RICE®

Web Development

COMP 431 / COMP 531

Scott E Pollack, PhD

January 12, 2016

Agenda

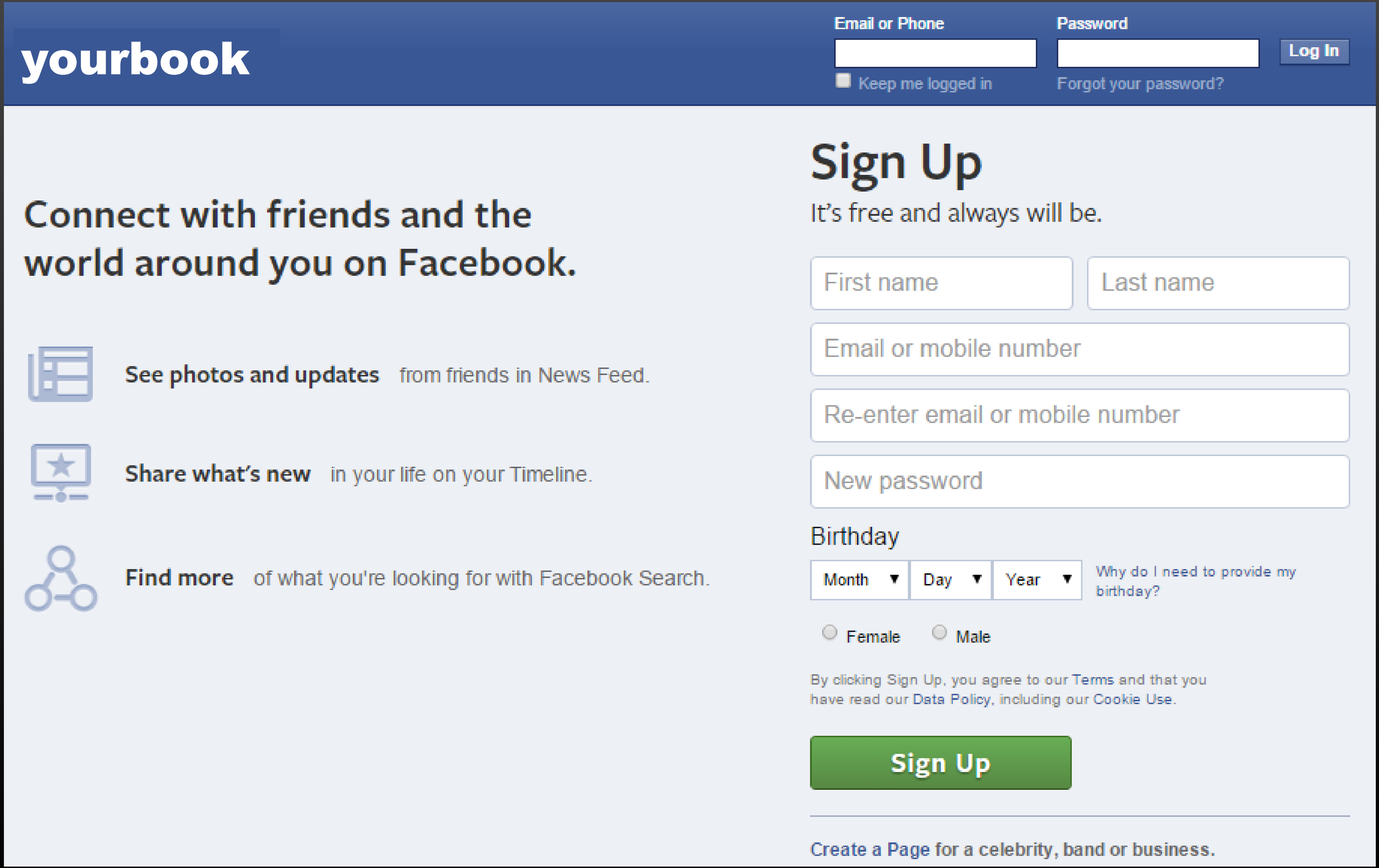
- Introductions
- Administration
- Resources
- Course Outline
- HTML
- In-class Exercise

What is this class about?

Full Stack Development with solid software engineering practices

- **Front-End Development**
  - HTML, CSS, JavaScript, MVC
- **Back-End Development**
  - web servers, databases, serialization, security
- **Result**
  - *near* “Cutting edge” multi-user web application

Final Web App



yourbook

Email or Phone

Password

Log In

□ Keep me logged in

Forgot your password?

Connect with friends and the world around you on Facebook.

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.

First name

Last name

Email or mobile number

Re-enter email or mobile number

New password

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.

Who is this class for?

- No prior HTML/CSS/JavaScript experience
- A desire to learn about web development and design
- Familiar with basic *nix commands (e.g., *ssh* and *git* and *svn*)
- The resilient motivated self-learner

The resilient motivated self-learner

CV in a Nutshell

- UC Berkeley BA Physics and Astrophysics
  - TA’ed introductory course in building a website (ca 1997)
- CU Boulder MS Physics, MS Astrophysics, PhD Physics
  - TA’ed various physics and astronomy courses
  - Taught course on Cosmology
  - Lots of tutoring (Computer Science, Physics, Astronomy, Math)
  - Thesis Focus: *Precision Measurement and Gravitational Astrophysics*
- Center for Experimental Nuclear Physics and Astrophysics, U. Washington
  - Taught course on Modern Physics
- Rice Quantum Institute
- Teaching online since 2008
  - Physics, astronomy, and computer science
- Finance since 2010

2σ TWO SIGMA

Administration

Class Sessions:	Tu/Th 2:30-3:45 Symonds II Lab
Instructor:	Scott Pollack
TA:	Terry Tang
Webpage:	http://www.clear.rice.edu/comp431
Announcements:	on Piazza
	(be sure you have signed up)

Grade Breakdown

Item	Count	COMP 431	COMP 531
In-class Exercises	24	1 → 24	1 → 24
Assignments (not including final)	7	9 → 63	6 → 42
Final Web Application	1	13 → 13	12 → 12
Web Site Surveys (COMP 531 only)	2		5 → 10
Paper/Presentation (COMP 531 only)	1		12 → 12

In-Class Exercises

- We will have one each class session
- No makeup if absent without prior confirmation
- Graded Pass/No-Pass
- To be performed and completed in class
  - But you are allowed to turn in through 2 AM
- Will focus on material presented in lecture

Assignments

- Requirements driven
- Graded on correctness, style, and completeness

Site Reviews by COMP531

- Review and provide user feedback on assignments 3 and 5

Paper/Presentation by COMP531

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

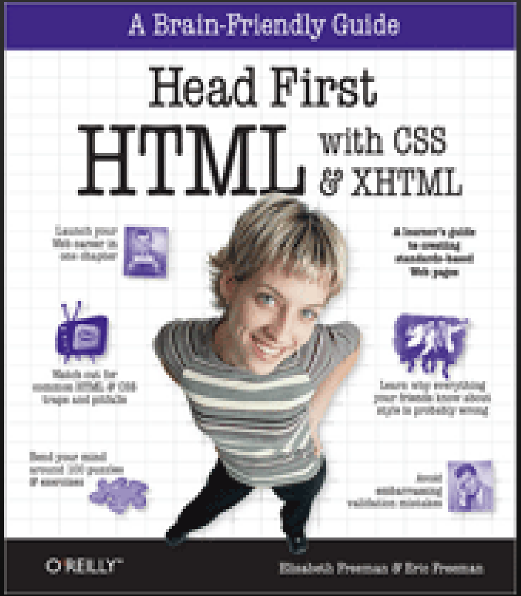
1. Simple HTML Page
2. Dynamic Page
3. Draft Front-End
4. JavaScript Game
5. Front-End Application
6. Draft Back-End
7. Integrated Web App
8. Final Full Web Application

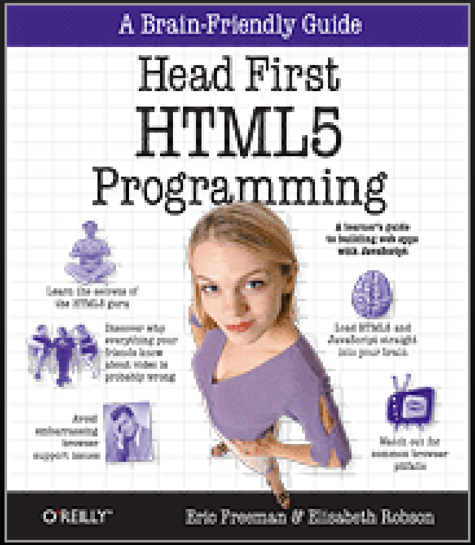
Agenda

- Introductions
- Administration
- Resources
- Course Outline
- HTML
- In-class Exercise

