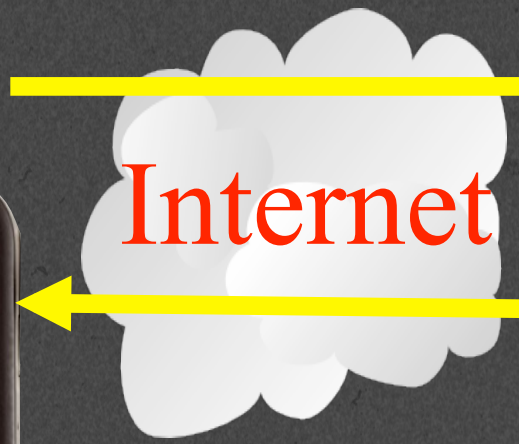
The Request/Response Cycle

Or, what happens when you type something into the address bar



HTML JavaScript

AJAX CSS

Cookies

HTTP Request

Response GET

POST

Python Data Store

Templates memcache

MVC

How Does This All Work?

- When you type an address into the URL bar, what happens?
- *Warning*: This lecture is heavy on the acronyms.

Client/Server Relationship

- Servers
  - Machines that hold shared resources
  - Always connected to the network
- Clients
  - Machines for personal use (laptops, phones, etc.)

Request/Response Cycle

- This is what happens when your computer (the client) requests a page and a server responds with the appropriate files.

Uniform Resource Locator

- URL – three parts:
  - protocol – how to connect
  - domain – where to find the document you want
  - document – what specific file is needed*
    - *Most pages are made up of multiple files*

Protocols

- HTTP – Hypertext Transfer Protocol
- HTTPS – Secure Hypertext Transfer Protocol
- FTP – File Transfer Protocol

Domain Names

- Identifies the entity you want to connect to
  - umich.edu, google.com, wikipedia.org
- Each has different top-level domain
  - Determined by Internet Corporation for Assigned Names and Numbers (ICAAN)
  - https://www.icann.org/resources/pages/tlds-2012-02-25-en

Document

- URLs can specify a specific document
  - http://www.intro-webdesign/contact.html
  - http://www.intro-webdesign/Ashtabula/harbor.html
  - Wikipedia
  - Do a pictures
- If no document is specified, the default document is returned.
  - Convention is *index.html*

The Request

- Once the IP address is determined, the browser creates an HTTP request.
- Lots of hidden information in this request
  - header, cookies, form data, etc

The Response

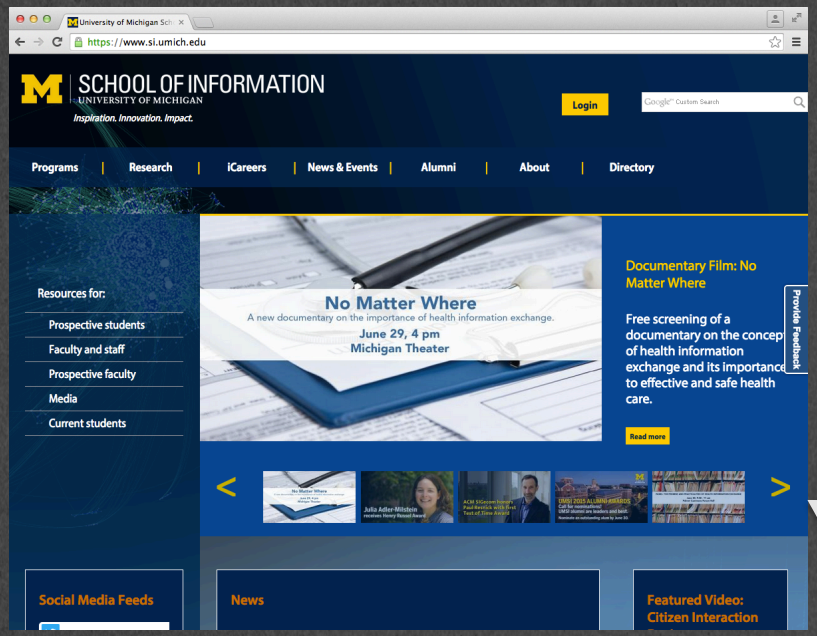
- The server returns files/output, not "web pages"
  - It is up to the browser to decide what to do with that info
- If the server can't fulfill the request it will send back error codes: 404, 500, etc.

What happens when you type "http://si.umich.edu/" into the address bar.

