



# **Web Development**

## **COMP 431 / COMP 531**

### **Lecture I: HTML**

**Instructor: Mack Joyner**

**Department of Computer Science, Rice University**

[mjoyner@rice.edu](mailto:mjoyner@rice.edu)

<http://www.clear.rice.edu/comp431>

# Goal: Multi-User Web Application

- **Full Stack Development**
  - **Front-End**
    - HTML, CSS, JavaScript, Framework
  - **Back-End**
    - Web servers, Databases, Security
- **Software Engineering Design**
  - **Test Driven Development**
    - Unit testing,
    - End-to-end testing
  - **Practice good design principles** (e.g., *encapsulation*)

# Final Web Application

# Ricebook

Email or Phone


Password

☐ Keep me logged in

[Log In](#)

[Forgot your password?](#)


## Connect with friends and the world around you on Ricebook



**See photos and updates** from friends in News Feed.



**Share what's new** in your life on your Timeline.



**Find more** of what you're looking for with Facebook Search.

## Sign Up

It's free and always will be.

**Birthday**

Month ▼

Day ▼

Year ▼

[Why do I need to provide my birthday?](#)

☐ Female ☐ Male

By clicking Sign Up, you agree to our [Terms](#) and that you have read our [Data Policy](#), including our [Cookie Use](#).

**Sign Up**

[Create a Page for a celebrity, band or business.](#)

# Assumptions/Expectations

- No prior HTML/CSS/JavaScript experience
- Familiar with basic unix commands (e.g., *ssh* and *scp*)
- Motivated to learn about web development and design

# Administration

Class Sessions:	Tu/Th 2:30-3:45 DH 1070
Instructor:	Mack Joyner
TAs:	Qinlan, Oliver, James, Sishan, Haoyuan
Course web site:	<a href="http://www.clear.rice.edu/comp431">http://www.clear.rice.edu/comp431</a>
Office Hours:	See course web site
Announcements:	on Piazza (be sure you have signed up)
Grades:	Canvas

# Acknowledgement

Course material adapted from previous version of course taught by  
Scott Pollack

# Grade Breakdown

Item	Count	COMP 43 I	COMP 53 I
In-class Exercises	24	15%	15%
Assignments (not including final)	7	60%	50%
Final Web Application	1	15%	15%
Quizzes	5	10%	10%
Paper/Presentation (COMP 53 I only)	1		10%

# In-Class Exercises

- We will have one each class session
- Based on material presented during the lecture
- Grade rubric
  - 1.0 = Submit before end of class
  - 0.5 = Submit before beginning of next class



# Assignments

*due Thursdays by 11:59pm*

- Requirements driven
- Graded on correctness, style, and completeness
- Rubric for each assignment