Textbooks

- No required texts
- Some relevant texts…





Office Hours

	Location	Times
Scott Pollack	Symonds II (?)	TR 3:45-4:45PM
Terry Tang	DCH 3113	M 6-8 PM and by appointment

Agenda

- Introductions
- Administration
- Resources
- Course Outline
- HTML
- In-class Exercise

Course Outline

	Topic			Topic			Topic
1/12	HTML		2/11	MVC		3/22	Unit Testing
1/14	JavaScript		2/16	AngularJS		3/24	Databases
1/19	Forms		2/23	Unit Testing		3/29	Authentication
1/21	CSS		2/25	Headless		4/5	Security
1/26	Events		3/1&3	*break*		4/7	Third-Party Auth
1/28	More JS		3/8	Web Servers		4/12	Scalability
2/2	HTML5		3/10	Python		4/14	Service APIs
2/4	Frameworks		3/15	REST		4/19	Integrating
2/9	Libraries		3/17	Hosting		4/21	Presentations

This class covers a **LOT** of material

It goes **REALLY** fast

It does not help that I talk fast…

You may spend a lot of time on the assignments

Start the assignments early

Reach out for help often

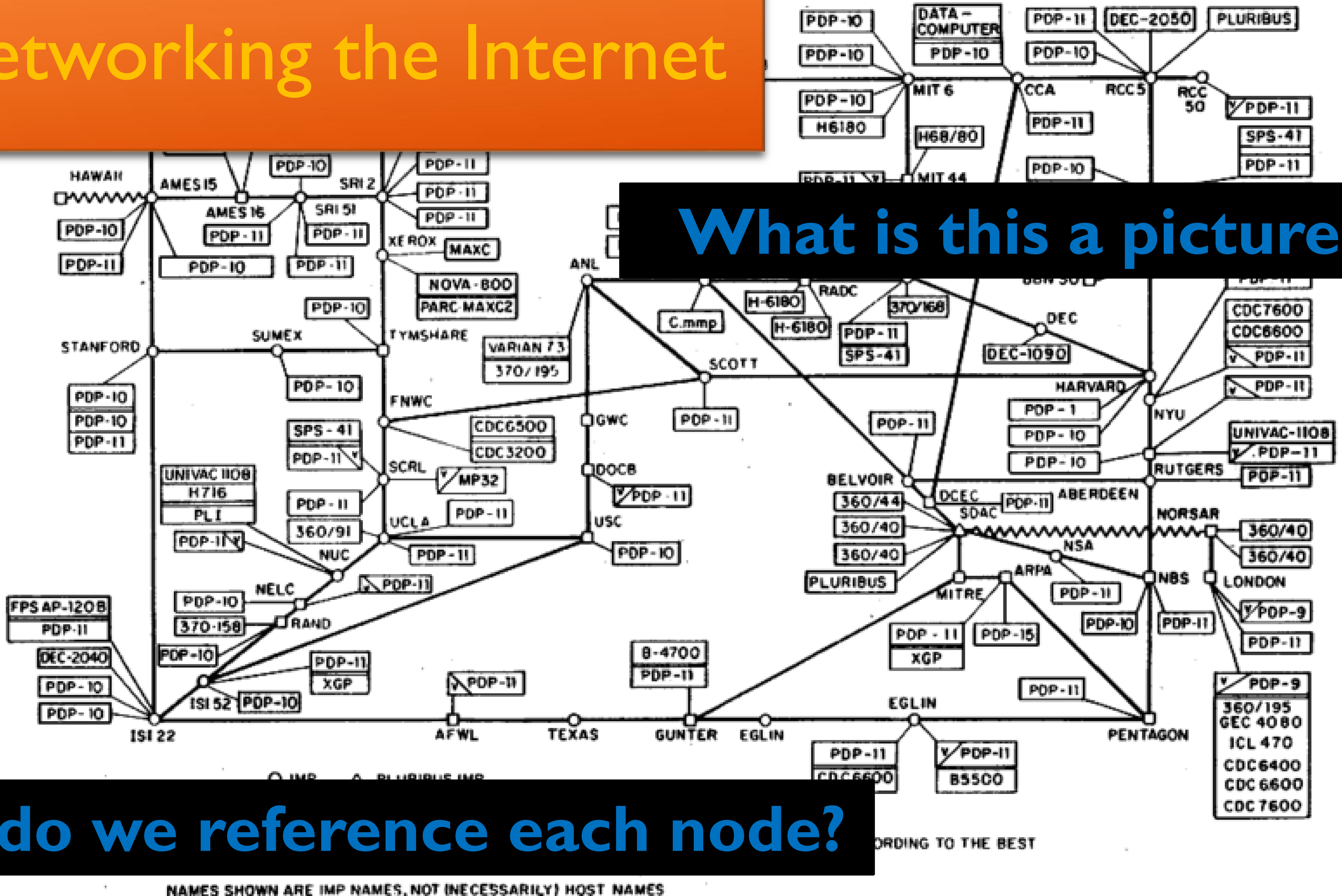
Agenda

- Introductions
- Administration
- Resources
- Course Outline
- HTML
- In-class Exercise

HyperText Markup Language

HTML

Networking the Internet



What is this a picture of?

How do we reference each node?