1. The browser look up the domain in the DNS

2. The DNS returns the IP address:54.88.175.189

The Request/ Response Cycle is initiated



3. The browser sends an HTTP request to the server located at that address.

4. The server finds the requested and sends it back as a response.

5. The browser takes the response and renders the HTML code as a nice graphical presentation, often repeating steps 3 – 5 as needed to request images and other supporting files.

Review

- A URL typically requires multiple rounds of the Request/Response cycle.
- The browser must request the main file but any supporting files and images are also returned

HTML5

What it is…and why we aren't starting at HTML 1.0

What is HTML?

- HTML stands for Hypertext Markup Language
- Markup languages are not the same as programming languages, they use *tags* to annotate documents.
- In HTML the tags indicate where headings, images, lists, links, line breaks, and other components should go.

.html files

- When your computer opens a .html file, it knows to open it in an Internet browser (Chrome, Firefox, Safari, etc.)
- The browser can read this file and know how to display it on the screen.
- Screen readers and other assistive devices can also utilize the HTML tags to present the information is special ways.

"Learning" HTML

- In the beginning you worry about *syntax*
  - What tags are there?
  - Did I remember to "end" my tag?
- Later, you will worry about *semantics*
  - Is there a tag that better conveys the meaning I am trying to get across?
  - If someone is searching my page can they find what they need and access it easily?

Early Years

- HTML (
  conn
  co
- Time
  new, eas
  there was no

It is required that HTML be a common language between all platforms. This implies no device-specific markup, or anything which requires control over fonts or colors, for example.

Mosaic

- In 1993, Mosaic emerged as the first graphical browser.
- WWW proliferates at a 341,634% annual growth rate of service traffic
- Mosiac had challengers though in the form of Netscape (1994), Internet Explorer (1995) and others.

The Browser Wars

- Browsers had proprietary tags
  - <marquee>...</marquee> (scrolling text)
  - <blink>...</blink> (blinking text).
- Other tags that went against the spirit of the original tenets of HTML were added, e.g. <font>, , and <bgcolor>
- Origination of "Best viewed on" messages.

Web Standards

- No one "runs" the Internet or the Web, some groups do take proactive roles:
  - Internet Engineering Task Force (IETF)
  - World Wide Web Consortium (W3C)
  - The Web Accessibility Initiative (WAI)

Evolution of Browsers

1990 – 1994	HTML was simple, content was primarily text-based
1993	Mosaic enters the scene with images and … BOOM!!!
1995 – 1999	Cross-browser compatibility falls apart
2000 – 2005	Browsers move toward separating content from style.
2005 – 2008	Using HTML files in coordination with CSS becomes new standard.

Evolution of HTML

1993	HTML 1.0 - Developed by Tim Berners-Lee to link document
1995	HTML 2.0 - Developed by Internet Engineering Task Force RFC to include stylized text and tables
1996	CSS 1
1997	HTML 3.2 – Developed by W3C and included browser specific features
1997	HTML 4.0 – A move back to normalizing the pages across platforms.
1998	CSS 2
1999	HTML 4.01 – Introduced different document types
2012	HTML 5 - Back to HTML plus multimedia and semantic tags

Where we are now

- HTML5 is a cooperation between W3C and the Web Hypertext Application Technology Working Group(WHATWG)

- Established Guidelines
  - New features should be based on HTML, CSS, the DOM, and JavaScript
  - Reduce the need for external plugins (e.g. Flash)
  - More markup to replace scripting
  - HTML5 should be device independent

Review

- Browsers translate HTML documents into viewable webpages
- HTML was intended to facilitate content types
- When designers want to do something new they write non-standard code to force browsers to do it
- New standards are written to handle new requirements and browsers adopt the new standards

