CS 337
Web Programming

HTML/CSS/HTTP/JavaScript/PHP/MySQL/
CGI/XML/JSON/HTTPD/W3C/OMG No Moar TLAs

HTML

What The Web Looks Like

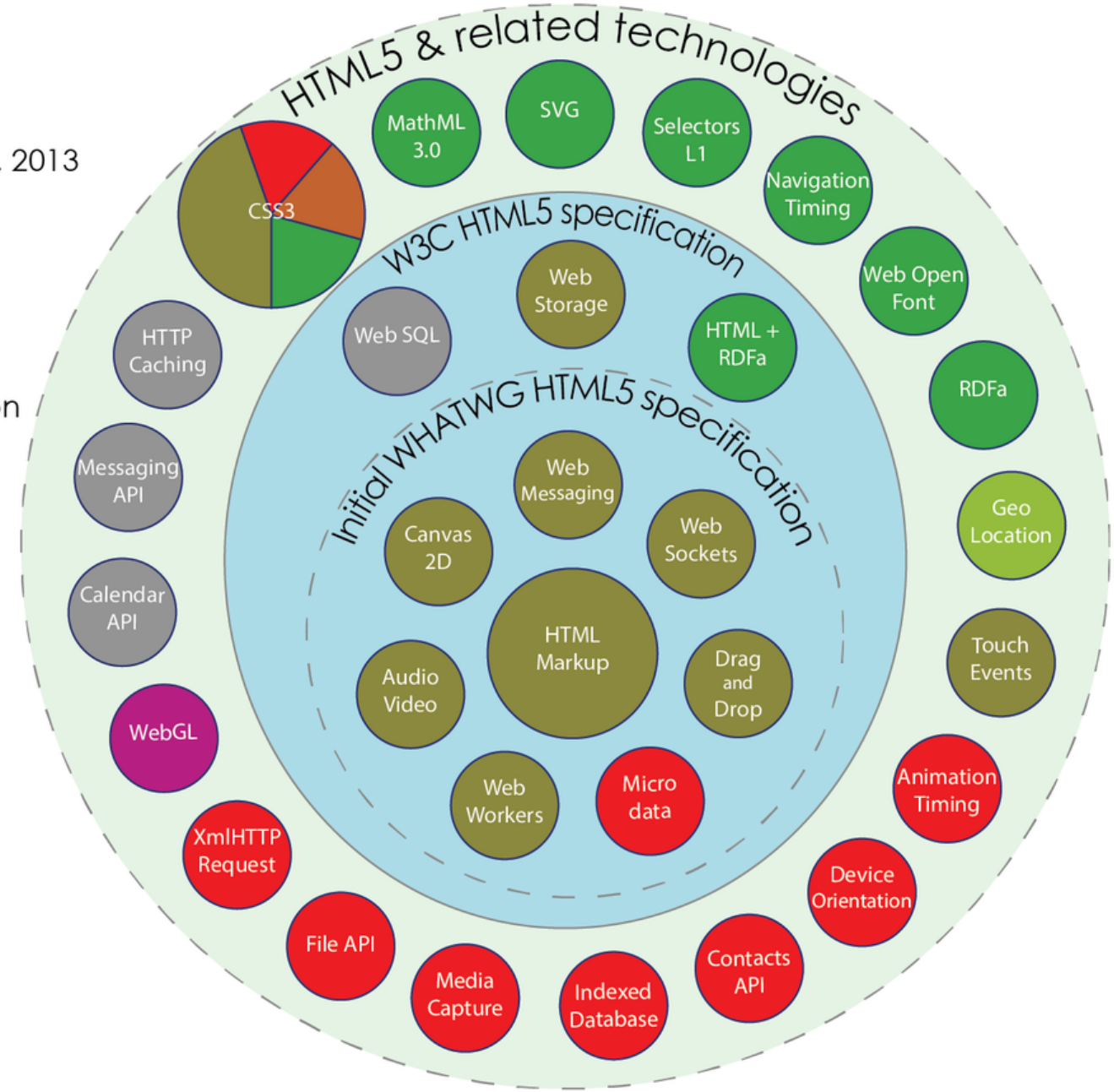
HTML

Hyper**T**ext **M**arkup **L**anguage

HTML5

Taxonomy & Status on January 20, 2013

- W3C Recommendation
- Proposed Recommendation
- Candidate Recommendation
- Last Call
- Working Draft
- Non-W3C Specifications
- Deprecated



by Sergey Mavrody CC BY · SA

http://en.wikipedia.org/wiki/File:HTML5-APIs-and-related-technologies-by-Sergey-Mavrody.png