## Paper/Presentation by COMP531

- Essay on topic related to web development
- 5 minute in class presentation

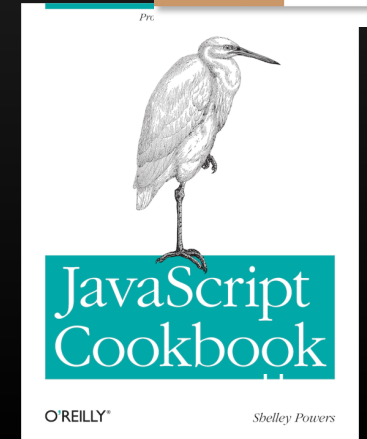
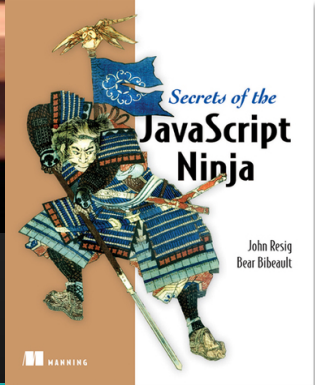
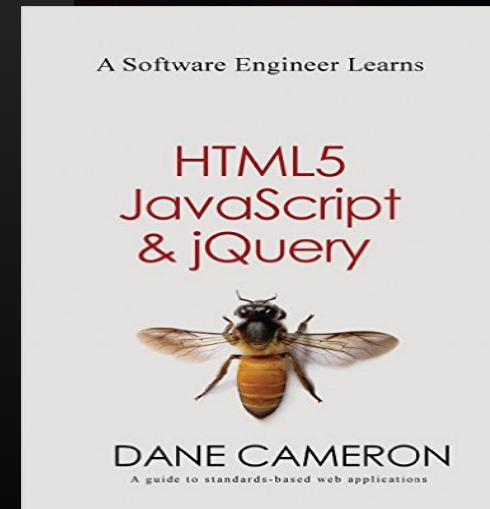
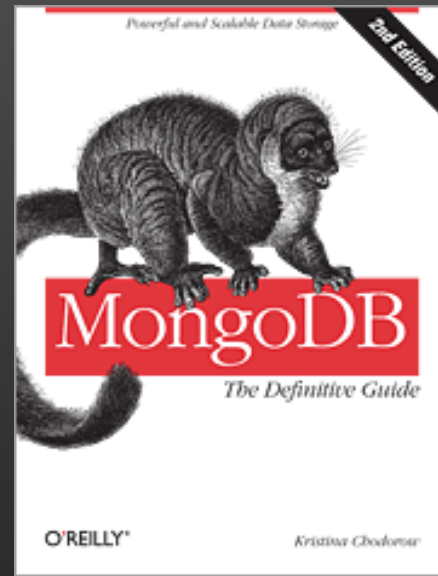
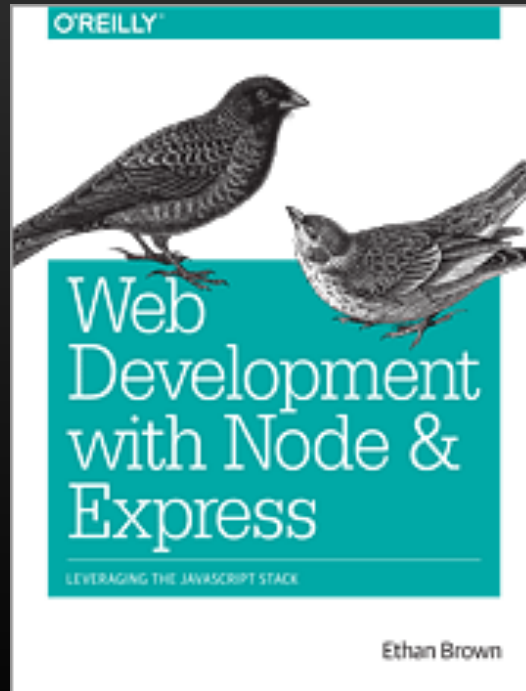
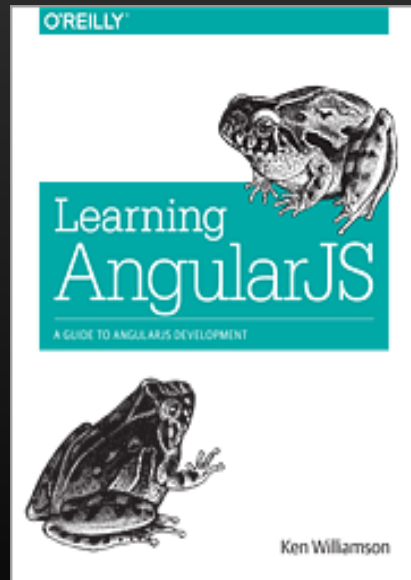
1. Simple HTML Page
2. Dynamic Page
3. JavaScript Game
4. Draft Front-End
5. Front-End Application
6. Draft Back-End
7. Integrated Web App
8. Final Full Web Application

# Late Assignment Policy

- You will have 3 slip days
  - When you use a slip day, you'll have an additional 24 hours to turn in assignment
  - Use README to indicate when you are using a slip day
- No late submissions (other than those using slip days) will be accepted

# Textbooks

- No required texts
- Some relevant texts...



Fall 2017

# Course Outline

HTML		AngularJS (2)	Hosting
Forms		Unit Testing	Databases
JavaScript		Mocking	Authentication
Scope		End-to-End	Security
Events		<i>Break</i>	Third-Party Auth
jQuery		Web Servers	Scalability
Modern JS		Back End	Service APIs
MVC		REST	Integrating
AngularJS (1)		Unit Testing	Presentations