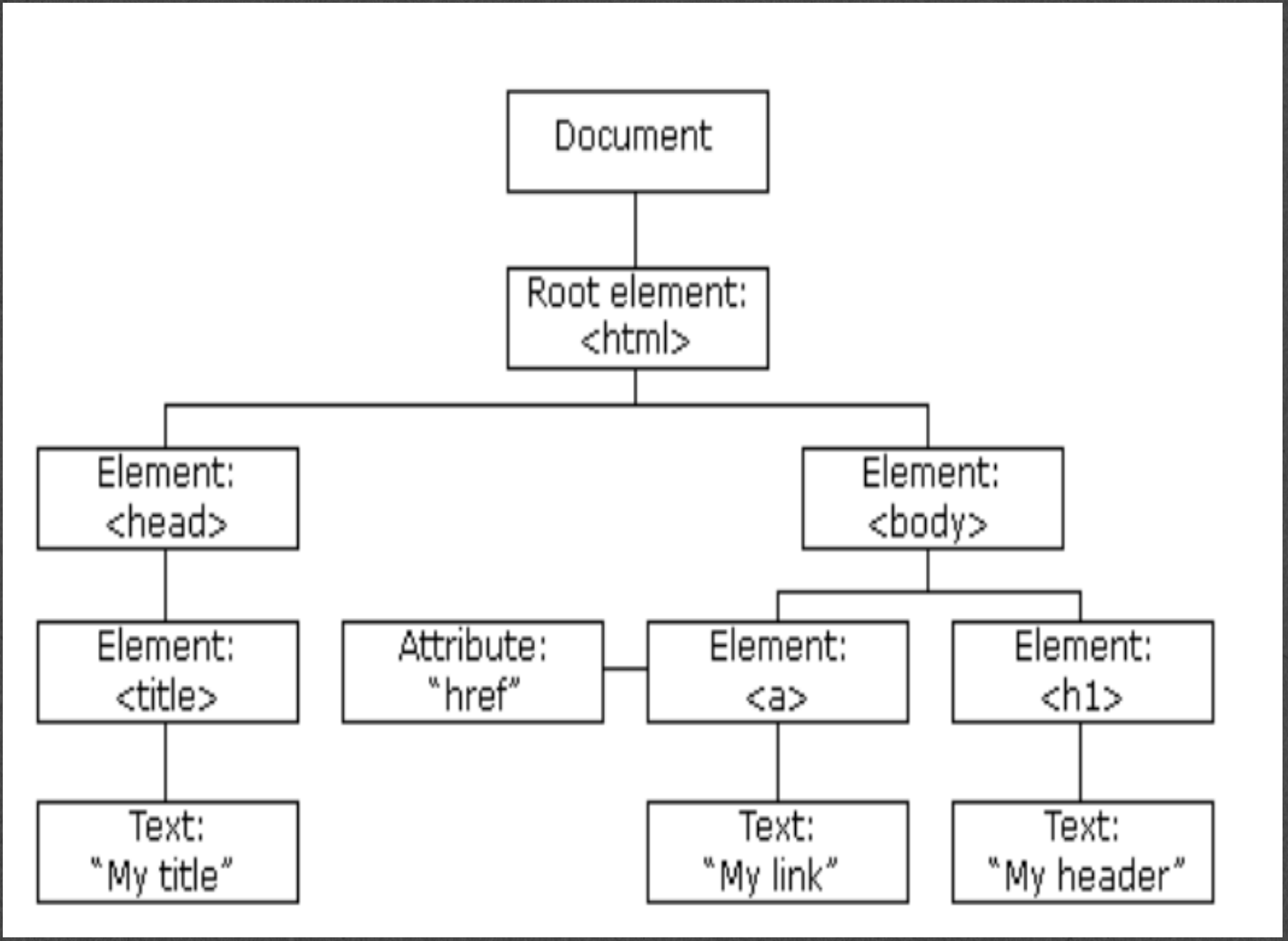
Document Object Model

Writing clean code

The Document Object Model (DOM)

- Basis of HTML5 is "*New features should be based on HTML, CSS, the DOM, and JavaScript…*"
- DOM provides common tree-like structure that all pages should follow
- Computer Scientists love trees (the mathematical kind) because you can test them.

HTML is built on the DOM



http://www.w3schools.com/js/js_htmldom.asp

Three parts of a well-formed document

- Doctype
  - Version of HTML that you will be using
- Head
  - Metadata
- Body
  - Displayable content

Doctype

- HTML5
  - <!DOCTYPE html>
- Previous versions dictated backwards compatibility
  - <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
  - <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

Head

- Additional information used by the browser
  - Meta data – language, title
  - Supporting files – JavaScript, Styling, Add-ons
- Other than title, meta-data is not displayed

