CS 117: Internet Scale Distributed Systems (Fall 2023)

Document Architecture: HTML & CSS

Noah Mendelsohn
Tufts University
Email: noah@cs.tufts.edu
Web: http://www.cs.tufts.edu/~noah

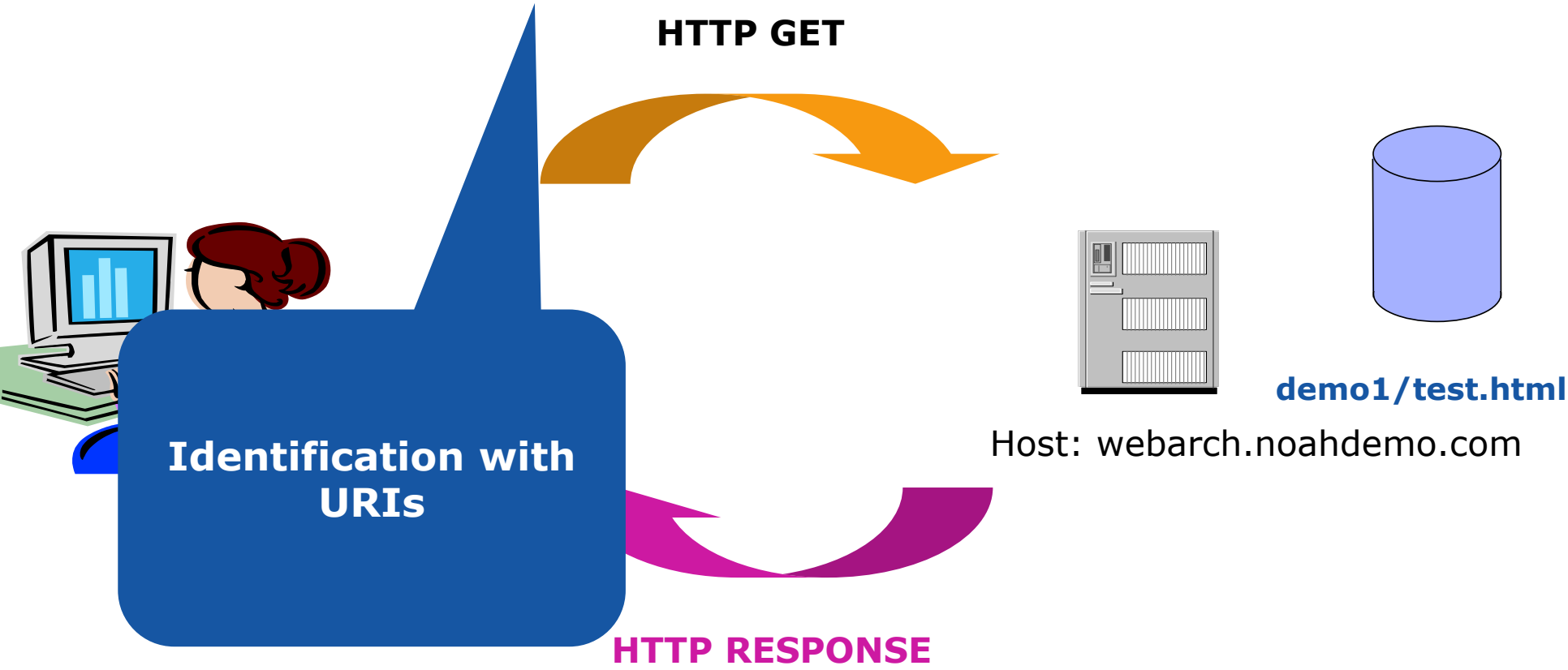
Quick Review of Web Architecture

Architecting a universal Web

- **Identification: URIs**
- **Interaction: HTTP**
- **Data formats: HTML, JPEG, GIF, etc.**

Three pillars of Web Architecture

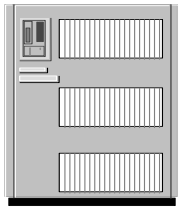
URI is http://webarch.noahdemo.com/demo1/test.html

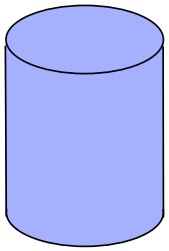


Three pillars of Web Architecture

URI is http://webarch.noahdemo.com/demo1/test.html

HTTP GET





demo1/test.html

Host: webarch.noahdemo.com

Interaction Using HTTP

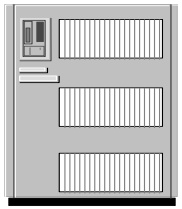
TP RESPONSE

Three pillars of Web Architecture

URI is http://webarch.noahdemo.com

HTTP GET

Representations using media types like text/html, image/jpeg, etc



demo1/test.html

Host: webarch.noahdemo.com

```
HTTP/1.1 200 OK
Date: Tue, 28 Aug 2007 01:49:33 GMT
Server: Apache
Transfer-Encoding: chunked
Content-Type: text/html

<!DOCTYPE html>
<html>
<head>
<title>Demo #1</title>
</head>
<body>
<h1>A very simple Web page</h1>
</body>
</html>
```

Architecting a universal Web

- **Identification: URIs**
- **Interaction: HTTP**
- **Data formats: HTML, JPEG, GIF, etc.**

What you should get from today's session

- **Quick look at design goals for HTML**
- **Learn how to write a simple Web page using HTML**
- **Learn the basics of using CSS for styling**
- **Learn how to validate your HTML and CSS**
- **Learn about higher level design issues relating to *document* formats**

Introduction to HTML

HTML Goals

- **Ubiquity – Metcalfe's law (previous lecture)**
- **Usable for a very wide range of *document* content – new and legacy**
- **Easy to learn**
- **To be the preferred format for Web publishing**
- **Usable with many languages (French, Chinese, English, etc.)**
- **Same HTML renders on wide range of devices – can look different**
- **Extensible…**
- **…and forwards compatible (old browsers still work!)**
- **A framework for Web *applications***

We will dive deeper into these characteristics later in the term.

A simple HTML document

```
<!DOCTYPE html>
<html>
<head>
<title>Demo #1</title>
</head>
<body>
<h1>A very simple Web page</h1>
</body>
</html>
```

link

A simple HTML document

Identifies this as an HTML document
(DOCTYPE optional, but for CS 117, please provide it)

```
<!DOCTYPE html>
<html>
<head>
<title>Demo #1</title>
</head>
<body>
<h1>A very simple Web page</h1>
</body>
</html>
```

link

Matched *tags*

```
<!DOCTYPE html>
<html>
<head>
<title>Demo #1</title>
</head>
<body>
<h1>A very simple Web page</h1>
</body>
</html>
```

link

Nested matched tags

```
<!DOCTYPE html>
<html>
<head>
<title>Demo #1</title>
</head>
<body>
<h1>A very simple Web page</h1>
</body>
</html>
```

link

A simple HTML document

The whole matched grouping is called an *element*

```
<!DOCTYPE html>
<html>
<head>
<title>Demo #1</title>
</head>
<body>
<h1>A very simple Web page</h1>
</body>
</html>
```