ORDING TO THE BEST

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

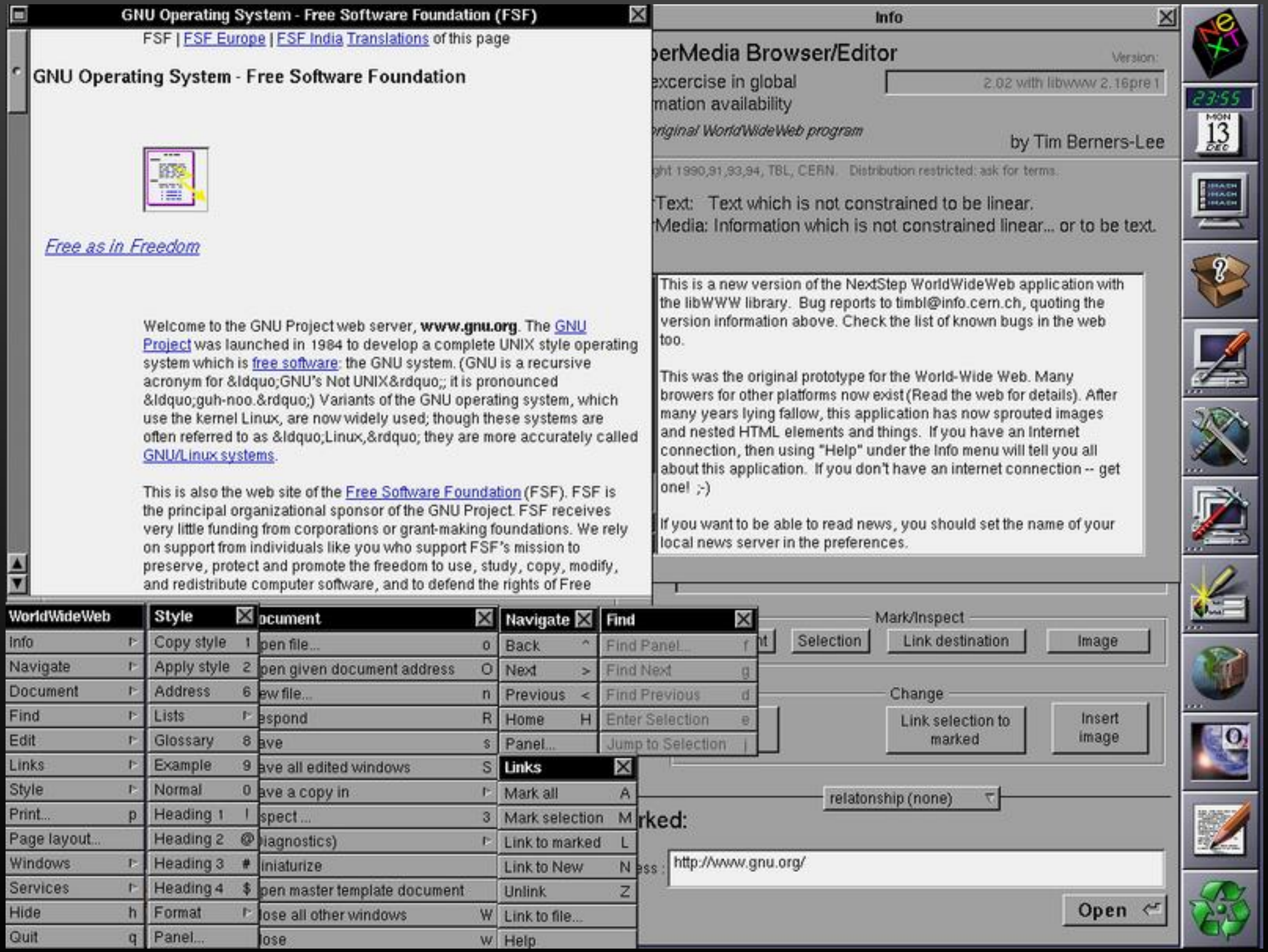
In the beginning…

“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



GNU Operating System - Free Software Foundation (FSF)

FSF | FSF Europe | FSF India Translations of this page

GNU Operating System - Free Software Foundation

Free as in Freedom

Welcome to the GNU Project web server, **www.gnu.org**. The GNU Project was launched in 1984 to develop a complete UNIX style operating system which is free software: the GNU system. (GNU is a recursive acronym for &ldquo;GNU's Not UNIX&rdquo;; it is pronounced &ldquo;guh-noo.&rdquo;) Variants of the GNU operating system, which use the kernel Linux, are now widely used; though these systems are often referred to as &ldquo;Linux,&rdquo; they are more accurately called GNU/Linux systems.

This is also the web site of the Free Software Foundation (FSF). FSF is the principal organizational sponsor of the GNU Project. FSF receives very little funding from corporations or grant-making foundations. We rely on support from individuals like you who support FSF's mission to preserve, protect and promote the freedom to use, study, copy, modify, and redistribute computer software, and to defend the rights of Free

Info

erMedia Browser/Editor

Version: 2.02 with libwww 2.16pre1

excercise in global mation availability

riginal WorldWideWeb program

by Tim Berners-Lee

ght 1990,91,93,94, TBL, CERN. Distribution restricted: ask for terms.

Text: Text which is not constrained to be linear.

Media: Information which is not constrained linear... or to be text.

This is a new version of the NextStep WorldWideWeb application with the libWWW library. Bug reports to timbl@info.cern.ch, quoting the version information above. Check the list of known bugs in the web too.

This was the original prototype for the World-Wide Web. Many browers for other platforms now exist (Read the web for details). After many years lying fallow, this application has now sprouted images and nested HTML elements and things. If you have an Internet connection, then using "Help" under the Info menu will tell you all about this application. If you don't have an internet connection -- get one! ;-)

If you want to be able to read news, you should set the name of your local news server in the preferences.

ess: http://www.gnu.org/

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

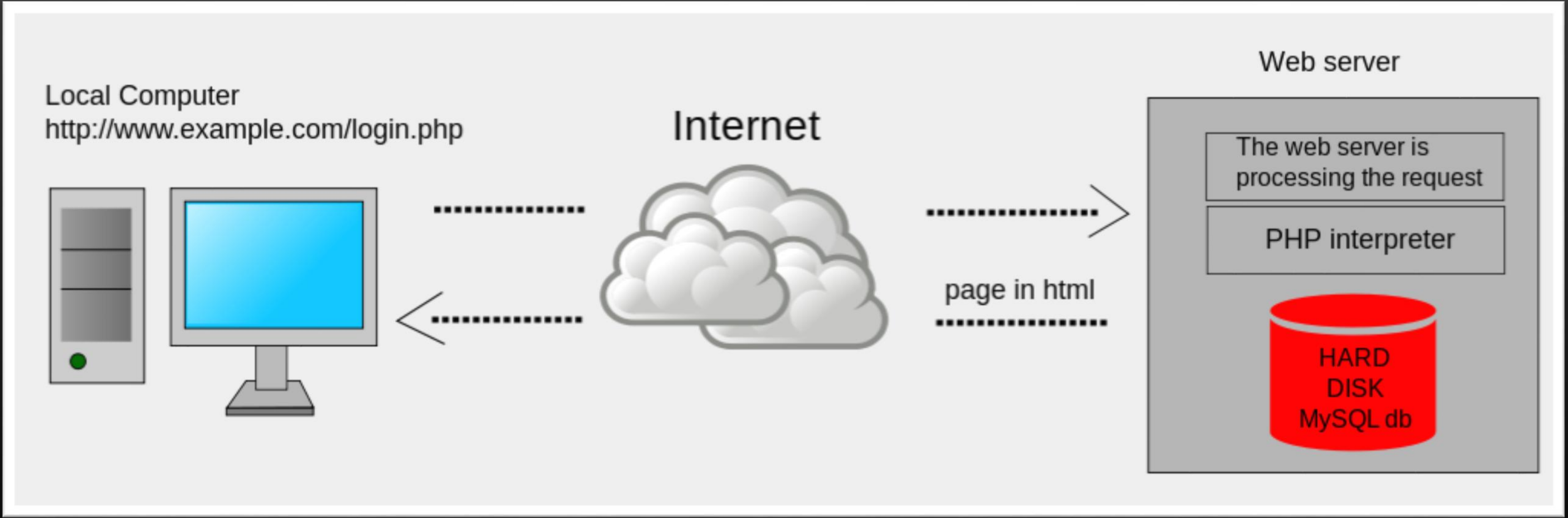
- (typically) an Internet connected machine “listening” on port 80 (and/or 443)
- Respond to requests (typically from a browser)
  - GET, POST, PUT, DELETE, HEAD, TRACE, OPTIONS, CONNECT, PATCH

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

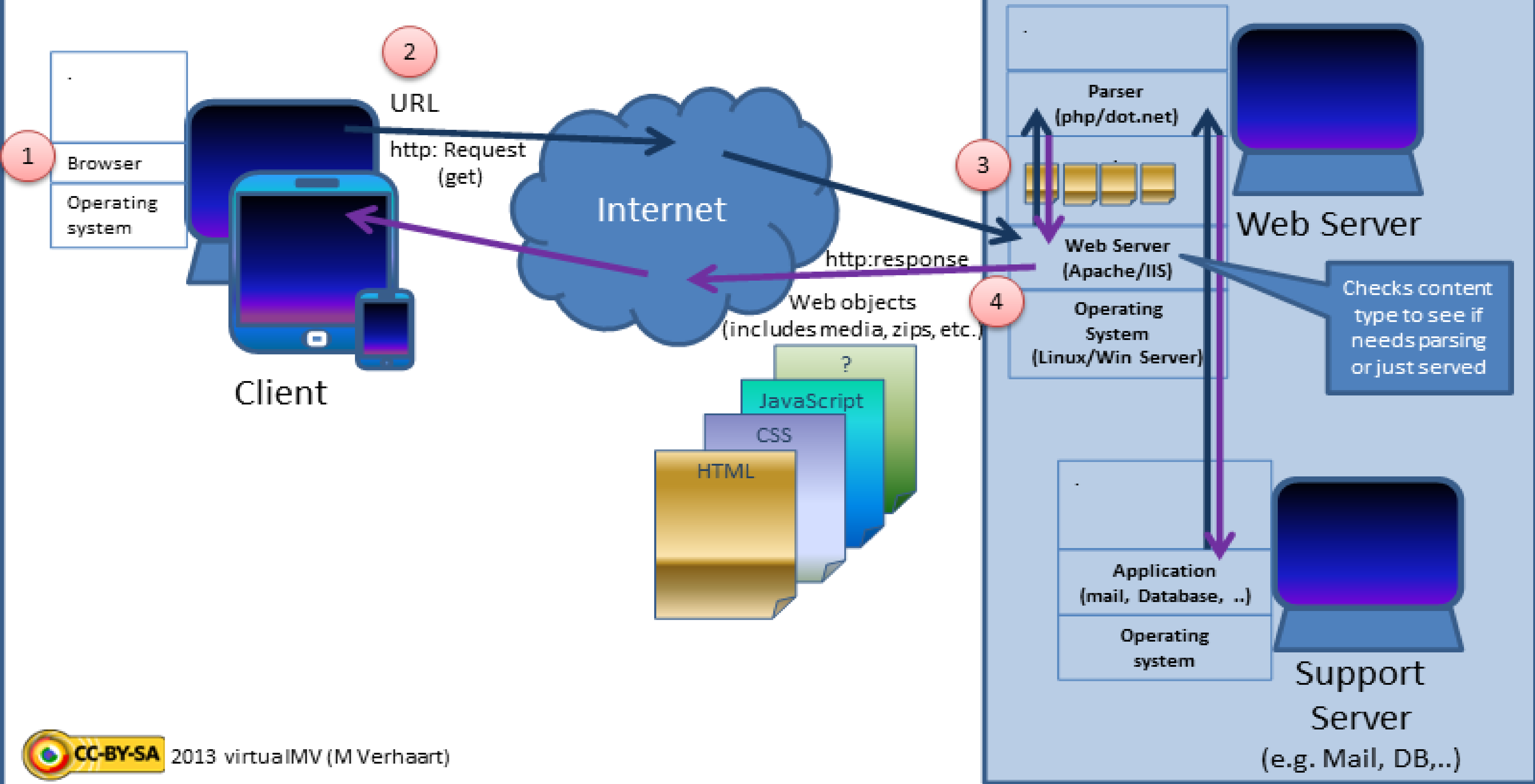
```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
ETag: "3f80f-1b6-3e1cb03b"
Content-Type: text/html; charset=UTF-8
Content-Length: 138
Accept-Ranges: bytes
Connection: close

<html>
<head>
```