# Advice

## Course covers lots of material

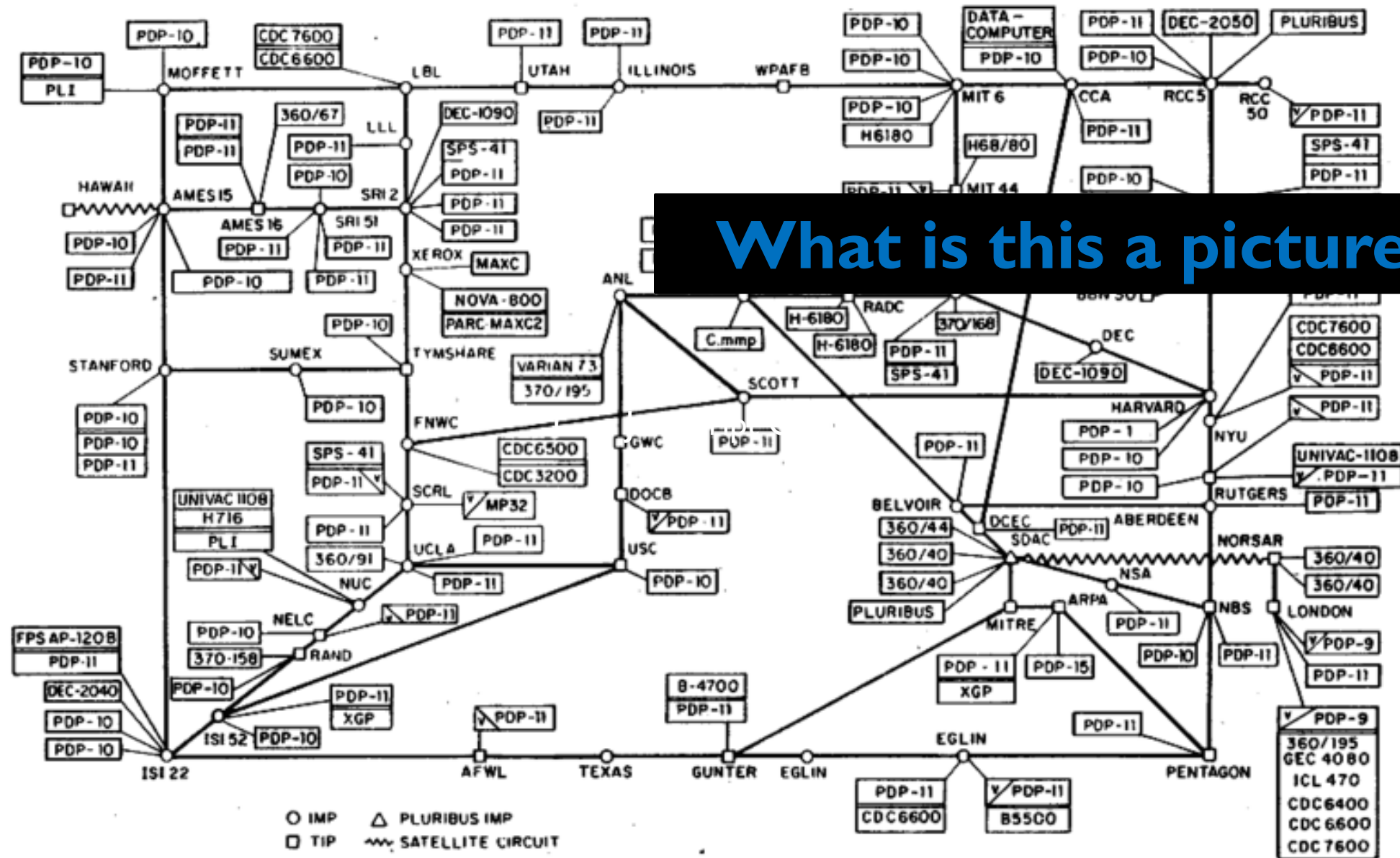
- Start assignments early
- Ask for help often
- Check to see if your question was already asked on Piazza

# Ready?

Hope you enjoy the course!