link

A simple HTML document

The <head> element has general information about the document

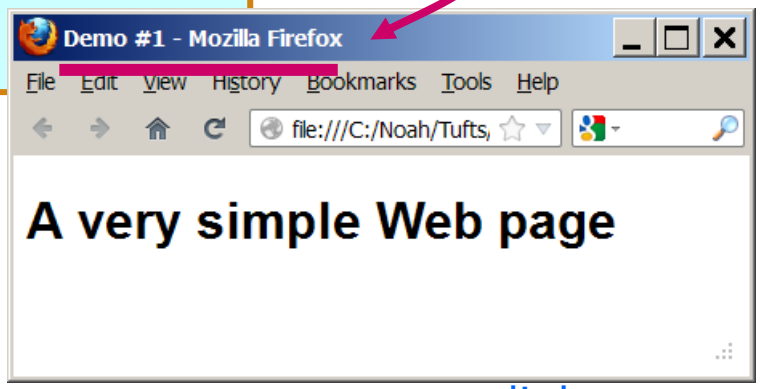
```
<!DOCTYPE html>
<html>
<head>
<title>Demo #1</title>
</head>
<body>
<h1>A very simple Web page</h1>
</body>
</html>
```

link

A simple HTML document

The <title> element gives a title – usually shown in browser title bar

```
<!DOCTYPE html>
<html>
<head>
<title>Demo #1</title>
</head>
<body>
<h1>A very simple Web page</h1>
</body>
</html>
```

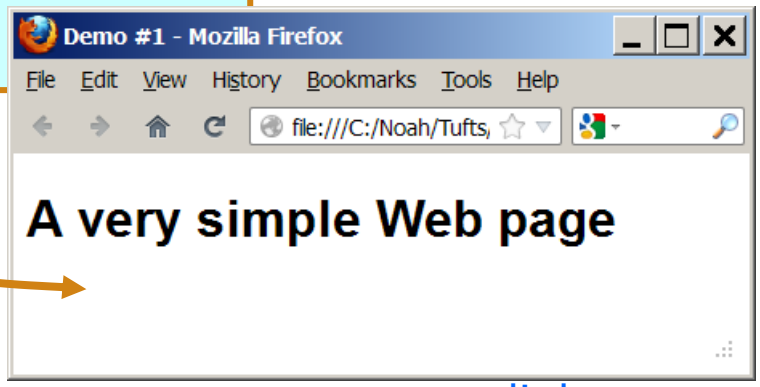


link

A simple HTML document

The <body> element is the actual content

```
<!DOCTYPE html>
<html>
<head>
<title>Demo #1</title>
</head>
<body>
<h1>A very simple Web page</h1>
</body>
</html>
```

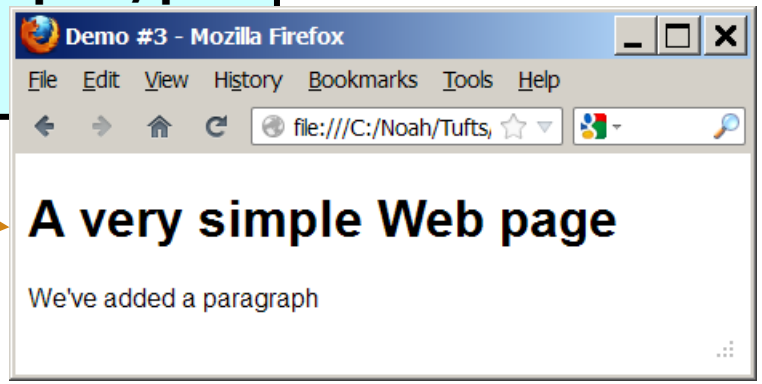


link

Headings

<h1> is a top-level heading

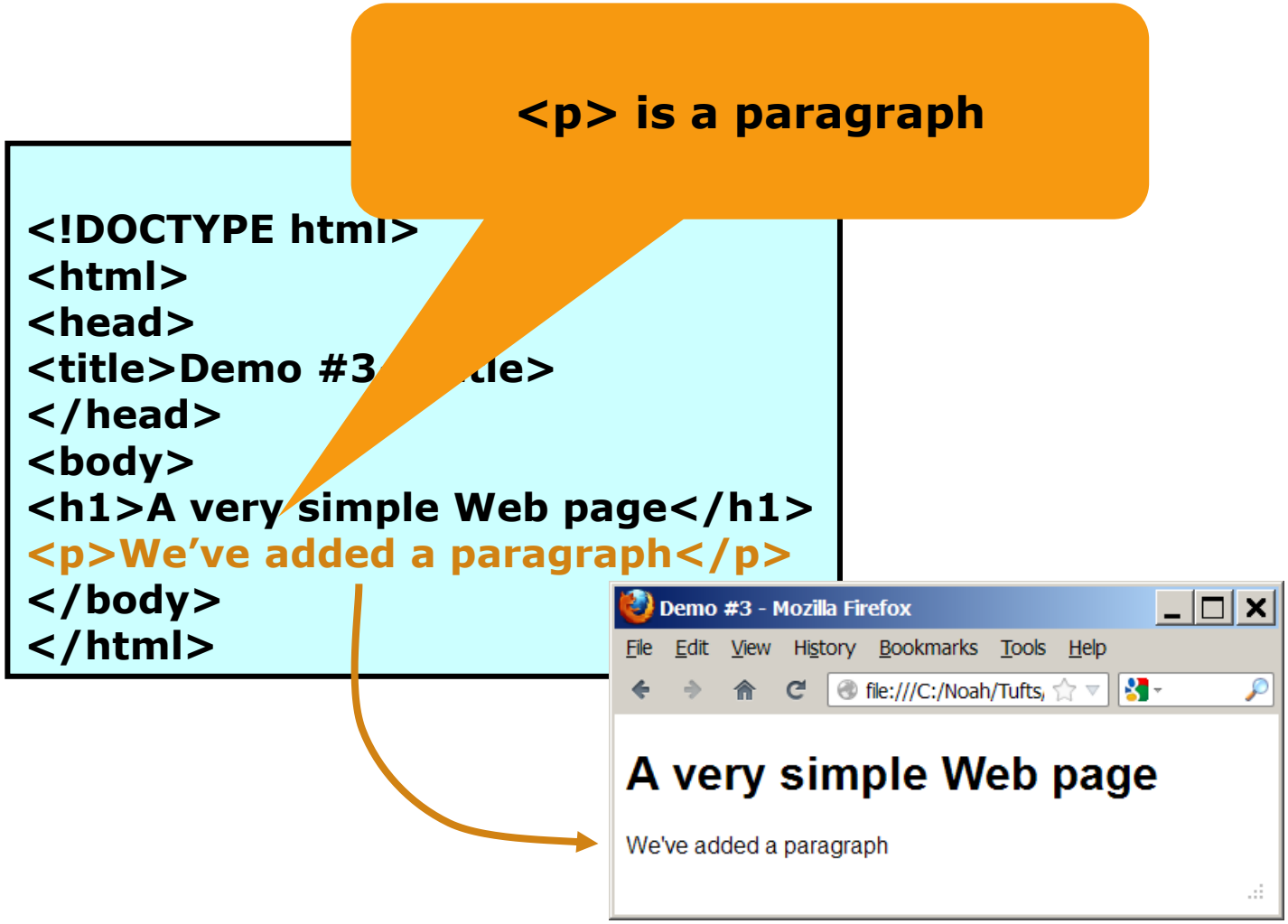
```
<!DOCTYPE html>
<html>
<head>
<title>Demo #3</title>
</head>
<body>
<h1>A very simple Web page</h1>
<p>We've added a paragraph</p>
</body>
</html>
```