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

Static vs. Dynamic



Client-server ecosystem



May be one or many computers

1 Browser

Operating system

Client

2 URL

http: Request (get)

Internet

3

Parser (php/dot.net)

Web Server (Apache/IIS)

Operating System (Linux/Win Server)

Web Server

Checks content type to see if needs parsing or just served

http:response

4

Web objects (includes media, zips, etc.)

?

JavaScript

CSS

HTML

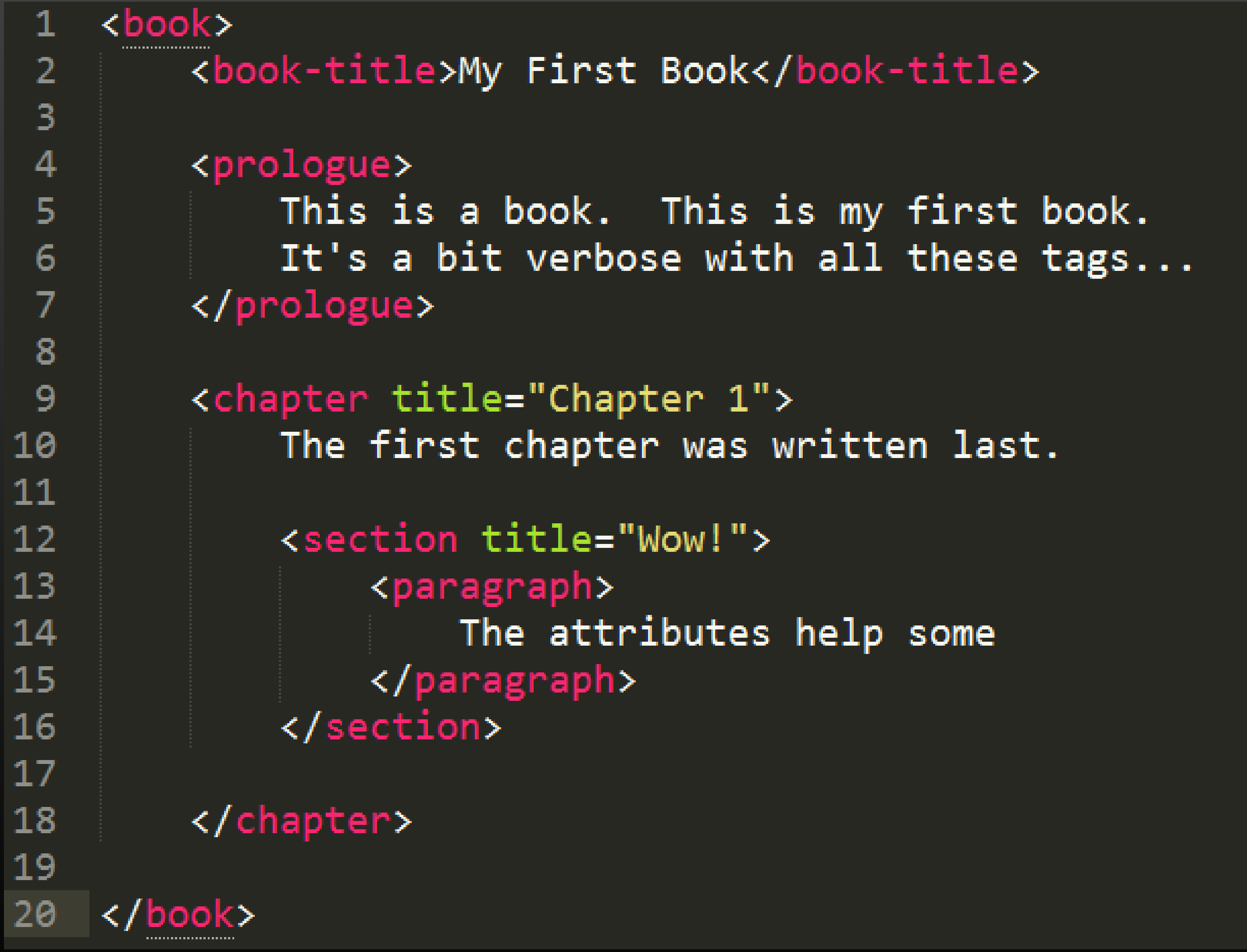
Application (mail, Database, ..)

Operating system

Support Server (e.g. Mail, DB,..)

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 <!



```
<book>
    <book-title>My First Book</book-title>

    <prologue>
        This is a book.  This is my first book.
        It's a bit verbose with all these tags...
    </prologue>

    <chapter title="Chapter 1">
        The first chapter was written last.

        <section title="Wow!">
            <paragraph>
                The attributes help some
            </paragraph>
        </section>

    </chapter>

</book>
```

Document Type Definition (DTD)

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

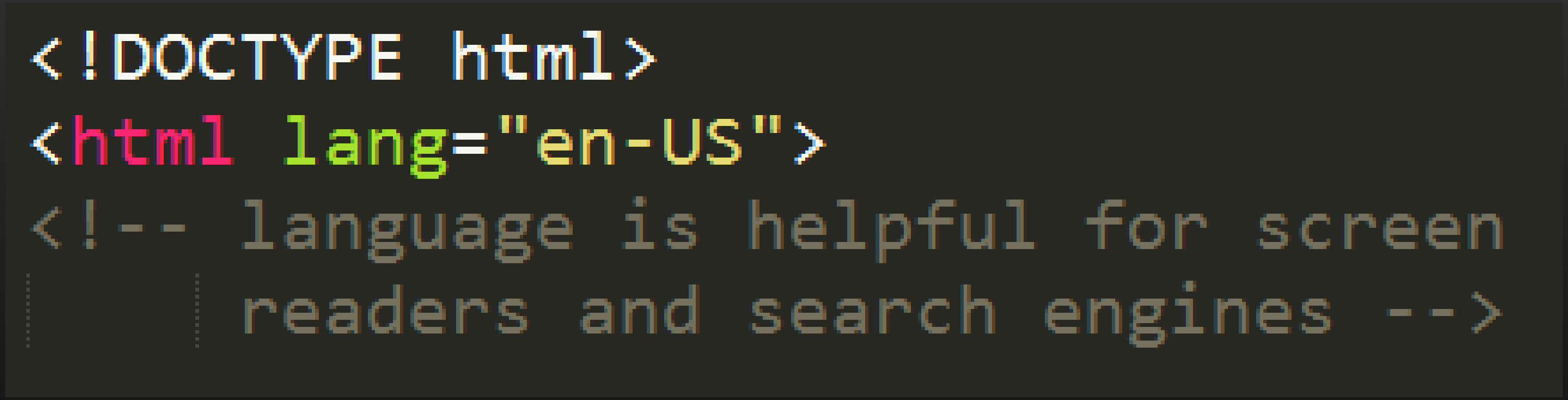
```
<!DOCTYPE root-element PUBLIC "FPI" ["URI"] [
   <!-- internal subset declarations -->
]>
```