# HyperText Markup Language

HTML



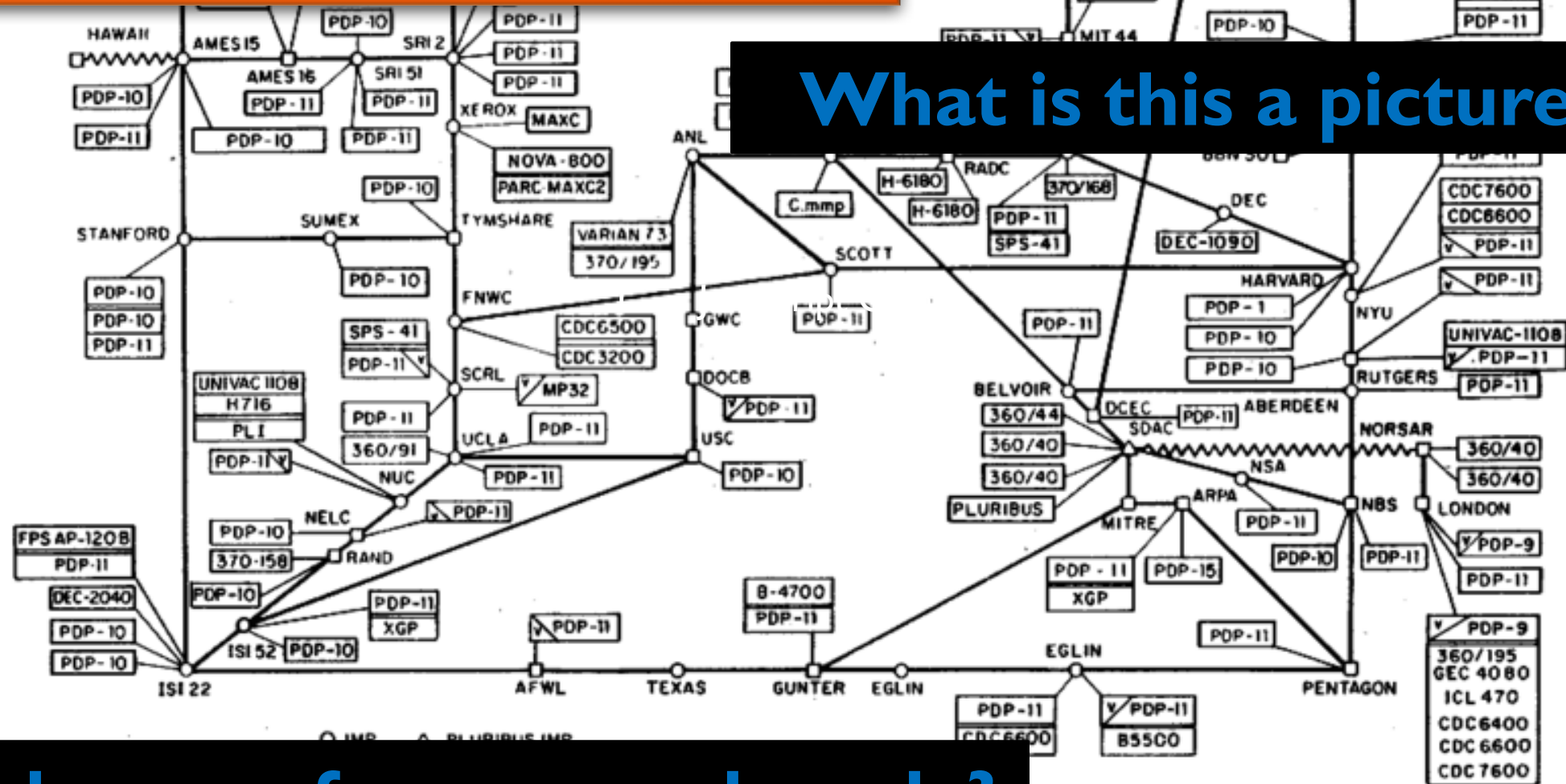
What is this a picture of?

(PLEASE NOTE THAT WHILE THIS MAP SHOWS THE HOST POPULATION OF THE NETWORK ACCORDING TO THE BEST INFORMATION OBTAINABLE, NO CLAIM CAN BE MADE FOR ITS ACCURACY)

NAMES SHOWN ARE IMP NAMES, NOT (NECESSARILY) HOST NAMES



# Networking the Internet



What is this a picture of?

How do we reference each node?