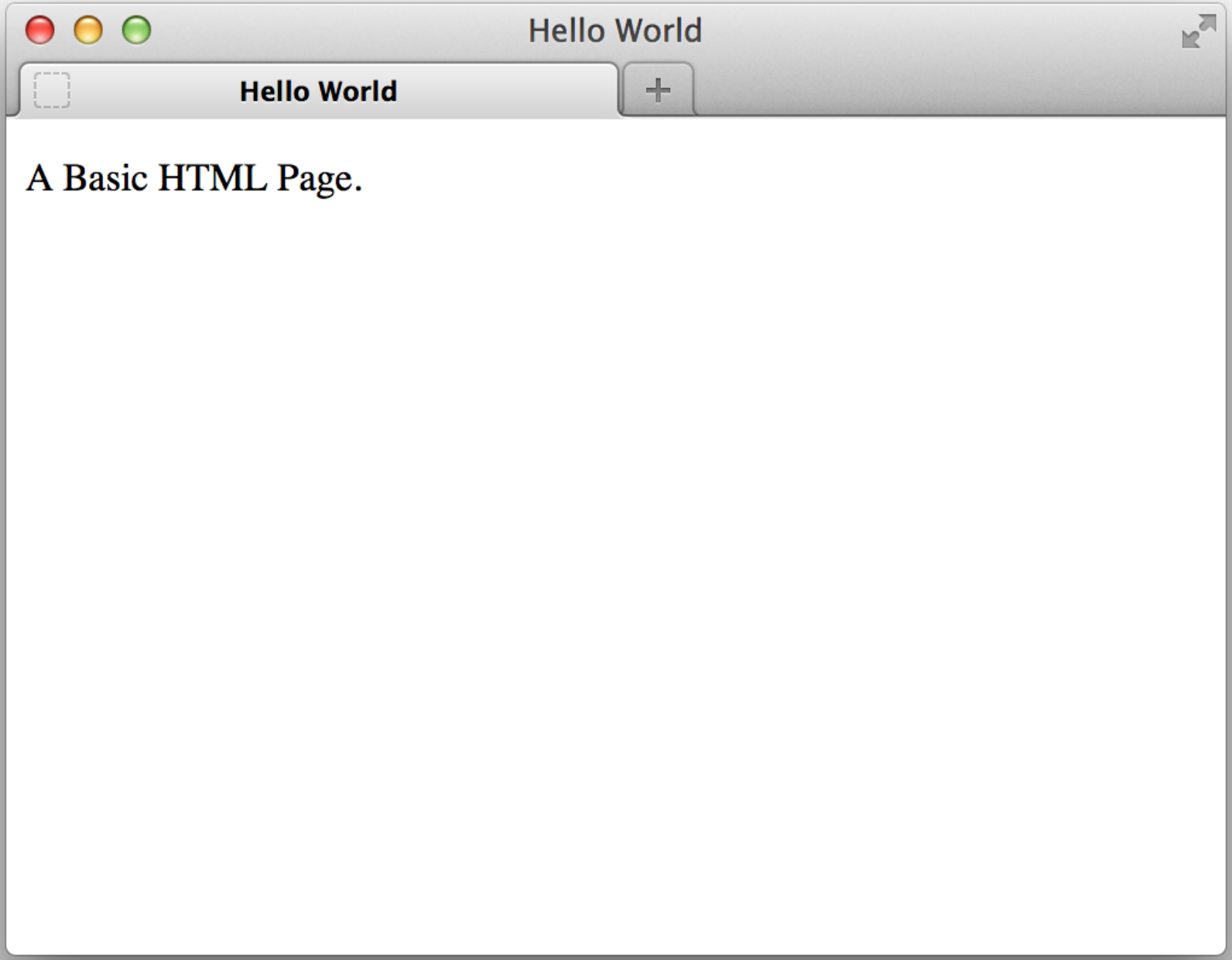
Some HTML

/some.html

```
<!doctype html>
<html>
<head>
  <title>Hello World</title>
</head>

<body>
  <p>A Basic HTML Page.</p>
</body>

</html>
```


Hello World
Hello World
A Basic HTML Page.