<h2>, <h3> etc. for lesser headings

Link

Paragraphs are the main text content



Link

Elements can have *attributes*

Elements *may* have one or more attributes…attribute values (typically) in matched quotes

```
<!DOCTYPE html>
<html>
<head>
<title>Demo #2</title>
</head>
<body>
<h1 id="simple">A very simple Web page</h1>
</body>
</html>
```

The *id* attribute

The id attribute gives the element a name…id's *must* be unique within a document

…because ID's can be used in URI *fragment identifiers*

```
<!DOCTYPE html>
<html>
<head>
<title>Demo #2</title>
</head>
<body>
<h1 id="simple">A very simple Web page</h1>
</body>
</html>
```

http://example.org/demo2.html#simple

Self-closing tags

Some tags are or may be self-closing (there's no </img> in this document)

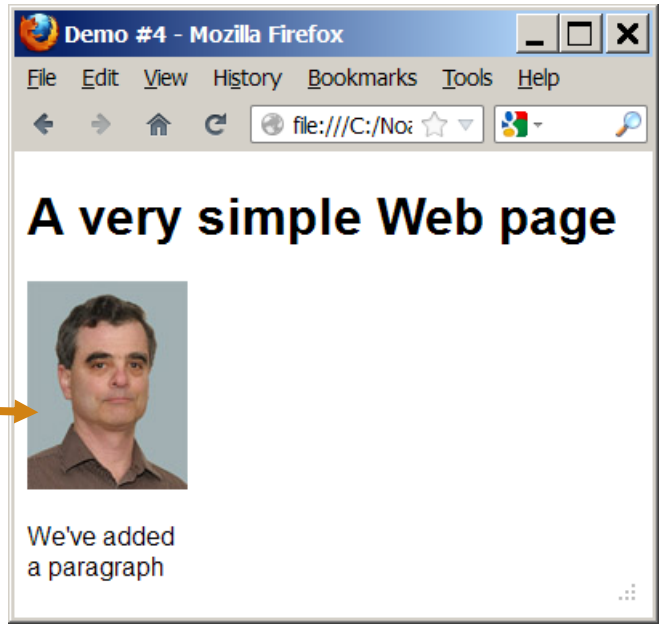
```
<!DOCTYPE html>
<html>
<head>
<title>Demo #4</title>
</head>
<body>
<h1>A very simple Web page</h1>
<img src="noah.jpg" alt="picture of Noah">
<p>We've added
<br>
a paragraph</p>
</body>
</html>
```

Link

Images

<img> includes an image

```
<!DOCTYPE html>
<html>
<head>
<title>Demo #4</title>
</head>
<body>
<h1>A very simple Web page</h1>
<img src="noah.jpg">
<p>We've added
<br>
a paragraph</p>
</body>
</html>
```

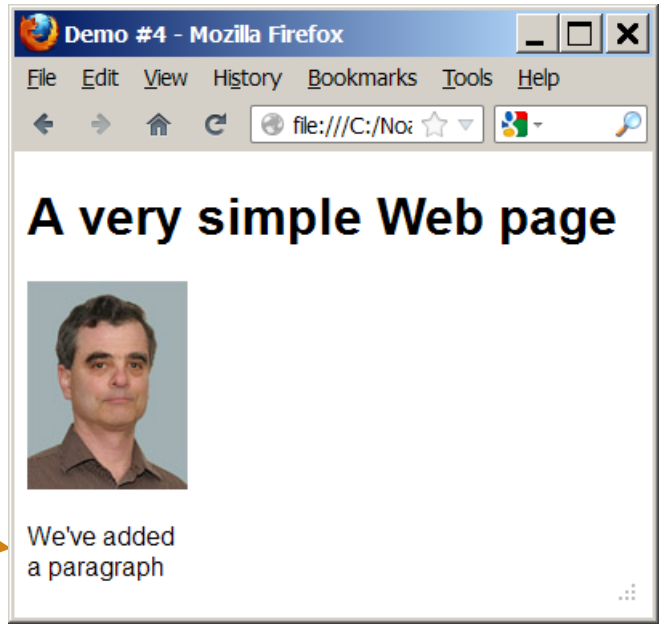


Link

Line breaks

***
 starts new line, but not new para***

```
<!DOCTYPE html>
<html>
<head>
<title>Demo #4</title>
</head>
<body>
<h1>A very simple Web page</h1>
<img src="noah.jpg>
<p>We've added
<br>
a paragraph</p>
</body>
</html>
```



Link

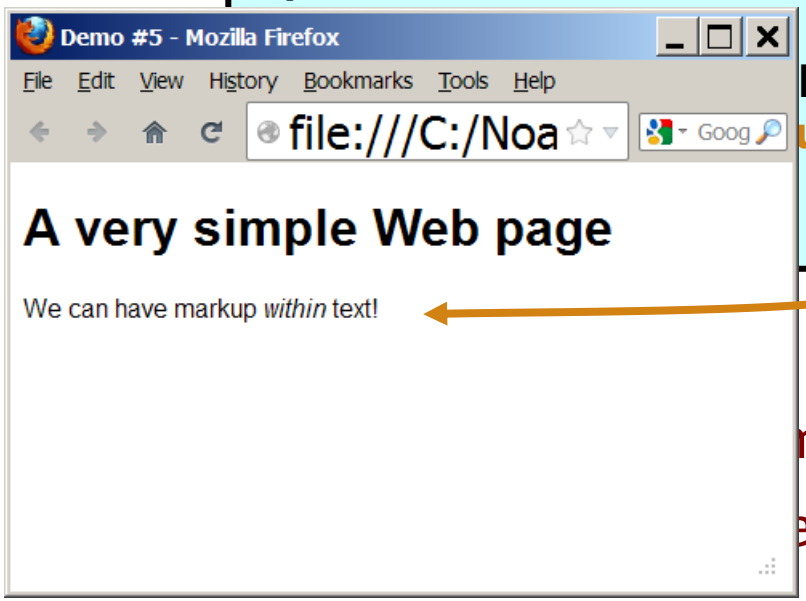
Marking up text with nested elements

```
<!DOCTYPE html>
<html>
<head>
<title>Demo #5</title>
</head>
```

b page</h1>

up within text!</p>

 is for "emphasis"



markup *within* text!

ed content – more on that later.

Link

Some other important HTML features

- **Lists: `<ul> & <ol>`**
- **Tables: `<table>`**
- **Block layout and formatting: `<div>`**
- **Input forms and submission: `<form>`**
- **Graphics: `<svg> & <canvas>`**
- **Video: `<video>`**
- **Tables: `<table>`**
- **Programmable documents and Web apps: `<script>`**
- **Etc.**

Links

Linking to other documents gives the Web its richness!

```
<!DOCTYPE html>
<html>
<head>
<title>Demo #6</title>
</head>
<body>
<h1>A very simple Web page</h1>
<p>We can link
to <a href="http://example.org/another.html">another document</a>.</p>
</body>
</html>
```

<a> is for "anchor"…this is how you put in links

Link

Linking to other documents gives the Web its richness!

```
<!DOCTYPE html>
<html>
<head>
<title>Demo #6</title>
</head>
<body>
<h1>A very simple Web page</h1>
<p>We can link
to <a href="http://example.org/another.html">another document</a>.</p>
</body>
</html>
```