8/22/17

Fall 2017

NAMES SHOWN ARE IMP NAMES, NOT (NECESSARILY) HOST NAMES

# In the beginning...

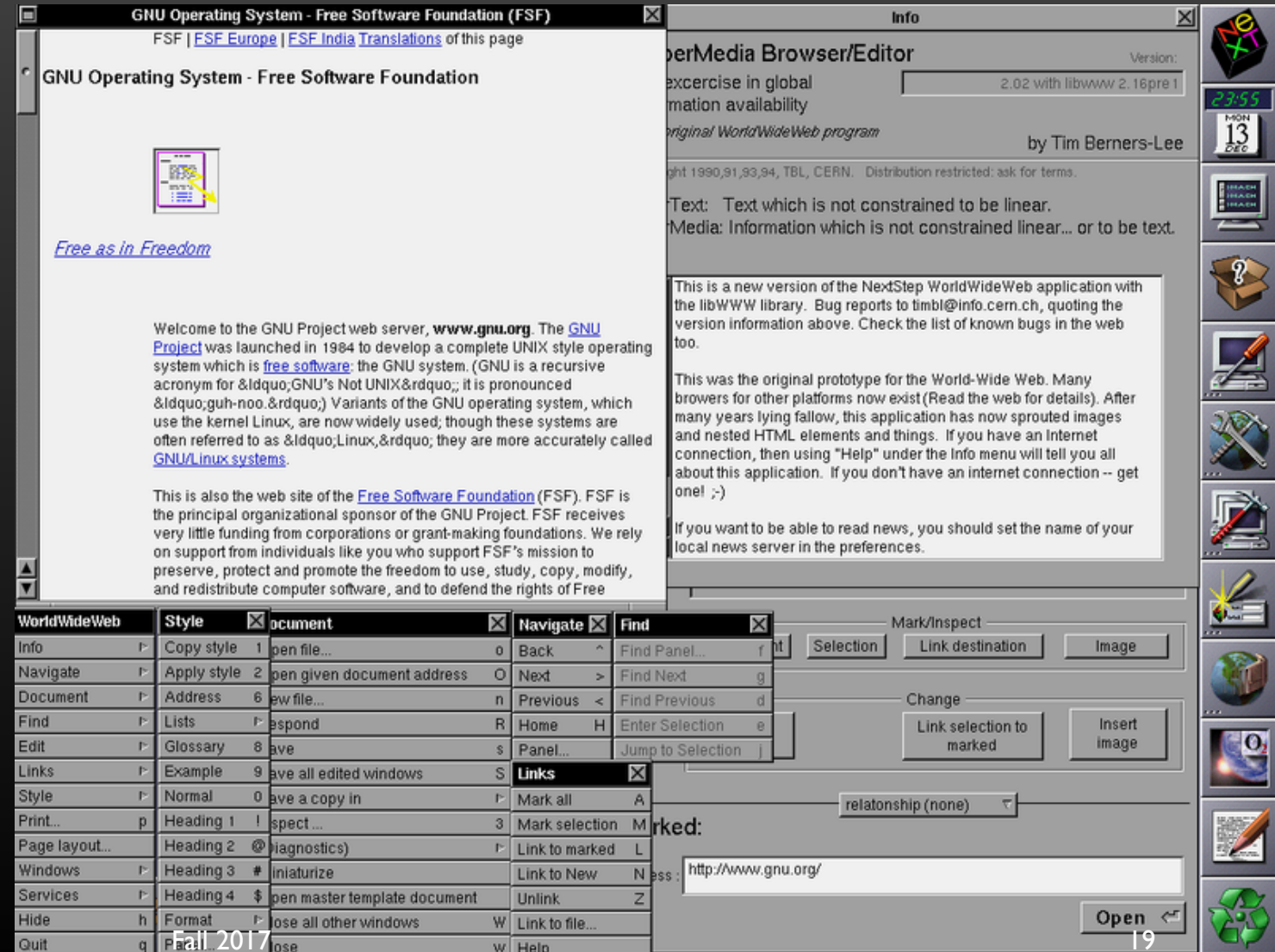
“Imagine, then, the references in this document all being associated with the network address of the thing to which they referred, so that while reading this document you could skip to them with a click of the mouse” [Tim Berners-Lee, March 12, 1989]

- **HyperText** is text with **hyperlink** references to other text that can be immediately accessed
- Generalized to **hypermedia** to include graphics, audio, video, etc.