HTML Defines Content and Structure

- Content consists of Text, Images, Links, Media Assets, etc
- Structure defines the basic formatting and semantic meaning of elements
  - i.e. `<title>Hello World</title>` defines the title of the page
  - Programs can analyze the structure of a document to derive meaning
    - `h1`, `h2`, `h3` tags could be used to generate a document outline
    - Headers in a table (`<th>`) could be used by screen readers to describe data to a visually impaired individual
  - We can use the document structure to define display styles

Structure of an Element

```
<title>Hello World</title>
```

- The entire line is referred to as ***the*** `title` ***element***
- The ***name*** of this element is “title”
- `<title>` is an ***opening tag***
- `</title>` is a ***closing tag***
- `Hello World` is the ***content*** of this element

Not All Elements Need a Closing Tag

```
<body>
  <p>
   Paragraph elements can have closing tags
  </p>
  <p>or not
  <ul>
    <li>List Item elements
    <li>may also omit closing tags
  </ul>
</body>
```

http://www.w3.org/TR/html5/syntax.html#optional-tags

Question…

```
<body>
  Is this first bit of text the
  content of a Paragraph element?
  <p>
  What about this second bit of text?
</body>
```

The HTML Docs

http://www.w3.org/TR/html5/

```
<body>
  Is this first bit of text the
  content of a Paragraph element?
  <p>
  What about this second bit of text?
</body>
```

- Is the first bit of bare text a legal content for the <body> tag?
- Where in the HTML specification is this question answered?

HTML vs XHTML

- HTML **is not** XML compliant
  - No namespaces
  - More forgiving parsing
- XHTML **is** XML compliant
  - Namespaces
  - Very strict document parsing. Any errors usually result in the document completely failing to be displayed
  - Not allowed to omit closing tags
  - *void elements* must be self closed
    - `<br/>`

Not All Elements Have Content

- `<br>` the Break tag acts as a newline character for HTML
- `<hr>` the Horizontal Rule tag draws a line across a page
- `<img src="foo.gif">` the Image Tag tells the browser to go load an image in this location
- These elements are called ***void elements*** and *must not have* closing tags

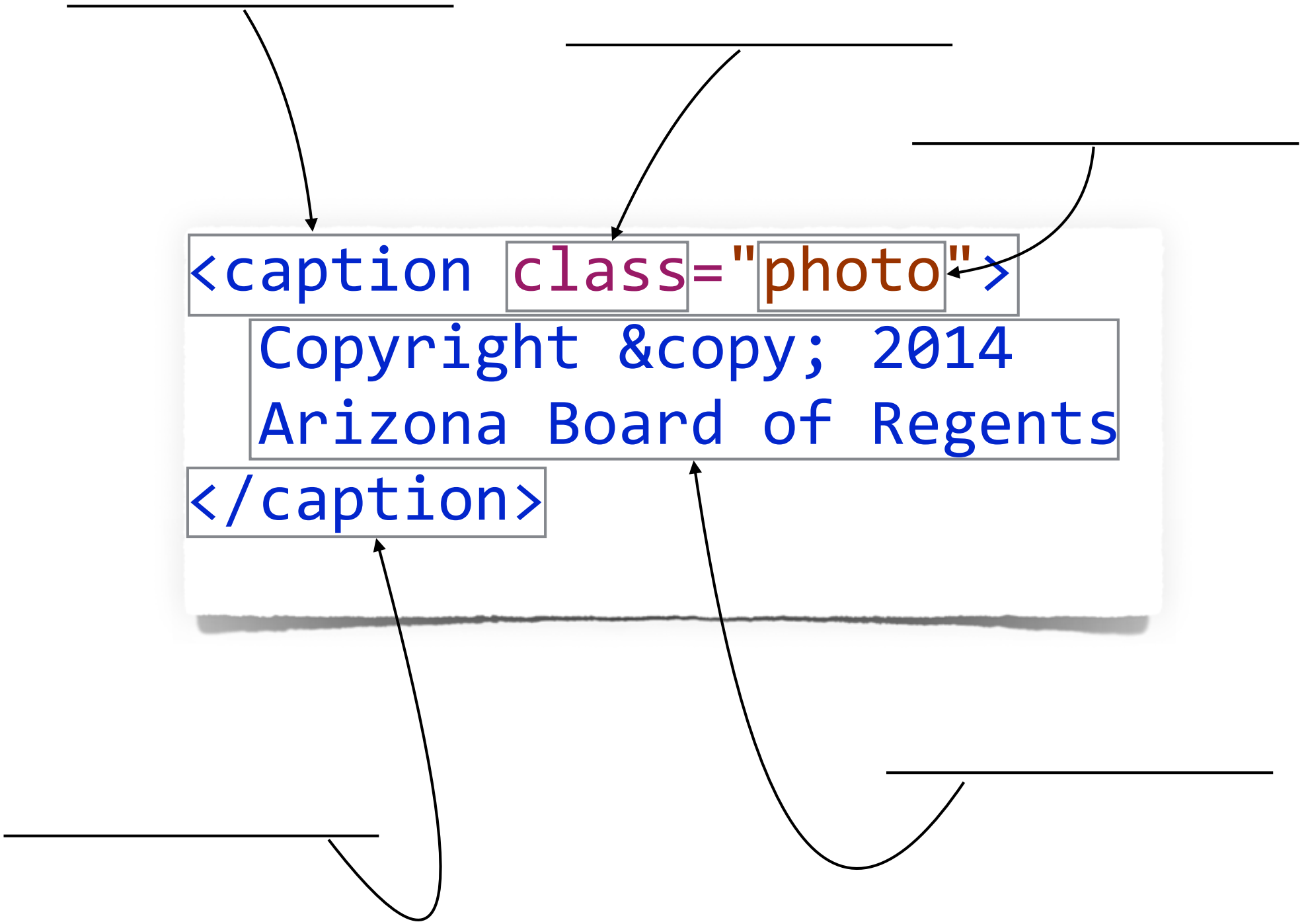
http://www.w3.org/TR/html5/syntax.html#void-elements