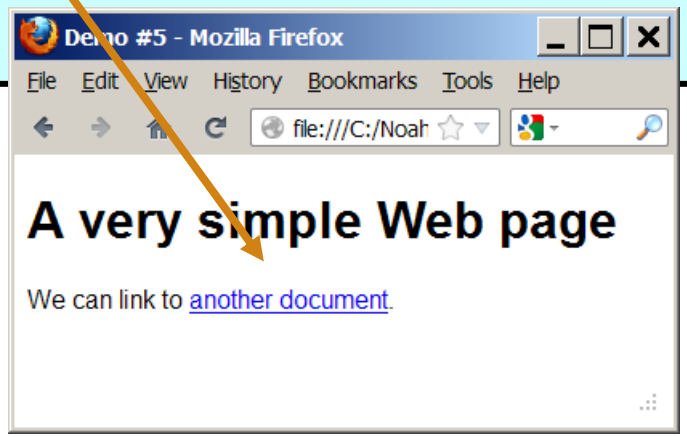
href = identifies the document we're linking

Link

Linking to other documents gives the Web its richness!

```
<!DOCTYPE html>
<html>
<head>
<title>Demo #6</title>
</head>
<body>
<h1>A very simple Web page</h1>
<p>We can link
to <a href="http://example.org/another.html">another document</a>.</p>
</body>
</html>
```

Content appears in the document

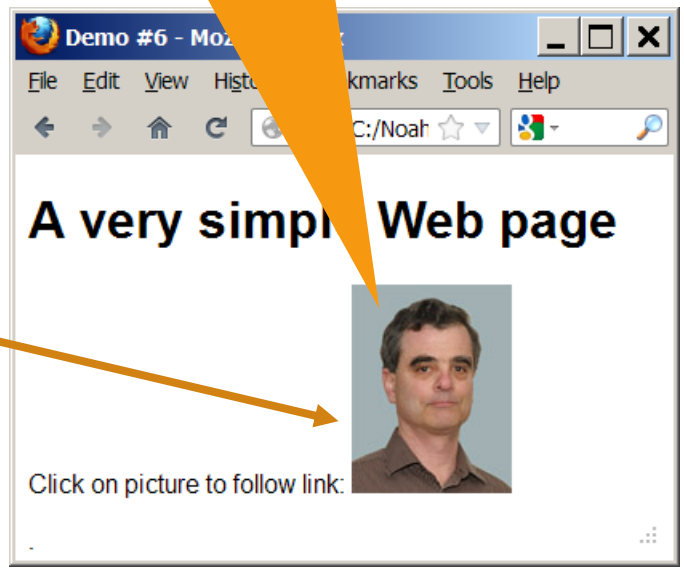


Link

Links don’t have to be text

```
<!DOCTYPE html>
<html>
<head>
<title>Demo #7</title>
</head>
<body>
<h1>A very simple Web page</h1>
<p>Click on picture to follow link:
<a href="http://www.cs.tufts.edu/~noah/">
  <img src="noah.jpg" alt="picture of Noah">
</a>
</p>.
</body>
</html>
```

Click picture to follow link



Link

HTML is *compositional*

```
<!DOCTYPE html>
<html>
<head>
<title>Demo #7</title>
</head>
<body>
<h1>A very simple Web page</h1>
<p>Click on picture to follow link:
<a href="http://www.cs.tufts.edu/~noah/">
  <img src="noah.jpg" alt="picture of Noah">
</a>
</p>.
</body>
</html>
```

<u><img> can appear in</u>

- Paragraphs
- Links
- Table cells
- <divs>
- Etc.

<u>Recursion</u>

- Tables within tables
- Tables in <divs>
- <divs> in tables
- Images in content in lists in tables

HTML References

- **The official HTML5 specification: HTML5: Edition for Web Authors (https://html.spec.whatwg.org/multipage/)**
  - This is long, detailed and often hard to understand, but it is authoritative
  - The history of the HTML specifications is interesting and convoluted
- **There are tons of good books on HTML – check Safari or your favorite bookseller for one that works for you**
- **A Web search will get you good (and bad!) advice on almost anything you might want to do**
- **A useful, if informal guide to new features in HTML5: *Dive into HTML5*, by Mark Pilgrim http://diveintohtml5.info/table-of-contents.html**
  - Most of these won't be of interest when you're starting out, but you might want to take a look
  - Support the author and buy the book form – info on the intro page!

Note: the Web site www.w3schools.com is not affiliated with W3C, and does not convey official advice from the consortium!

Introduction to CSS

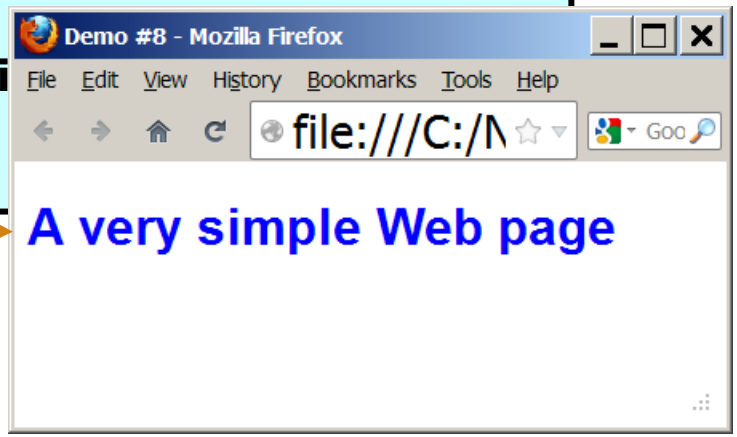
HTML and CSS: content and styling

- **HTML conveys the logical structure and content of a document**
- **Each element is given default styling by the browser**
  - E.g. <h1> is usually in a big font and boldface
- **Cascading Stylesheets (CSS) can be used to override the presentation details of any element!**

A simple bit of CSS on an element

color:blue overrides the color style for this <h1>

```
<!DOCTYPE html>
<html>
<head>
<title>Demo #8</title>
</head>
<body>
<h1 style="color:blue">A very si
</body>
</html>
```



Link

What you can control using CSS

- color:red
- **font-weight:bold**
- *font-style:italic*
- *font-size:20px*
- background:yellow
- border:solid
- text-align:center
- border:dashed; text-align:center;

Example:

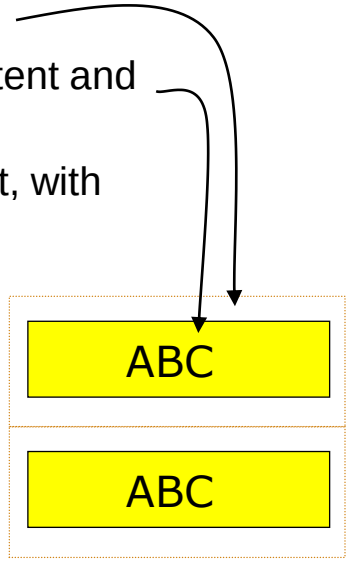
```
<p style="text-align:center; border:solid; color:red;">CSS is cool!</p>
```

CSS is cool!

What you can control using CSS

- color:red
- **font-weight:bold**
- *font-style:italic*
- *font-size:20px*
- background:yellow
- border:solid
- text-align:center
- border:dashed; text-align:center;

- **How things are positioned:**
  - margins: around the outside
  - padding: space between content and bounding box
  - float: forcing a box right or left, with wraparound
- **Advanced**
  - Animations
  - Opacity
  - Different stylesheet for mobile, print, small screen



And lots, lots, lots more!

Where can you put your CSS?