# World Wide Web

An information space with uniform resource identified (URI) web resources interconnected by hypertext links accessible via the Internet





# A Brief History of HTML

- 1989 – World Wide Web [Nexus]
- 1991 – HTML Tags formally introduced [Lynx '92]
- 1993 – HTML specification (W3C '94) [NCSA Mosaic/Netscape]
- Nov 1995 – HTML 2.0 (JavaScript) [Internet Explorer, Opera]
- Jan 1997 – HTML 3.2
- Dec 1997 – HTML 4.0 (strict/transitional/frameset)
- 2000's – Browser Wars [Mozilla/Firefox, Safari, Camino, Chrome, ...]
- Oct 2014 – HTML5

# Web Servers

- (typically) an **Internet** connected machine “listening” on port 80 (and/or 443)
- Respond to **requests** (typically from a browser)
  - GET, POST, PUT, DELETE, HEAD, TRACE, OPTIONS, CONNECT, PATCH

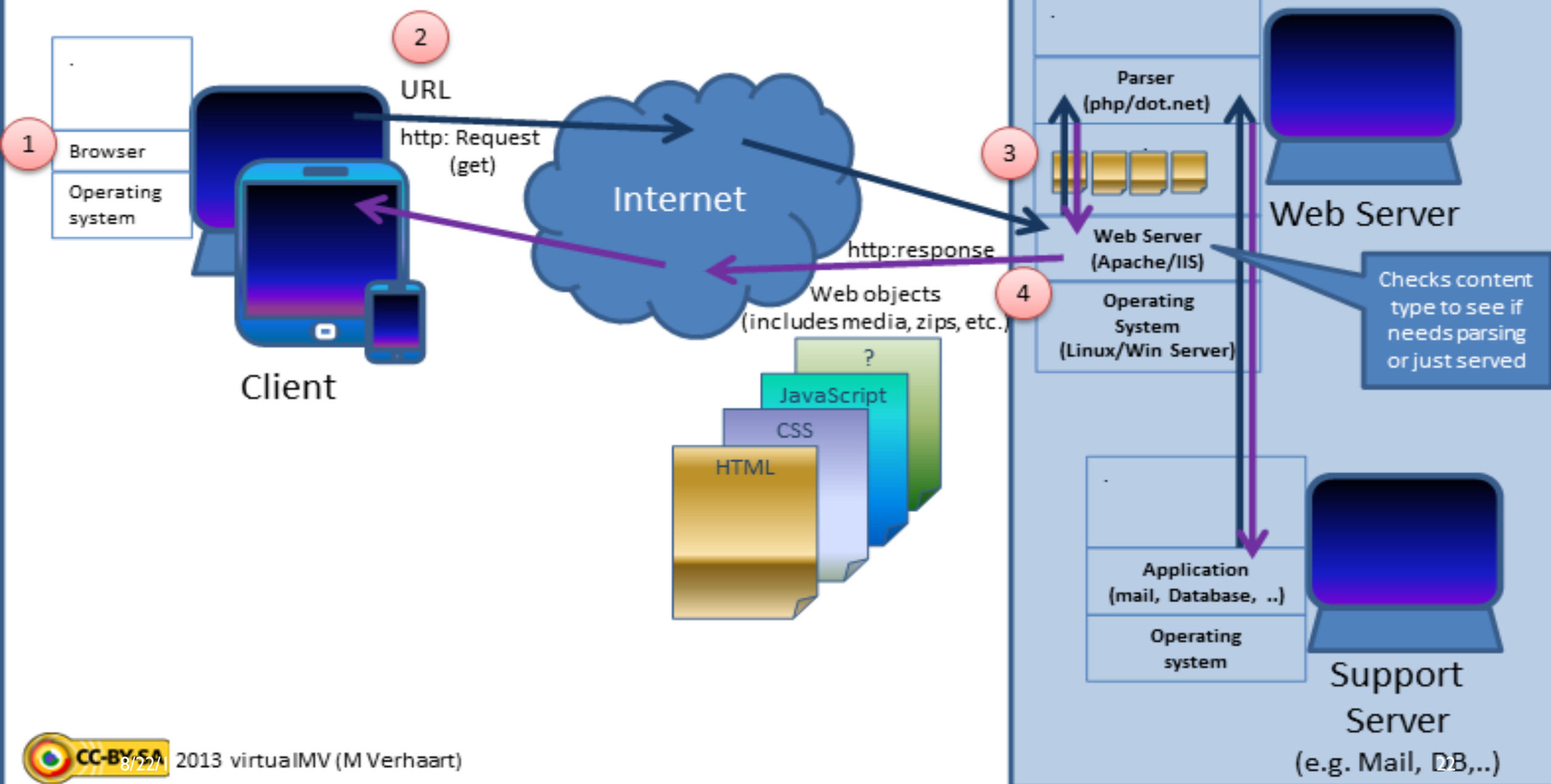
```
GET /index.html HTTP/1.1
Host: www.example.com
```