- 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

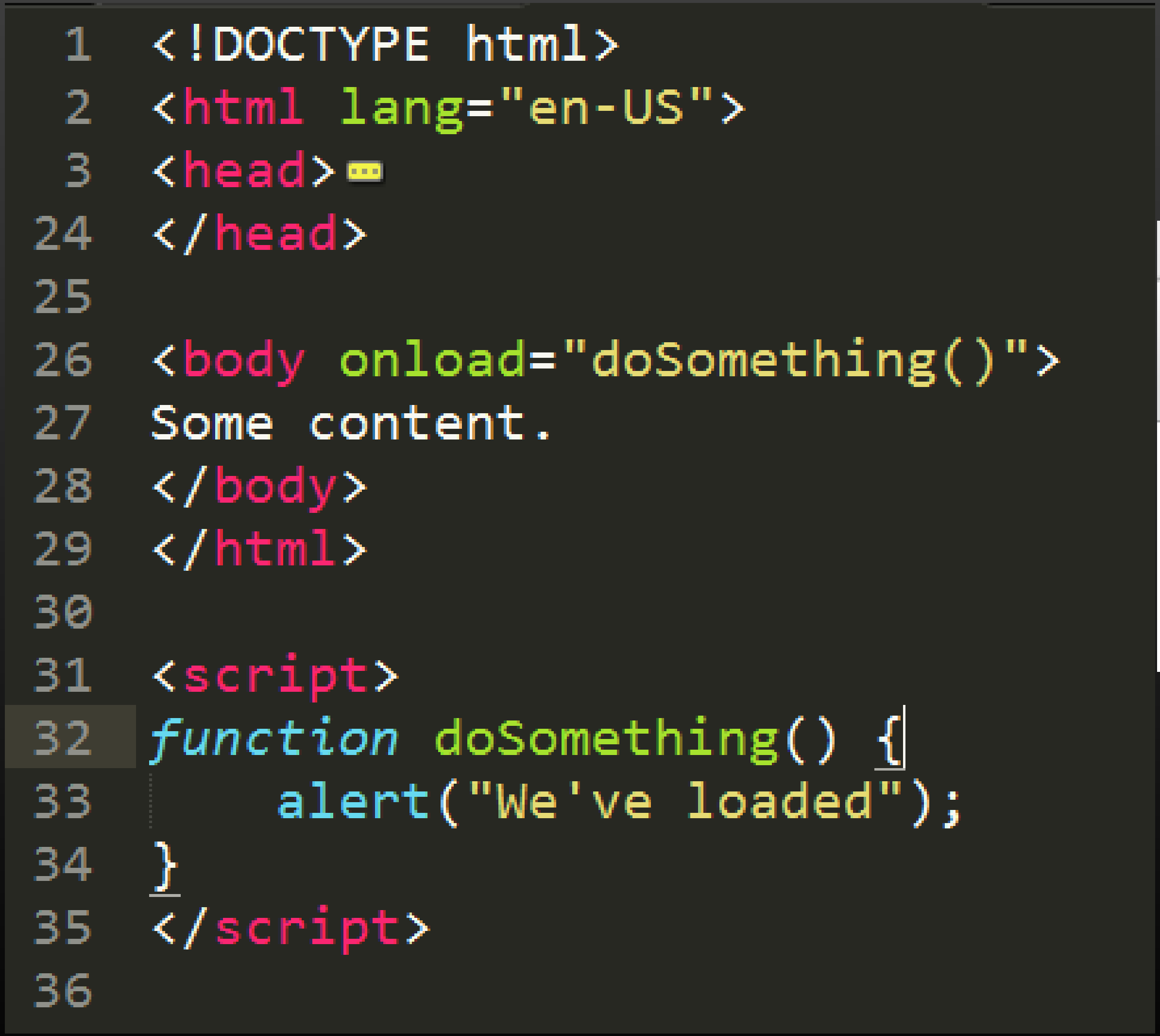
- HTML5 is not a true SGML
  - No DTD associated with HTML5



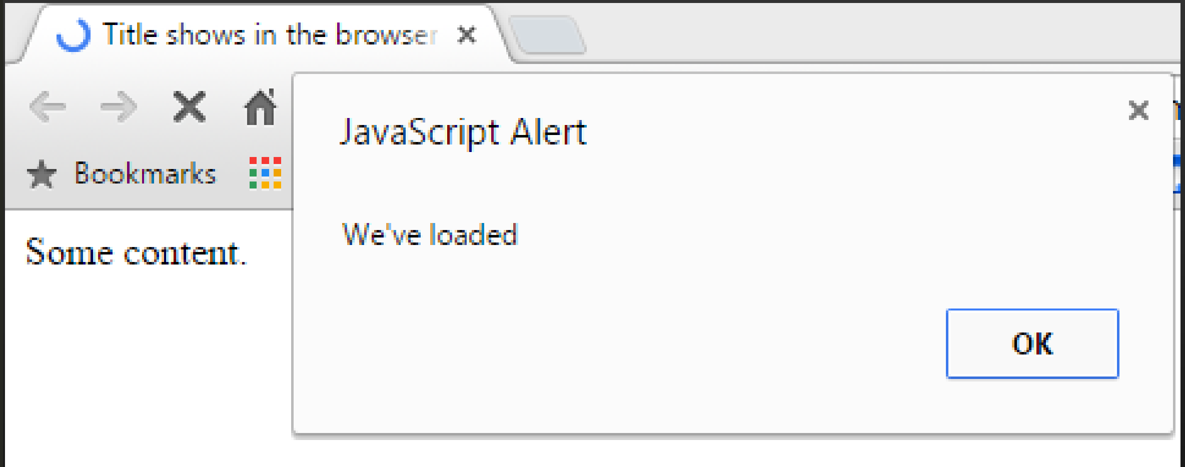
```
<!DOCTYPE html>
<html lang="en-US">
<!-- language is helpful for screen
     readers and search engines -->
```