Use sparingly for local changes you won't want to override

- **On an element**
  - `<p style="color:blue">A blue paragraph</p>`
- **At the top of your HTML file**

Can be useful to make sure formatting stays w/file, e.g. offline

  -
```
<html>
<head>
<style type="text/css">
p {color:blue;}   /* all paragraphs blue */
</style>
</head>
<body>
<p>This will be blue!</p>
</body>
</html>
```
- In an external stylesheet file

External stylesheets are usually best practice

  -
```
<html>
<head>
<link rel="stylesheet" type="text/css"
      href="http://www.cs.tufts.edu/comp/117/style/150ids.css" />
</head>
</html>
```

CSS Selectors

A simple CSS selector

p "selects" all <p> elements

```
<!DOCTYPE html>
<html>
<head>
<style type="text/css">
p {color:red;}
</style>
</head>
<body>
<h1>A very simple Web page</h1>
<p>This will be red</p>
<p>...and this will be red</p>
<p>...and this too</p>
</body>
</html>
```

Link

A simple CSS selector

p "selects" all <p> elements

```
<!DOCTYPE html>
<html>
<head>
<title>Demo #9</title>
<style type="text/css">
p {color:red;}
</style>
</head>
<body>
<h1>A very simple Web page</h1>
<p>This will be red</p>
<p>...and this will be red</p>
<p>...and this too</p>
</body>
</html>
```

A very simple Web Page

This will be red
...and this will be red
...and this too

Link

Using element *classes* with CSS

p.hilite "selects" all
<p class="hilite"> elements

```
<!DOCTYPE html>
<html>
<head>
<title>Demo #10</title>
<style type="text/css">
p.hilite {color:red;}
</style>
</head>
<body>
<h1>A very simple Web page</h1>
<p class="hilite">This will be red</p>
<p>...this won't...</p>
<p class="hilite">...but this will!</p>
</body>
</html>
```

A very simple Web Page

This will be red
...this won't...
...but this will!

You can make up your own class names

Classes can be used with multiple element types

.hilite "selects" all
<... class="hilite"> elements

```
<!DOCTYPE html>
<html>
<head>
<title>Demo #10</title>
<style type="text/css">
.hilite {font-style:italic;
         color:red;}
</style>
</head>
<body>
<h1 class="hilite">A very simple
                   Web page</h1>
<p>This will be normal/p>
<p class="hilite">This will be red
          and italic</p>
<p>This will be normal</p>
</body>
</html>
```

A very simple Web Page

This will be normal
This will be red and italic
This will be normal

Link

Using element *identifiers* with CSS

p#somep "selects" a single <p id="somep"> element

```
<!DOCTYPE html>
<html>
<head>
<style type="text/css">
p#somp {color:red;}
</style>
</head>
<body>
<h1>A very simple Web page</h1>
<p>Normal</p>
<p id="somep">...red...</p>
<p>...normal again</p>
</body>
</html>
```

A very simple Web Page

Normal
...red...
...normal again

Id's must be unique in each document
By the way: <a href="doc#somep"> links the paragraph

is useful for styling within text runs

** can be styled, but does not introduce line breaks**

```
<!DOCTYPE html>
<html>
<head>
<style type="text/css">
.hilite {font-style:italic;
         color:red;}
</style>
</head>
<body>
<h1>A very simple Web page</h1>
<p>This is an <span class="hilite">
interesting</span> paragraph!</p>
</body>
</html>
```

A very simple Web Page

This is an *interesting* paragraph.

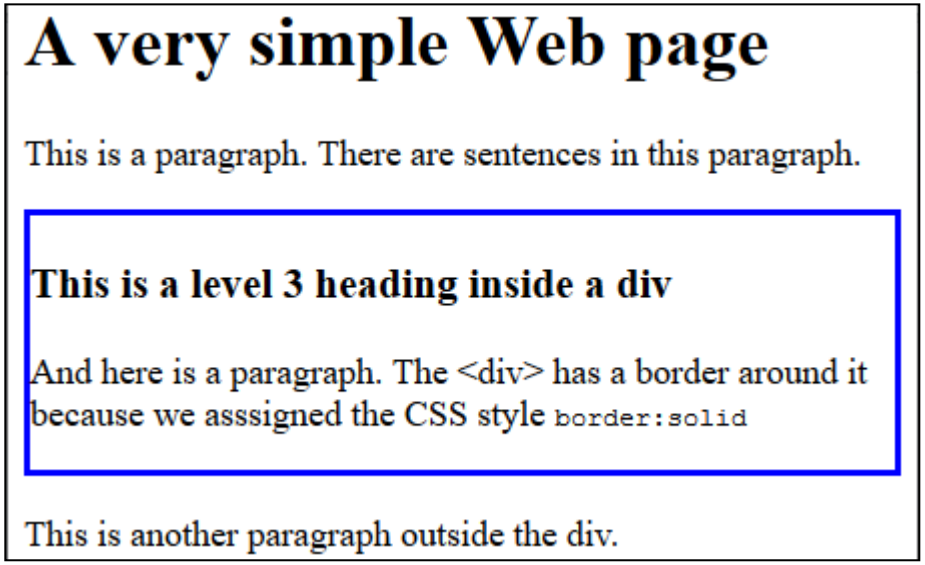
is useful for designating layout blocks

** sets off a block for layout purposes…it can also be styled (e.g. this one has a border)**

```
<html>
<head>
<title>Demo #11</title>
<style type="text/css">
div.boxdiv {border:solid; border-color:blue}
</style>
</head>
<body>
<h1 class="hilite">
 very simple Web page
</
<p>This is a paragraph. </p>

<div class="boxdiv">
  <h3>This is a level 3 heading
        inside a div</h3>
  <p>And here is a paragraph. …</p>
</div>

<p>This is another paragraph
outside the div.</p>
</body>
</html>
```



Link

With CSS float:right…we see the div layout moved

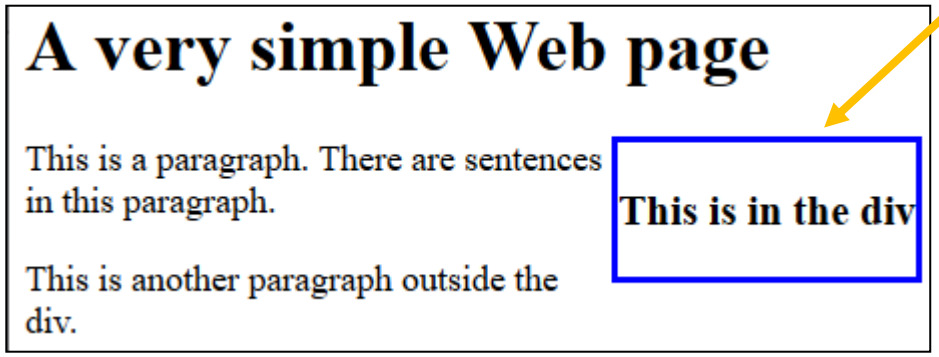
CSS `float:right` applied to the `<div>`

```
<!DOCTYPE html>
<html>
<head>
<title>Demo #12</title>
<style type="text/css">
 div.boxdiv {float:right;
             border:solid;
              border-color:blue;}
</style>
</head>
<body>
<h1 class="hilite">A very simple Web page</h1>

<div class="boxdiv">
  <h3>This is in the div</h3>
</div>

<p>This is a paragraph.
There are sentences in this paragraph.</p>


<p>This is another paragraph
outside the div.</p>
</body>
</html>
```