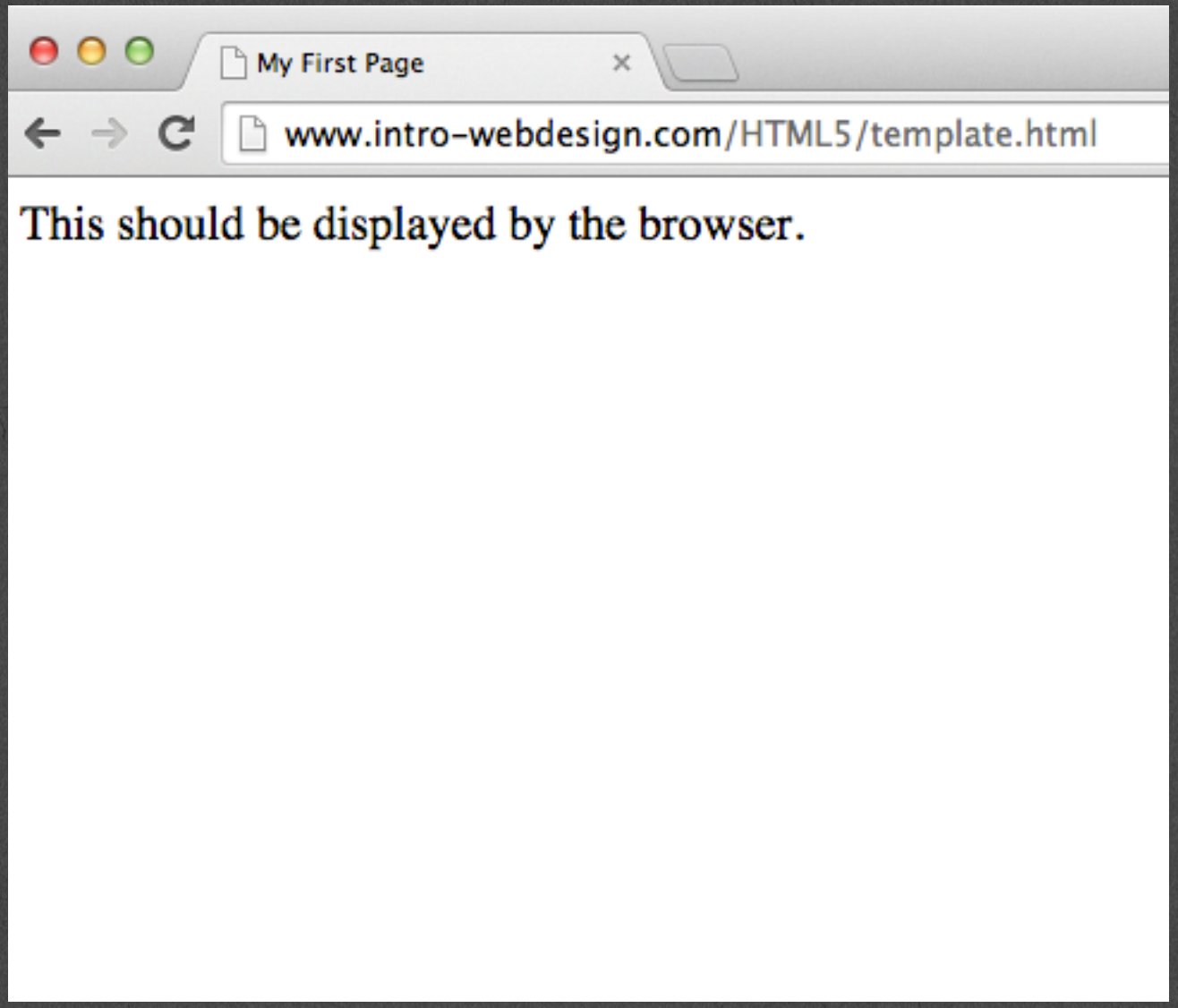
Body

- Bulk of your page
- Important to write well-formatted (tree-like) code.
- Most of the content is displayed by the browser, but there may be some meta-data too