- 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



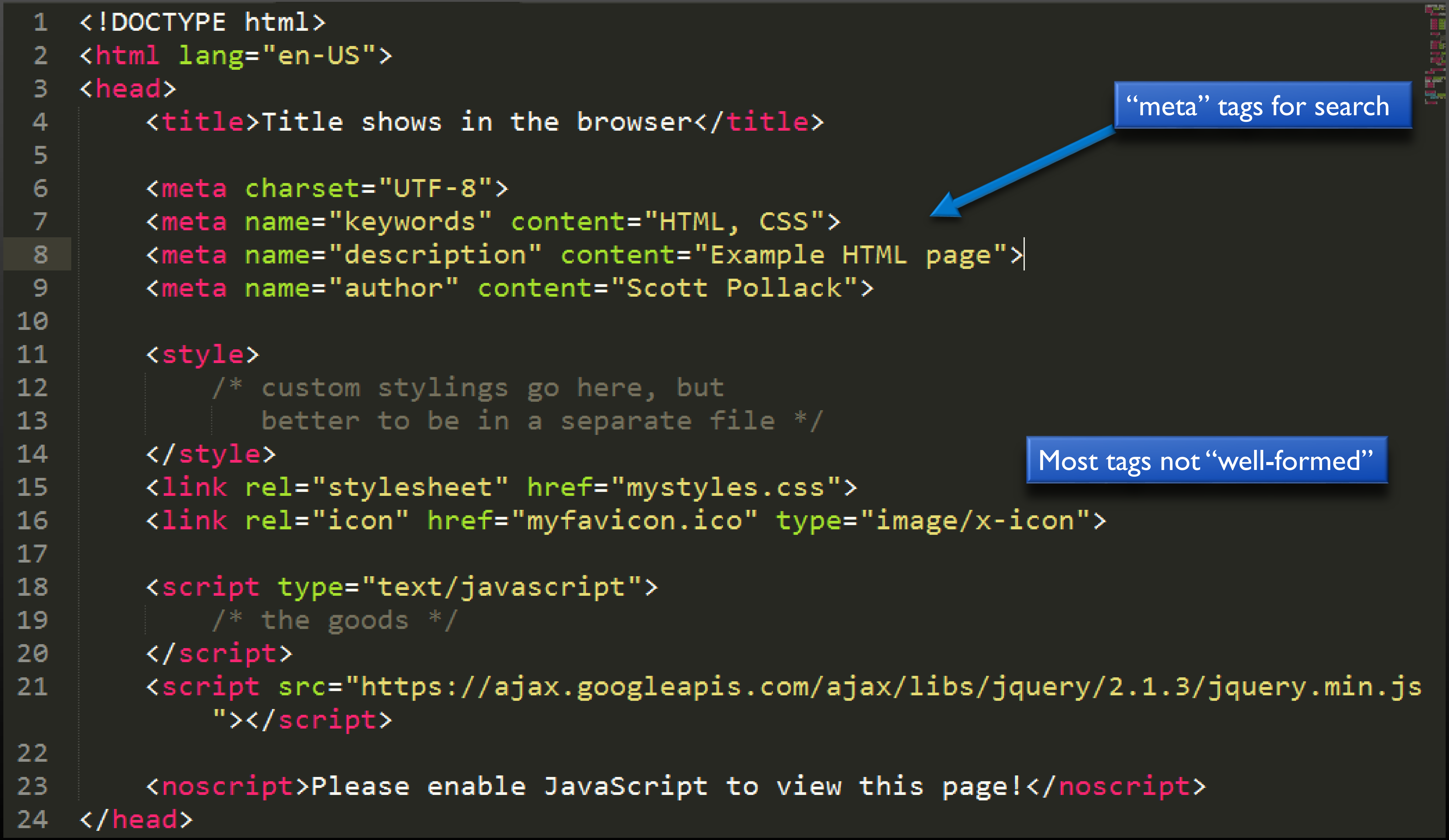
```
<!DOCTYPE html>
<html lang="en-US">
<head>…
</head>

<body onload="doSomething()">
Some content.
</body>
</html>

<script>
function doSomething() {
    alert("We've loaded");
}
</script>

```



Some content.

JavaScript Alert

We've loaded

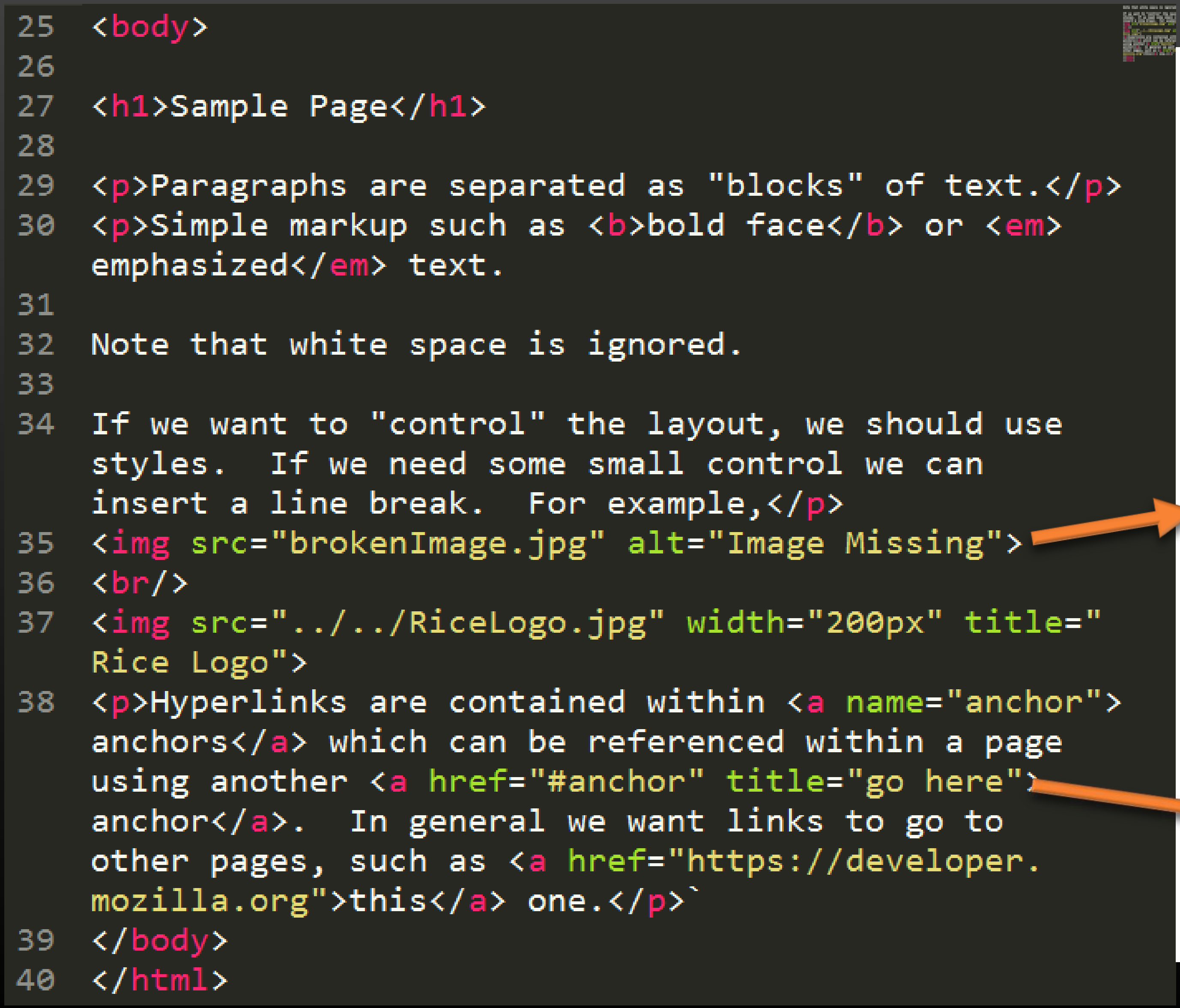


```html
<!DOCTYPE html>
<html lang="en-US">
<head>
    <title>Title shows in the browser</title>

    <meta charset="UTF-8">
    <meta name="keywords" content="HTML, CSS">
    <meta name="description" content="Example HTML page">
    <meta name="author" content="Scott Pollack">

    <style>
        /* custom stylings go here, but
           better to be in a separate file */
    </style>
    <link rel="stylesheet" href="mystyles.css">
    <link rel="icon" href="myfavicon.ico" type="image/x-icon">

    <script type="text/javascript">
        /* the goods */
    </script>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>

    <noscript>Please enable JavaScript to view this page!</noscript>
</head>
```

