" id="myP"`<br>`style="color: white;">Hello</p>` | `p {color: red;}`<br>`p.test {color blue; background-color: yellow;}`<br>`p#myP { background-color: orange;}` | Hello |
| `<p style="color: white;" id="myP">`<br>`Hello <span>World</span>`<br>`<em>!</em>`<br>`</p>` | `p#myP { background-color: orange;}`<br>`p span {background-color: purple}` | Hello World! |