- “Serves” content
- **Web browser** makes requests and eventually renders a page for the user.

```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
ETag: "3f80f-1b6-3e1cb03b"
Content-Type: text/html; charset=UTF-8
Content-Length: 138
Accept-Ranges: bytes
Connection: close
```

```
<html>
<head>
```

# Client-server ecosystem





# Standard Generalized Markup Language (SGML)

- Case *insensitive*
- Elements/Tags `<example>`
- Open and Close make for well-formed tags `<br/>`
- Attributes `<example tested>`
- Valued Attributes `<example tested="true">`
- Content `<p>Information</p>`
- Processing instruction `<?`
- Document Type Definition `<!`

```
1  <book>
2      <book-title>My First Book</book-title>
3
4      <prologue>
5          This is a book.  This is my first book.
6          It's a bit verbose with all these tags...
7      </prologue>
8
9      <chapter title="Chapter 1">
10         The first chapter was written last.
11
12         <section title="Wow!">
13             <paragraph>
14                 The attributes help some
15             </paragraph>
16         </section>
17
18     </chapter>
19
20 </book>
```

# Document Type Definition (DTD)

- First line of document, defines the document type
  - Instructs the browser to render in “quirks mode” or “standards mode”

```
<!DOCTYPE root-element PUBLIC "FPI" ["URI"] [  
    <!-- internal subset declarations -->  
>
```

- Example

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html lang="ar" dir="ltr" xmlns="http://www.w3.org/1999/xhtml">
```



# HTML5

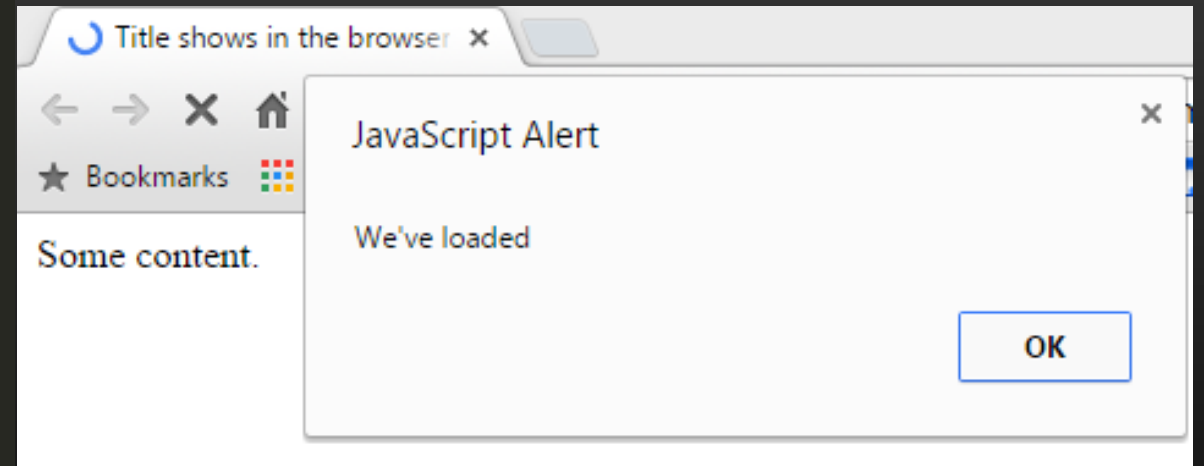
- HTML5 is not a true SGML
  - No DTD associated with HTML5

```
<!DOCTYPE html>
<html lang="en-US">
<!-- language is helpful for screen
      readers and search engines -->
```

- **Forgiveness:** HTML body by default
  - We can omit `<html>`, `<head>`, `<body>` tags to make a page more concise.
  - ... but let's not do that: always write “complete” HTML

# Basic Structure of an HTML page

```
1  <!DOCTYPE html>
2  <html lang="en-US">
3  <head>
24 </head>
25
26 <body onload="doSomething()">
27   Some content.
28 </body>
29 </html>
30
31 <script>
32   function doSomething() {
33     alert("We've loaded");
34   }
35 </script>
36
```



# Markup