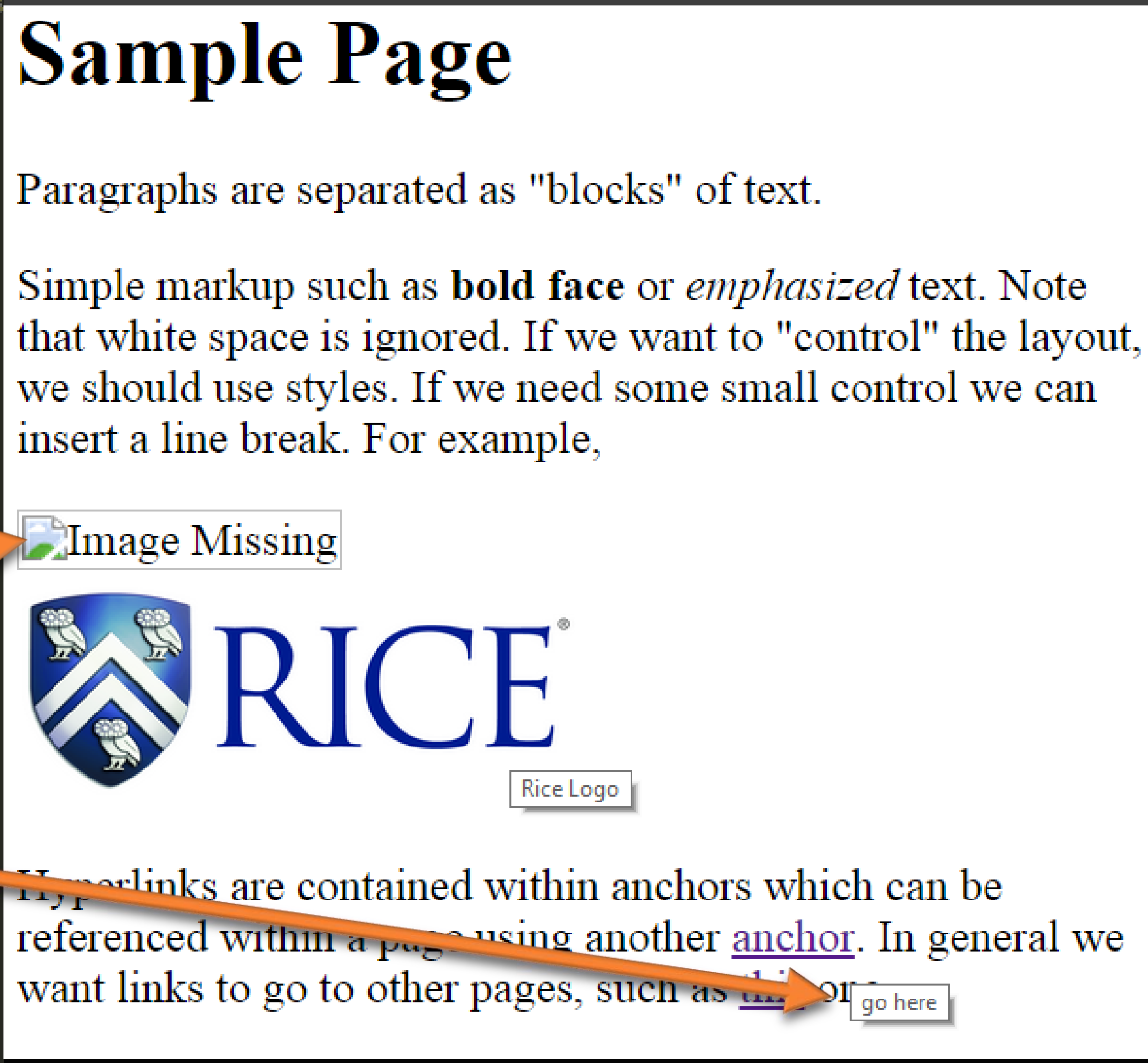
“meta” tags for search

Most tags not “well-formed”

Markup



```
<body>

<h1>Sample Page</h1>

<p>Paragraphs are separated as "blocks" of text.</p>
<p>Simple markup such as <b>bold face</b> or <em>
emphasized</em> 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,</p>
<img src="brokenImage.jpg" alt="Image Missing">
<br/>
<img src="../../RiceLogo.jpg" width="200px" title="
Rice Logo">
<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>`
</body>
</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,

Image Missing

Rice Logo

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 one.

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>

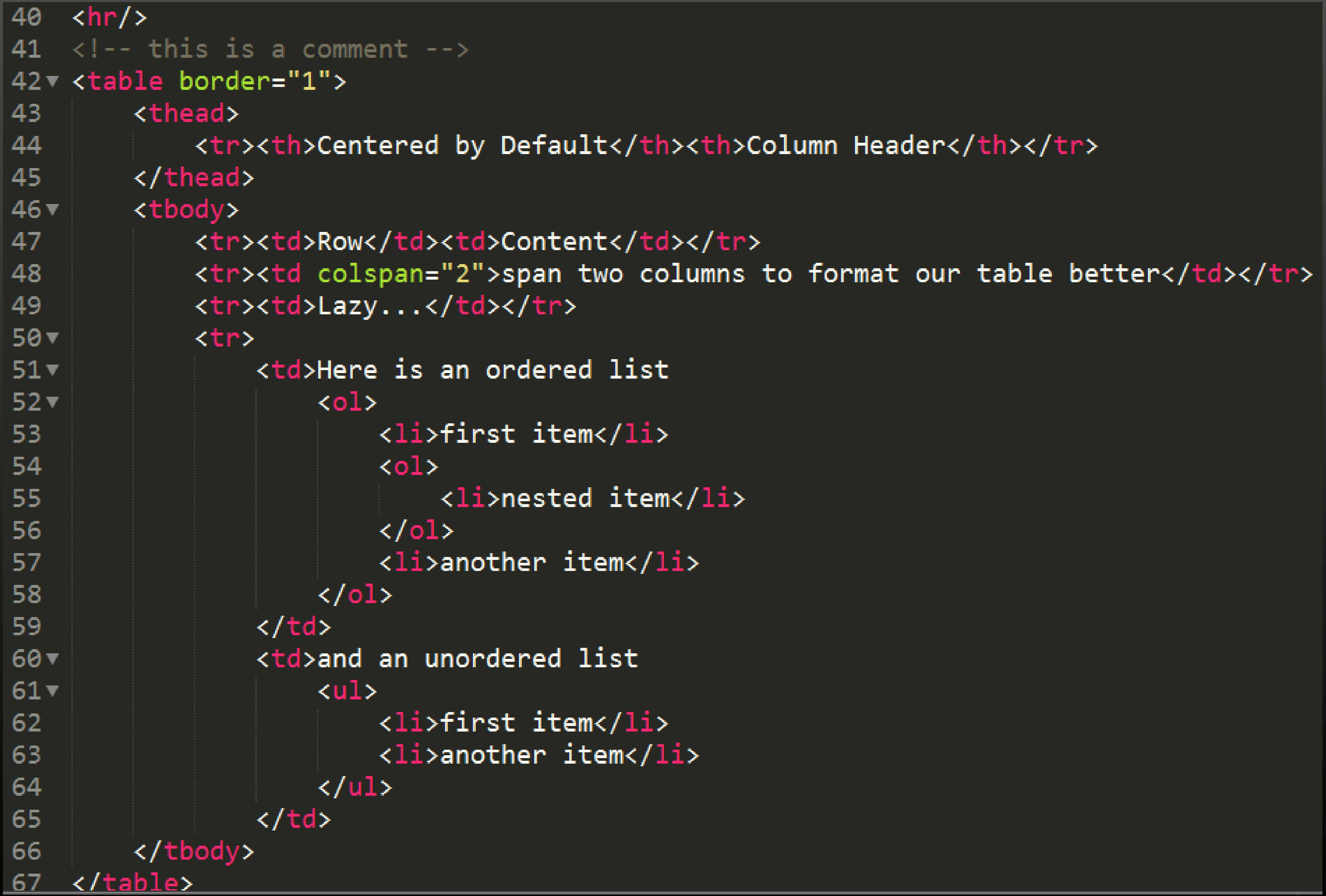
<span>

Block

<p>

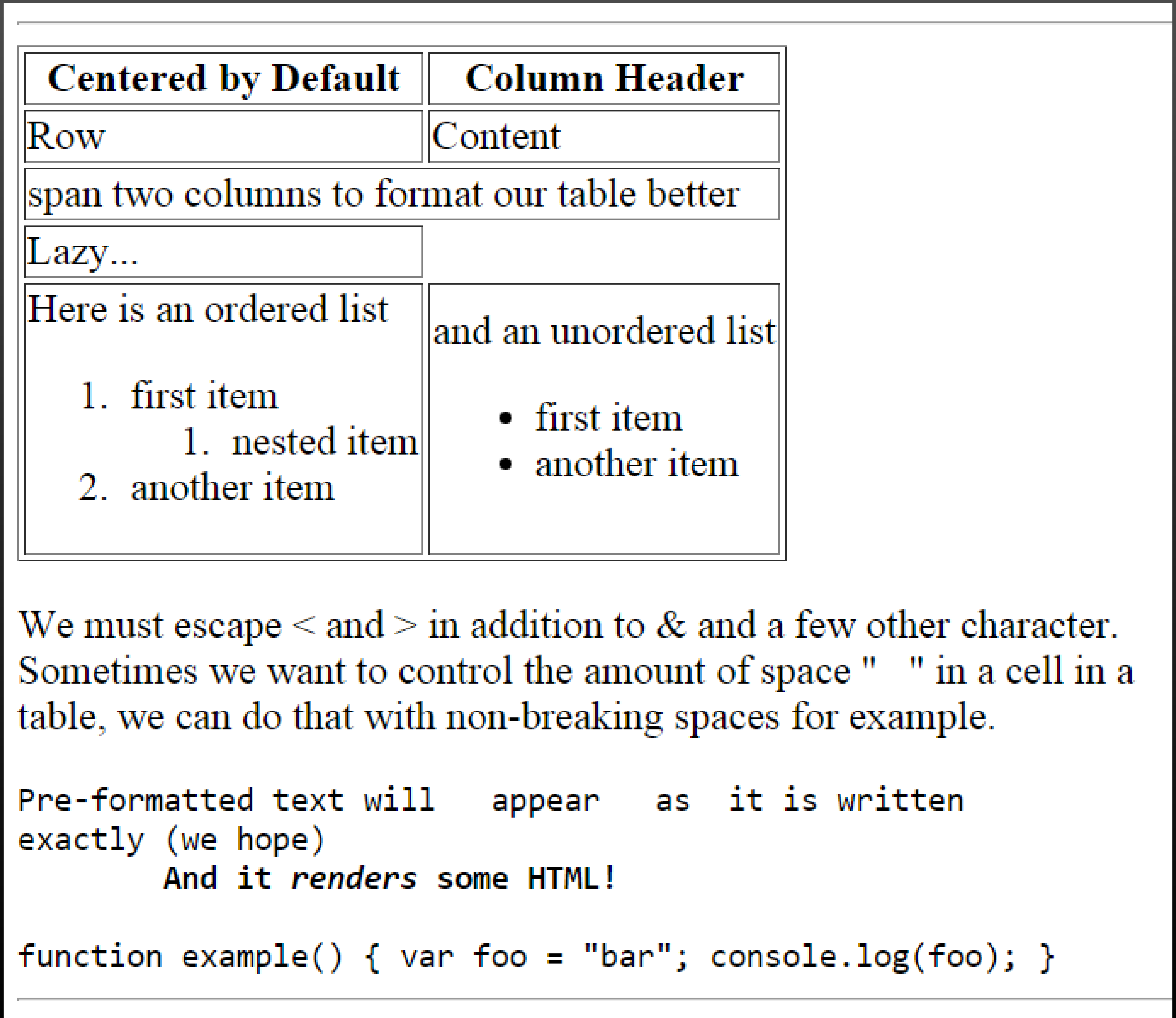
<div>

More HTML fundamentals