Attributes

```
<img src="foo.gif" class=thumbnail>
```

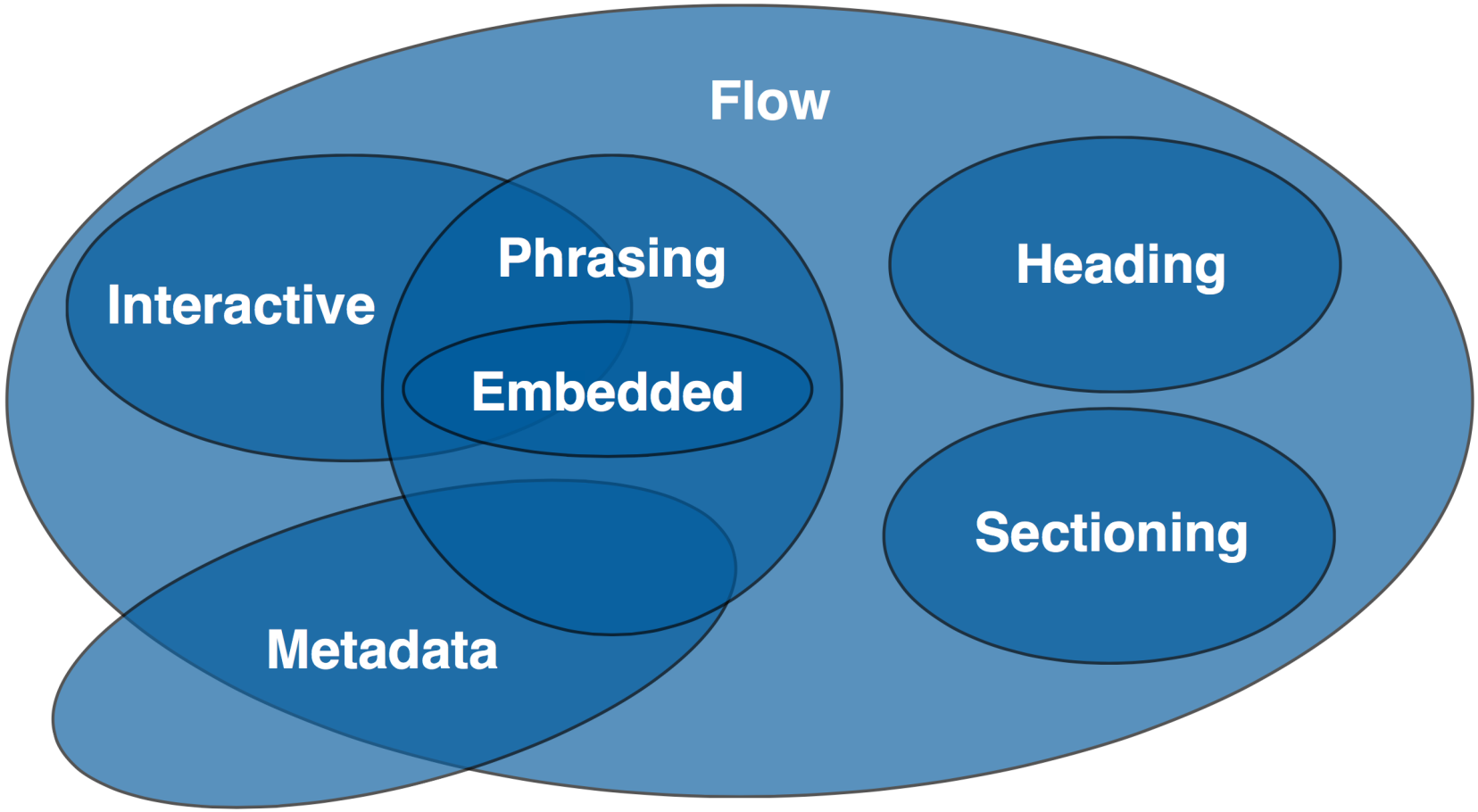
- Attributes for an element are defined in the element's ***opening tag***
- Attributes always have an ***attribute name***
- Attributes may optionally have a ***value***
- Attribute values may be surrounded with either single quotes, double quotes, or nothing, depending on the content of the value