```
25 <body>
26
27 <h1>Sample Page</h1>
28
29 <p>Paragraphs are separated as "blocks" of text.</p>
30 <p>Simple markup such as <b>bold face</b> or <em>
   emphasized</em> text.
31
32 Note that white space is ignored.
33
34 If we want to "control" the layout, we should use
   styles. If we need some small control we can
   insert a line break. For example,</p>
35 
36 <br/>
37 
38 <p>Hyperlinks are contained within <a name="anchor">
   anchors</a> which can be referenced within a page
   using another <a href="#anchor" title="go here">
   anchor</a>. In general we want links to go to
   other pages, such as <a href="https://developer.
   mozilla.org">this</a> one.</p>
39 </body>
40 </html>
```

## Sample Page

Paragraphs are separated as "blocks" of text.

Simple markup such as **bold face** or *emphasized* text. Note that white space is ignored. If we want to "control" the layout, we should use styles. If we need some small control we can insert a line break. For example,



Hyperlinks are contained within anchors which can be referenced within a page using another anchor. In general we want links to go to other pages, such as this or go here.

# Inline vs Block

- We markup to control layout. Sometimes we want “blocks” of content that are positioned separately from other blocks.
- Other times we want inline layout or styling

## Inline

`<b>`

`<em>`

`<br>`

`<span>`

## Block

`<p>`

`<div>`

# More HTML fundamentals

```
40 <hr/>
41 <!-- this is a comment -->
42 <table border="1">
43     <thead>
44         <tr><th>Centered by Default</th><th>Column Header</th></tr>
45     </thead>
46     <tbody>
47         <tr><td>Row</td><td>Content</td></tr>
48         <tr><td colspan="2">span two columns to format our table better</td></tr>
49         <tr><td>Lazy...</td></tr>
50         <tr>
51             <td>Here is an ordered list
52                 <ol>
53                     <li>first item</li>
54                     <ol>
55                         <li>nested item</li>
56                     </ol>
57                     <li>another item</li>
58                 </ol>
59             </td>
60             <td>and an unordered list
61                 <ul>
62                     <li>first item</li>
63                     <li>another item</li>
64                 </ul>
65             </td>
66     </tbody>
67 </table>
```

Centered by Default	Column Header
Row	Content
span two columns to format our table better	
Lazy...	
Here is an ordered list <ol style="list-style-type: none"><li>1. first item<ol style="list-style-type: none"><li>1. nested item</li></ol></li><li>2. another item</li></ol>	and an unordered list <ul style="list-style-type: none"><li>• first item</li><li>• another item</li></ul>

We must escape `<` and `>` in addition to `&` and a few other character. Sometimes we want to control the amount of space " " in a cell in a table, we can do that with non-breaking spaces for example.

Pre-formatted text will appear as it is written exactly (we hope)

And it *renders* some HTML!

```
function example() { var foo = "bar"; console.log(foo); }
```

# More HTML fundamentals

```
68  
69 <p>We must escape &lt; and &gt; in addition to &amp; and a few other character.  
Sometimes we want to control the amount of space "&nbsp;&nbsp;&nbsp;" in a cell in a  
table, we can do that with non-breaking spaces for example.</p>  
70 <pre>  
71 Pre-formatted text will appear as it is written  
72 exactly (we hope)  
73 <strong>And it <em>renders</em> some |HTML!</strong>  
74 </pre>  
75 <code>  
76 function example() {  
77     var foo = "bar";  
78     console.log(foo);  
79 }  
80 </code>  
81 <hr/>
```

# More HTML fundamentals

```
1  <!DOCTYPE html>
2  <html lang="en-US">
3  <head>
4      <title>About Me</title>
5
6      <meta charset="utf-8">
7      <meta name="keywords" content="HTML, CSS">
8      <meta name="description" content="Bio information">
9      <meta name="author" content="Mack Joyner">
10 </head>
11
12 <body>
13     <p> I went to Galveston for summer vacation. </p>
14 </body>
15 </html>
16
```



# Some HTML Resources

- W3Schools  
<http://www.w3schools.com/>
- Mozilla Developer Network  
<https://developer.mozilla.org/>
- Validate your HTML  
<http://validator.w3.org/>
- Real-time HTML rendering (includes tidy)  
<http://jsfiddle.net/bpenp4dq/>

# Completing Inclass and HW Assignments

- Checkout an svn repository to use for this class.
  - You only need one for now
  - svn checkout [https://svn.rice.edu/r/comp431\\_531/turnin/F17/<netid>](https://svn.rice.edu/r/comp431_531/turnin/F17/<netid>)
  - Create inclass-I directory
- When you are ready to turn in your assignment
  - svn add <files>
  - svn commit -m <message>
- Commit early and often.