```html
<hr/>
<!-- this is a comment -->
<table border="1">
    <thead>
        <tr><th>Centered by Default</th><th>Column Header</th></tr>
    </thead>
    <tbody>
        <tr><td>Row</td><td>Content</td></tr>
        <tr><td colspan="2">span two columns to format our table better</td></tr>
        <tr><td>Lazy...</td></tr>
        <tr>
            <td>Here is an ordered list
                <ol>
                    <li>first item</li>
                    <ol>
                        <li>nested item</li>
                    </ol>
                    <li>another item</li>
                </ol>
            </td>
            <td>and an unordered list
                <ul>
                    <li>first item</li>
                    <li>another item</li>
                </ul>
            </td>
    </tbody>
</table>
```

More HTML fundamentals

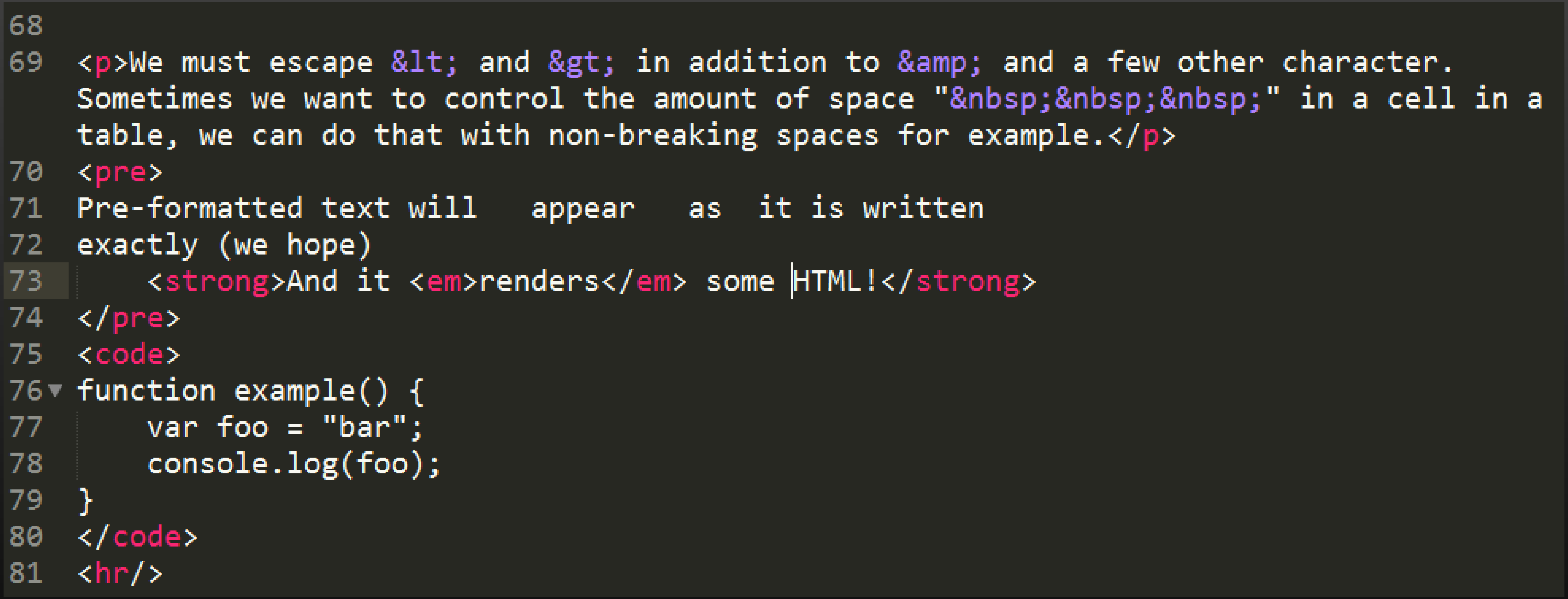


Centered by Default	Column Header
Row	Content
span two columns to format our table better	
Lazy...	
Here is an ordered list 1. first item 1. nested item 2. another item	and an unordered list • first item • another item

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



```
<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>
<pre>
Pre-formatted text will   appear   as  it is written
exactly (we hope)
    <strong>And it <em>renders</em> some HTML!</strong>
</pre>
<code>
function example() {
    var foo = "bar";
    console.log(foo);
}
</code>
<hr/>
```

Universal Resource Indicators (URIs)

Likely most familiar with universal resource locations (URLs)

https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js

mailto:somebody@rice.edu?subject=NeatO&body=TaDa

http://somesite.com:8080/example/page.php?q=5&b=8#someAnchor

scheme: //server :port path-to resource ?query #fragment

e.g., file:///C/Windows/...

Links on a page can be absolute or relative

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/

Agenda

- Introductions
- Administration
- Resources
- Course Outline
- HTML
- In-class Exercise

How to Turnin

- Mac or Linux

  ```
  scp <local_file> <netID>@ssh.clear.rice.edu:<remote_file>
  ```

  Windows

  Use *scp* if you have it.

  Otherwise on RiceOwls you can navigate to

  ```
  \\storage.rice.edu\<netID[0]>-home\<netID>
  e.g., \\storage.rice.edu\s-home\sep1
  ```

```
${HOME}/comp531/inclass/inclass-1/myPage.html

~/comp531/inclass> turnin COMP431-S16:inclass-1
```

"431" regardless of which class you are enrolled in. Case sensitive!

In-Class Exercise: Hello HTML! http://jsfiddle.net/bpenp4dq/

Create a web page telling us about you. Your page should contain:

- Your name
- A picture
- Some text, such as a story of your last trip
- Some links, such as to your favorite places, organizations, or your Facebook, Google+, or other social networking page
- A textarea for people to leave a comment or question and include a submit button (but it does nothing)

- Follow proper syntax (don't be lazy)
  - Jsfiddle performs extra wrapping,
  - so write your page in a text file
  - Validate you can view your page in your browser and it is correct
- Satisfy all of the requirements
  - to the "T"
  - i.e., don't do things differently than were asked for

you should be submitting: `inclass-1/hello.html`

COMP431-S16:inclass-1