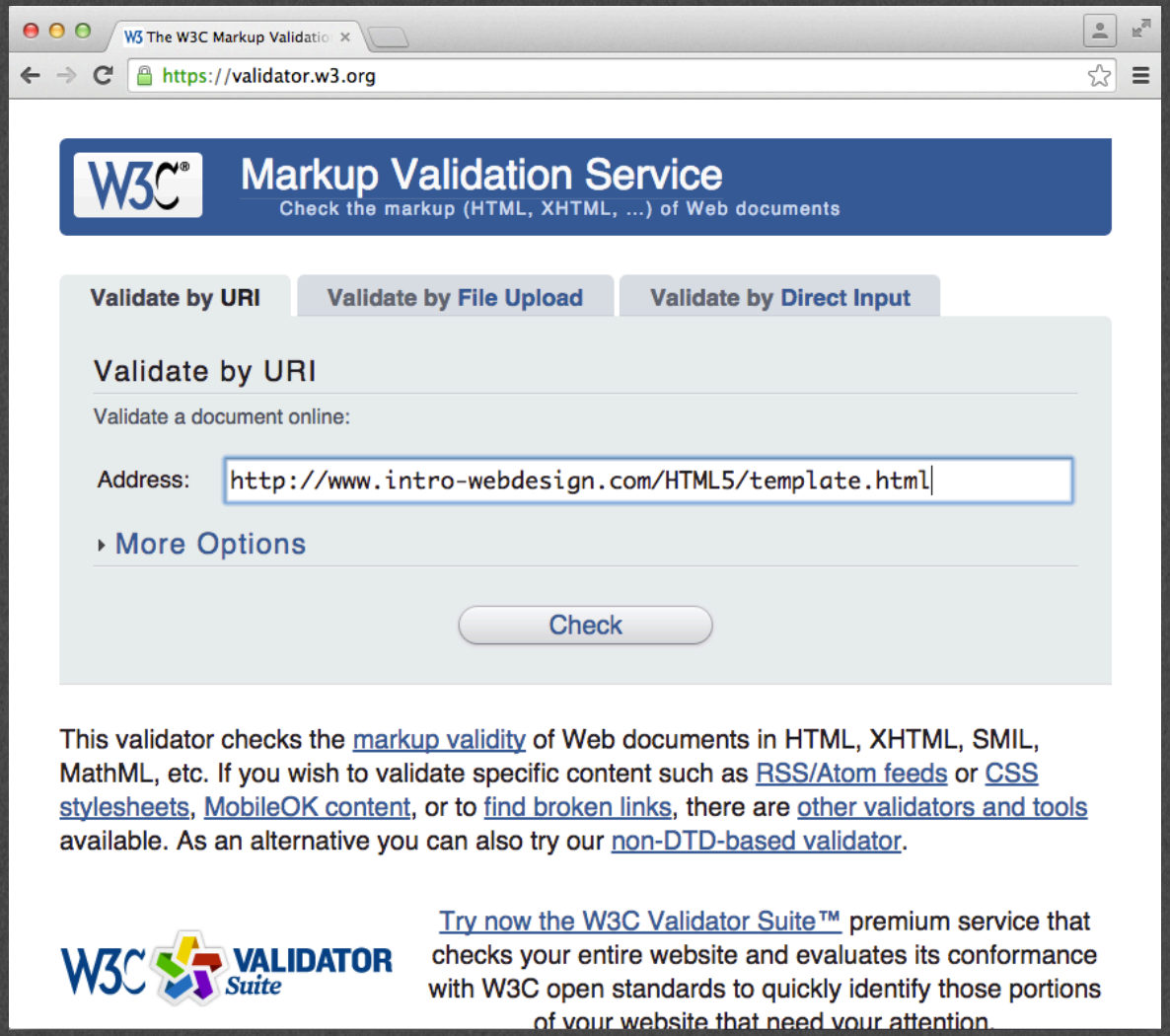
Example

- **Example: template.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>My First Page</title>
</head>
<body>
    This should be displayed by the browser.
</body>
</html>
```


My First Page
www.intro-webdesign.com/HTML5/template.html
This should be displayed by the browser.

Validate the Code



Success!!

This document was successfully checked as HTML5!

Result:	Passed, 2 warning(s)	
Address :	http://www.intro-webdesign.com/HTML5/template.htm	
Encoding :	utf-8	(detect automatically)
Doctype :	HTML5	(detect automatically)
Root Element:	html	

HTML5 Tags and Syntax

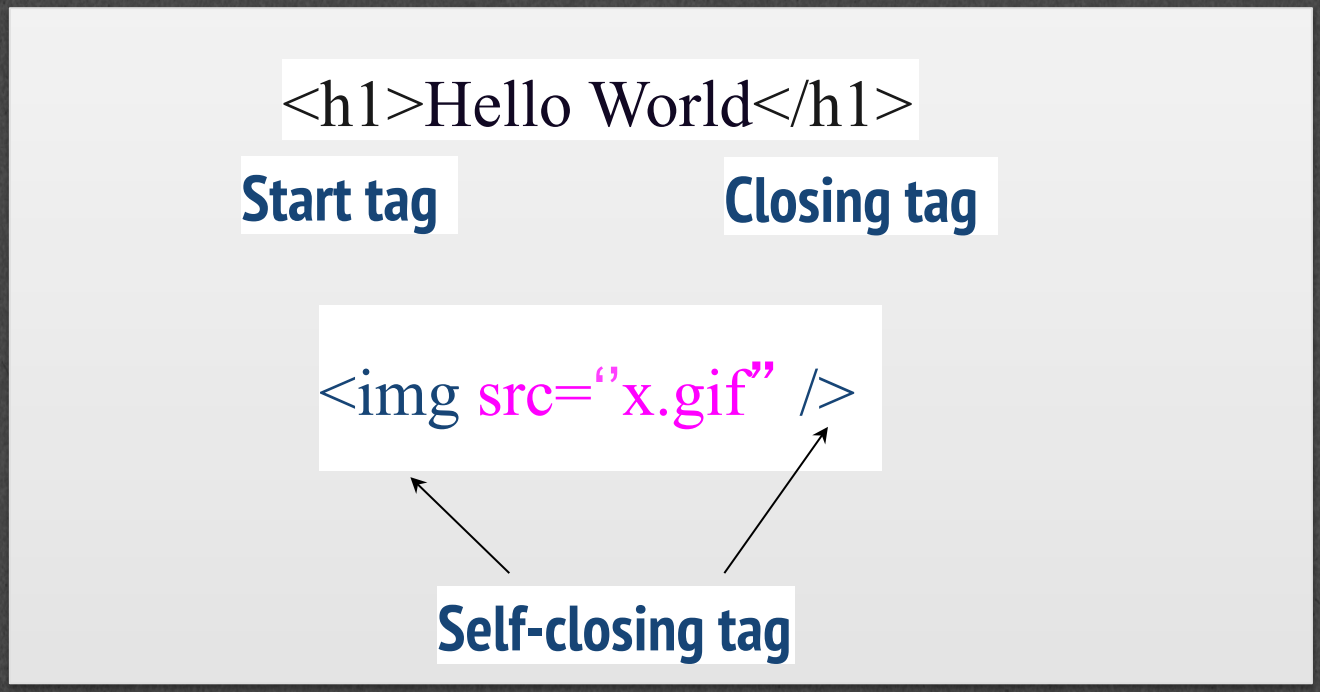
My first big disappointment to you.

HTML tags

- I can't teach you all of the tags
- I can't teach you all of the tags
- You don't want me to teach you all of the tags

Finally, some tags…

- Tags have a beginning and an end



- Some tags have *attributes* (src, href, etc..)

Display

- One of the most important attributes of an element is its display. The two most common are *block* and *inline*
  - block (can take width and height)
    - *Newline is inserted before and after, e.g. it "Takes up" whole width*
  - inline (can not take width and height)
    - *Only uses as much space as needed to contain the element.*

Common Tags

- Headings (block)
  - <h1>, <h2>, <h3>, <h4>, <h5>, <h6>
  - These tags have syntax and semantics
- Paragraphs (block)
  - ….
  - Should only contain inline elements
- Divs (block)
  - ...
  - Generic section that is larger than a paragraph

More tags

- Ordered lists

```
<ol>
    <li> Item One </li>
    <li> Item Two </li>
</ol>
```

- Unordered lists

```
<ul>
    <li> Item One </li>
    <li> Item Two </li>
</ul>
```

- Line breaks