Link

CSS Reference Material

- **CSS work has been modularized in to many separate specifications**
  - E.g. selectors are separate from the main language
- **W3C Home Page for CSS: http://www.w3.org/Style/CSS/Overview**
  - You can find specifications and other useful material from there
- **Many, many tutorials on the Web – search for the feature you want**
  - E.g. “CSS Animations”
- **View Source**
  - If the HTML source has a link to the stylesheet, most browsers let you follow that

Things to Learn From the Design of HTML & CSS

HTML illustrates some key points of document design

- **It's a text format, not binary**
  - Easy to generate from: editors, programs, scripts, etc.
  - Want to learn how something's done? Just View Source on a page that does it!
- **Semantic markup**
  - <H1> means "This is a top level header", not "Make this big and bold"
  - Use tags like <table> when you have tabular data, not to fudge screen layout
  - Why? Tools, search engines, transcoders etc. can tell more about your document
  - See principle of model/view separation
  - Almost all the formatting of HTML elements is now defined in terms of CSS properties
- **Model/view separation**
  - Encode content separately from presentation
  - *Use CSS for formatting wherever practical!*
  - An important principle of software design: *supports information extraction & reuse*
  - An example of the broader principle of "separation of concerns"

To really understand the power of model/view separation, check out the legendary CSS Zen Garden (http://www.csszengarden.com/)

HTML: Some other things to note

- ***Mixed content* is deeply interesting**
  - Strings have nested structure that's marked up in the string
  - *Many types of documents need this richness*
  - Most important example: hyperlinks
  - *Traditional string types don't handle this well – hard to import HTML & XML into C, Java, etc.*
- **Linkability**
  - It's easy to turn most HTML content into a link
  - It's easy to link not just to an HTML document, but to content within a document
  - *Metcalfe's law at work again: the power of linking*
- **Extensible**
  - New tags are added over time
  - There's a rule: older browsers ignore new tags but not the content of those tags
  - Who gets to define new tags is the subject of much controversy

More that may not be obvious about HTML & CSS

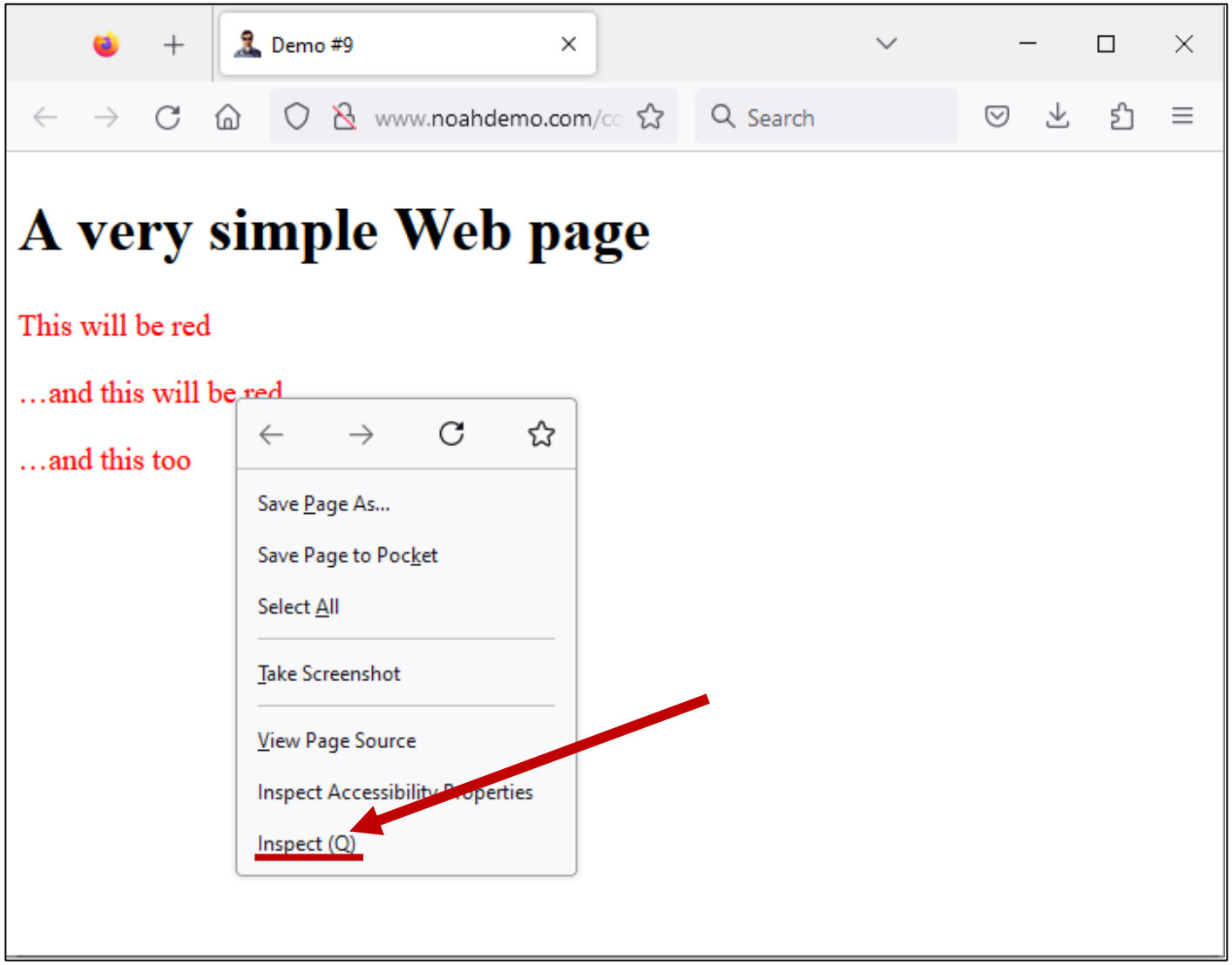
- **HTML can do a lot without JavaScript**
  - We'll explore why that's important later
  - Note that there is no JavaScript on any of the CS 117 course pages (so far!)
  - *Please do not use JavaScript in your course submissions for now!*
- **CSS and HTML are Declarative Languages**
  - Not "Do step1 then step2", but rather "make it so" (e.g. make sure this is centered, make sure it's blue, etc.)
  - Neither CSS nor HTML is imperative or Turing-complete
  - We'll discuss later the advantages of declarative languages
- **Constraint-based programming is cool**
  - "Center this" vs. "FirstCharPos = window.left + window.size/2 – string.width/2"