```
<br/>
```

Attributes

- Attributes provide additional information about an element
- Always specified in the start tag
- Attributes come in name/value pairs

Images

- Images (inline)

- Images rarely work the first time
  - Show a broken link, too big, too small, etc.

- Save yourself heartache and size/carefully name your picture before you use it.

Images

<img src=”logo.jpg”	Image filename
alt="company logo"	Info for screen readers, broken links
title = "AAA1 LLC"	Displays on hover
class = "thumbnail"/>	Extra formatting (height, width, position, etc.)

More Attributes

- As you learn the tags, you learn their specific attributes. Some apply to any tag
  - class – applies special properties to groups of elements
  - id – specifies a unique id to one element on the page
  - style – specifies a certain visual style (avoid this one!!!)
  - accesskey – a shortcut key to activate an element
  - tabindex – the order that the tab butotn brings elements into focus

Hyperlinks

Creating a linked document

Links

- Links are what make the Web a web.
- The interlinked nature of the web leads to the "knowledge" that search engines appear to have.

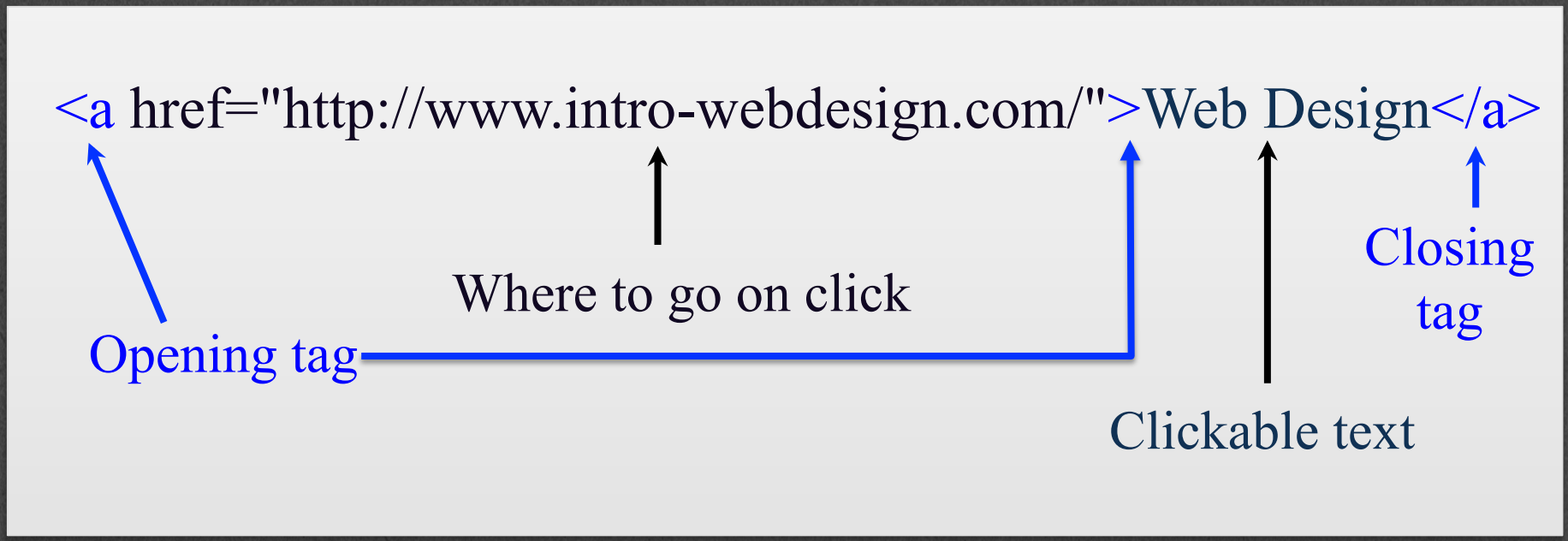
Types of links

- External
- Absolute
- Relative
- Internal
- Graphical

Anchor links

- The <a> tag stands for anchor link
- Links have one attribute within the tag AND need additional content
  - Attribute: reference to location of new content
  - Content is the "clickable" part, may be text or image.

Absolute reference



Relative References

Second Page

Local file in the same folder

Second Page

Local file in a different folder

History section

Segment in this same file that has been identified as history section

Absolute vs Relative

- When would you use absolute links?
- Are there any benefits to using local links?
- Your links should NEVER have folders that are specific to your computer

Using Images as the Link

- The "clickable" component doesn't have to be text.

```
<a href="http://www.redcross.org">
  <img src = "http://www.redcross.org/images/redcross-logo.png"
    alt = "Red Cross logo"/>
</a>
```