Testing your HTML Files

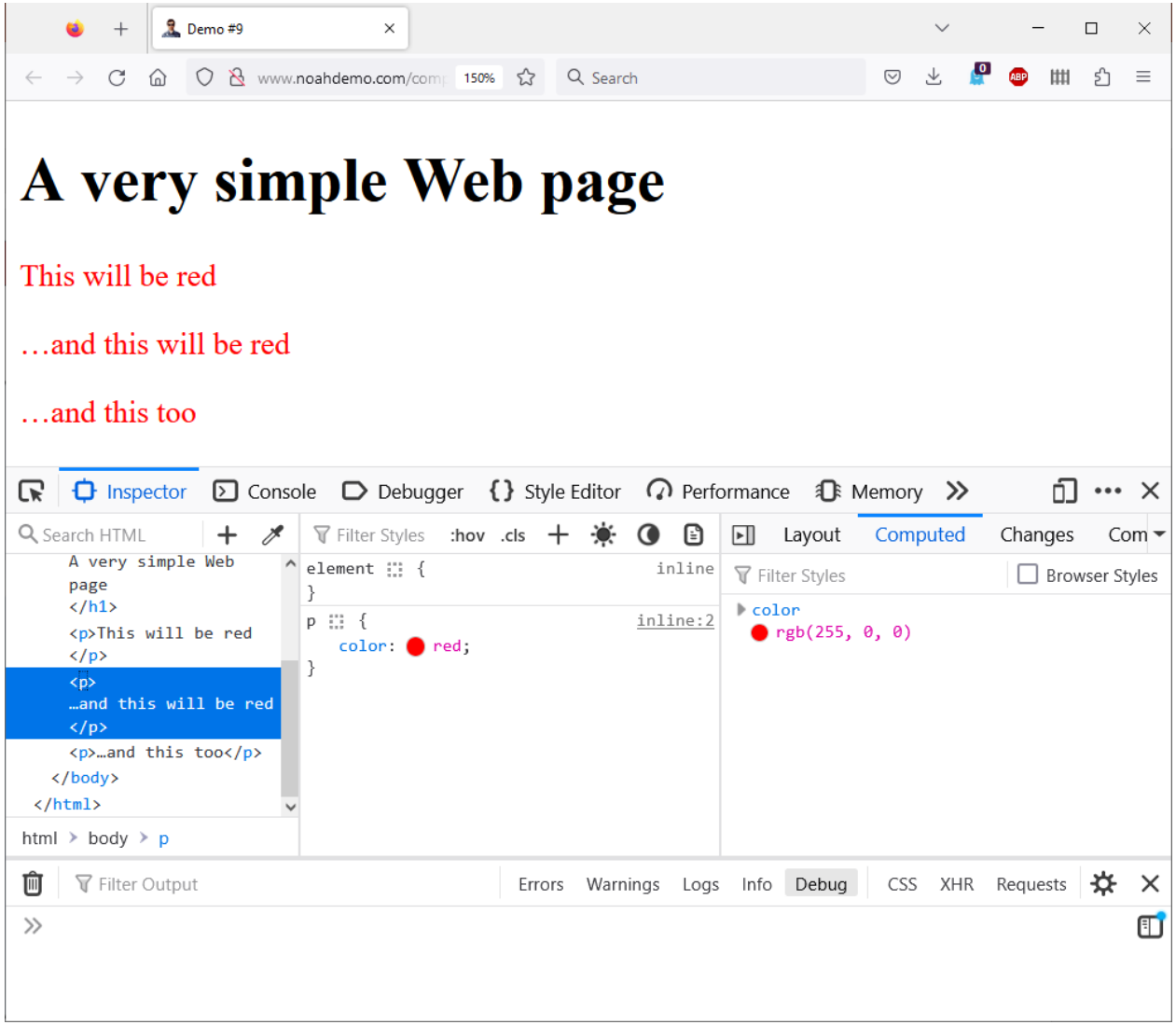
It’s easy to try out your HTML files

- **Create a file on your local machine**
- **Use your operating system’s convention for file extensions (usually .html)**
- **If you open the file, your OS will use your default browser**
- **Which browser?**
  - Any up-to-date one should do
  - I recommend Firefox, Safari and/or Chrome
  - For the features we use, compatibility should be quite good
- **HTML & CSS Debugging**
  - Firefox: right click and “Inspect Element”, or install Firebug addin
  - Safari: Preferences -> Advanced -> Show develop menu
  - Chrome: CTRL-SHIFT-I
- **Links**
  - Relative links will resolve to your local filesystem
  - Most others will attempt to retrieve from the Web
  - Make sure your stylesheet links will resolve if you’re testing locally

Debugging w/Firefox Inspect Element



Debugging w/Firefox Inspect Element



Debugging w/Firefox Inspect Element

Hover mouse here to show layout highlighting in main window

Demo #9

www.noahdemo.com/com 150% Search

A very simple Web page

This will be red

…and this will be red

…and this too

Inspector Console Debugger Performance Memory

Search HTML

```
A very simple Web page
</h1>
<p>This will be red
</p>
<p>
…and this will be red
</p>
<p>…and this too</p>
</body>
</html>
```

html > body > p

Filter Styles

```
element inline
}
p inline:2 {
  color: red;
}
```

Layout Computed Changes Com

Filter Styles Browser Styles

```
color
rgb(255, 0, 0)
```

Filter Output Errors Warnings Logs Info Debug CSS XHR Requests

Is your HTML file *correct?*

The W3C Markup Validation Service - Mozilla Firefox

File Edit View History Bookmarks Tools Help

validator.w3.org

Tufts COMP150IDS HP Pavilion p7-1225 R... Windows Virtual PC: C... Westport Point Weath...

The W3C Markup Validation Ser...

Markup Validation Service

Check the markup (HTML, XHTML, ...) of Web documents

Validate by URI | Validate by File Upload | Validate by Direct Input

Validate by URI

Validate a document online:

Address: http://www.eecs.tufts.edu/~yourname/test.html

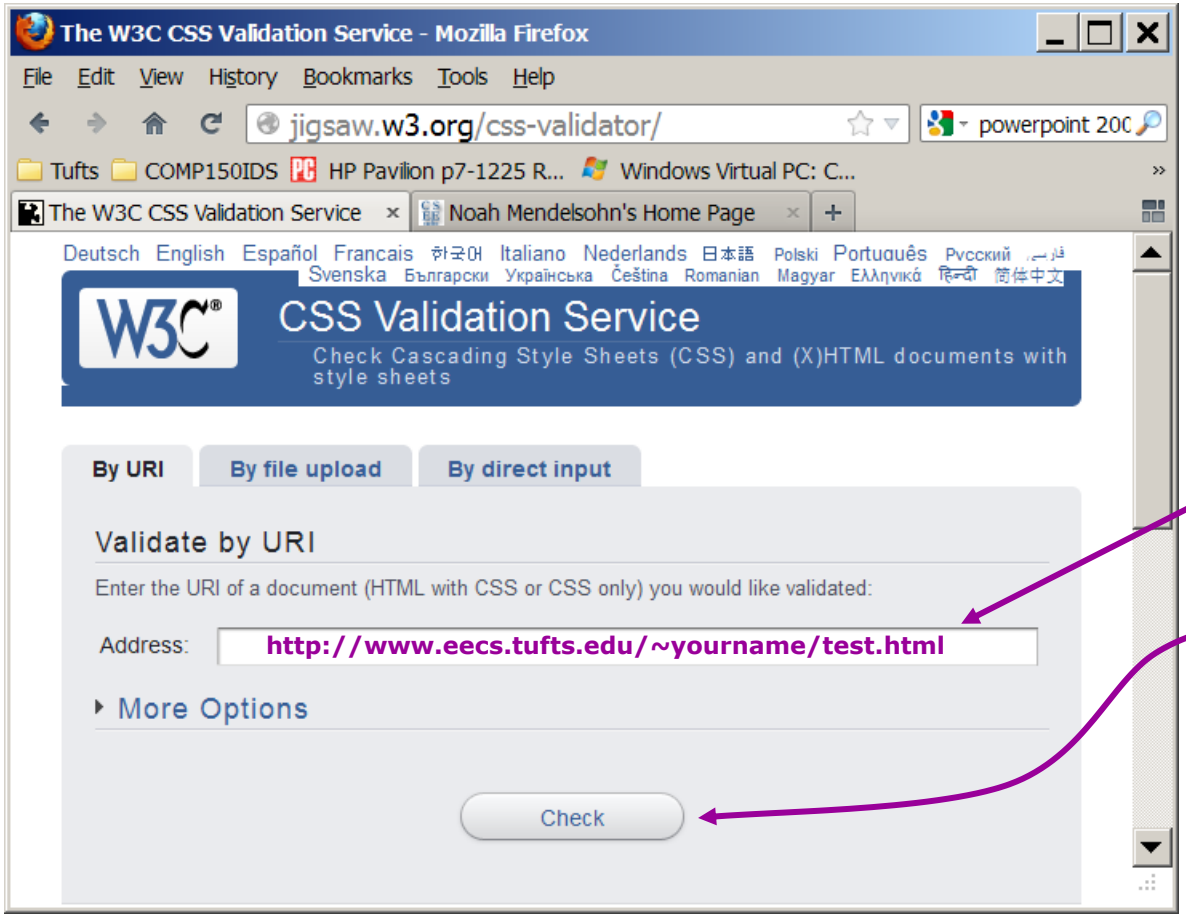
More Options

Check

This validator checks the markup validity of Web documents in HTML, XHTML, SMIL, MathML, etc. If you wish to validate specific content such as RSS/Atom feeds or CSS stylesheets, MobileOK content, or to find broken links, there are other validators and tools available. As an alternative you can also try our non-DTD-based validator.

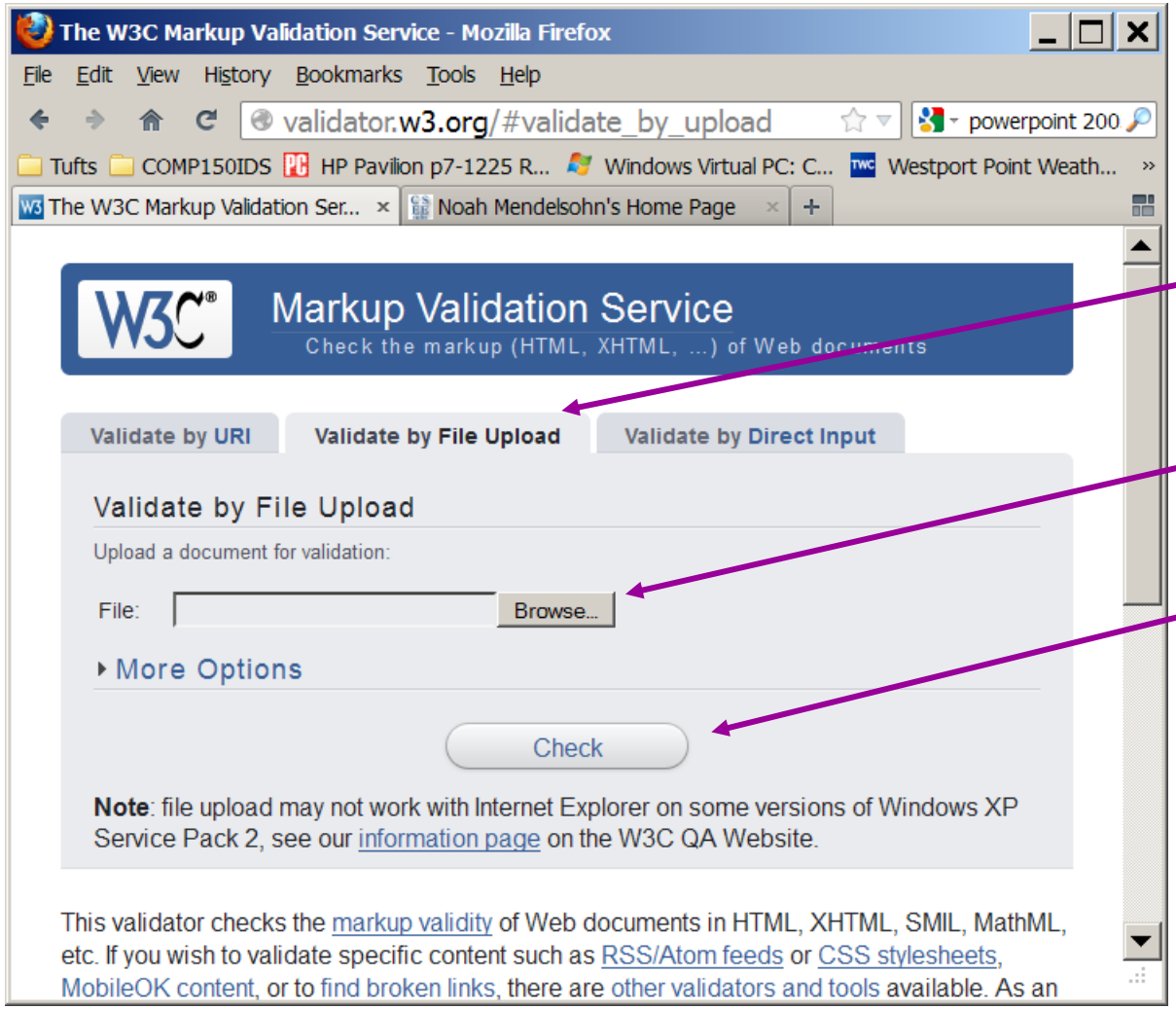
1. Go to validator.w3.org
2. Enter the URI for your page
3. Press "Check" to validate
4. You can ignore warning that the HTML5 validator is experimental

Is your CSS file *correct?*



1. Much easier to check if HTML & CSS are on Web.
2. Go to http://jigsaw.w3.org/css-validator/
3. Enter the URI for your HTML page *or* your CSS sheet
4. Press "Check" to validate CSS

Validating HTML from your hard drive



1. Go to validator.w3.org
2. Select: Validate by File Upload
3. Browse to find HTML on your local drive
4. Press "Check" to validate

Warning: the validator can find linked CSS, etc. if it's on the Web and referenced with an absolute URI. It will *not* go back to your hard drive to find more!

Homework scores may be reduced for any HTML or CSS file you submit that doesn't validate!

Publishing your HTML Files

Steps for publishing an HTML file at Tufts

- **Copy your file to `~yourUTLN/public_html/filename.html` on linux.eecs.tufts.edu**
- **Make sure there are public "execute" permissions on the public_html directory (and any subdirectories if you used them):**

```
chmod o+x ~yourUTLN/public_html
```

- **Make sure there are public "read" permissions on the file itself:**

```
chmod o+r ~yourUTLN/public_html/filename.html
```

- **If you've done this right, your file should be accessible from Web browsers all over the world as:**

```
http://www.eecs.tufts.edu/~yourUTLN/filename.html
```

If it doesn't work, there's a good chance you didn't set the permisisons right!

If it’s not working, you’ll want to check permissions

- **ls –ld <dirname>: lists permissions on named directory itself**
- **ls –l <dirname>: lists permissions on the files *in* the directory**
- **Sample output:**

```
$ ls -ld public_html
drwx--x--x+ 1 noah None 0 Aug 29 19:56 public_html
```

Directory is *executable* by anyone

```
$ cd public_html
$ ls -l testfile.html
-rw-r--r--+ 1 noah None  291328 Aug 25 20:55 testfile.html
```

HTML file is *readable* by anyone

All this is needed so the Apache Web server can get at your file.