```
<a href="http://www.redcross.org">
  <img src = "imgs/redcross-logo.png" alt = "Red Cross logo"/>
</a>
```

Usability Issues

- Make sure the clickable component has an informative name
- Information in the images should be available to those who can't see the image

Review

- A page without links is rare
- Links can be external, internal, and embedded
- Use caution when using images in links

Special Entities

- Tags always start with a bracket (<)
- What if you want the browser to display a bracket, not start a tag?

Special Entities

If you want….	Then use…
<	&lt;
>	&gt;
©	&copy;
blank space	
¢	&cent;
&	&amp;

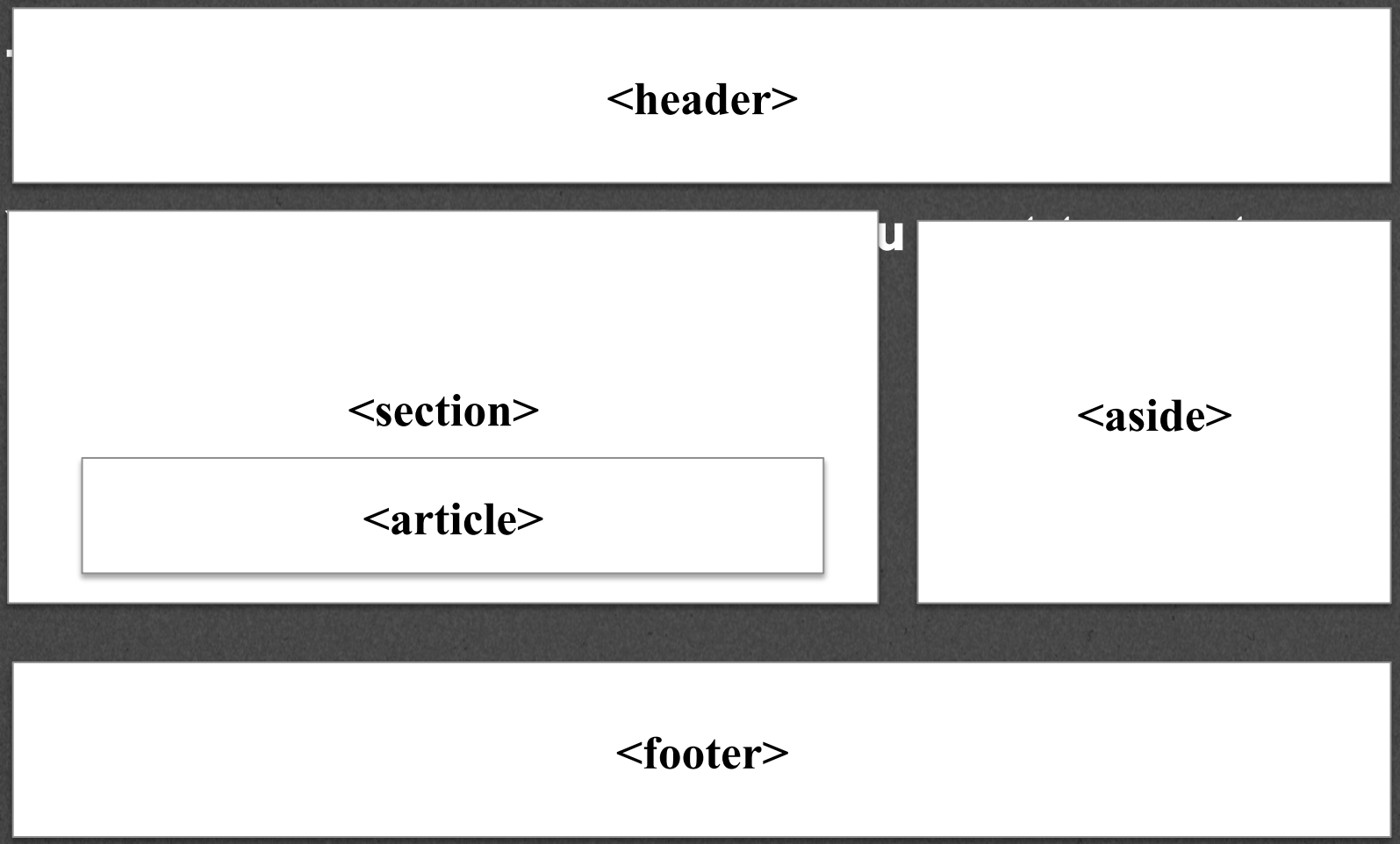
Review

- How do you know the difference between a tag and an attribute?
- What two symbols end a self-closing tag?

Semantic HTML5 Tags

How to Design

-
- u



Using Semantic Tags

- In the beginning (insert dramatic music of your choice…) there was div
- was a way to group related content together
- Divs almost always had special classes/ids associated with them

```
<div class = “header”>…</div>
<div class = “navigation”>…</div>
<div class = “footer”>…</div>
```

<header>

- A group of introductory or navigational aids: title, navigation links,

```
<header>
    <h1>This is the Title</h1>
    <h2>The author is Colleen</h2>
</header>
```

- Not to be confused with <head> or the different headings.

<nav>

- A section of the page that links to other pages or to parts within the page.

```
<nav>
    <ul>
        <li><a href="#overview">Overview</a></li>
        <li><a href="#history">History</a></li>
        <li><a href="#development">Development</a>
    <ul>
</nav>
```

- Often found in the <header> tag

<footer>

- A section that contains info such as copyright data, related documents, and links to social media

```
<footer>
   &copy; 2015 by Colleen van Lent<br>
   <a href="http://www.intro-webdesign.com/HTML5">Introduction to
   HTML5</a>
</footer>
```

- Typically at the bottom of the page, but not required.

<figure>

- More semantics than image. Can include:
  - caption
  - multiple multi-media resources

```
<figure>
    <img src="sunset.gif" alt = "Ashtabula sunset">
    <figcaption> A sunset over Lake Erie.  Taken in
        Ashtabula Ohio</figcaption>
</figure>
```

Other New Tags

- Structural Elements
  - article, aside, main, menuitem, summary, section
- Form Elements
  - datalist, keygen, output
- Input Types
  - color, date, email, list
- Graphics Elements
  - canvas, svg
- Media Elments
  - audio, embed, source, track, video

Acknowledgements/Contributions

Initial Development: Colleen van Lent , University of Michigan School of Information