http://www.w3.org/TR/html5/syntax.html#attributes-0


<caption class="photo">
Copyright © 2014
Arizona Board of Regents
</caption>

HTML Elements

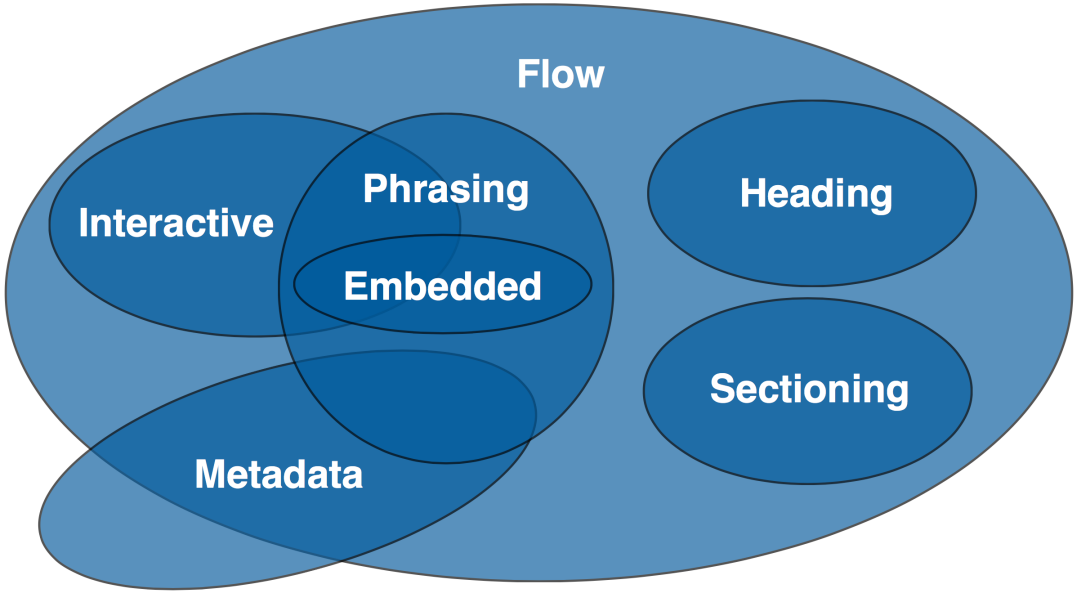
Categories of Elements



http://www.w3.org/TR/html5/dom.html#kinds-of-content

Categories of Elements

- Elements may belong to multiple categories
- We'll mostly be dealing with Flow and Phrasing elements



Categories of Elements

- HTML Specification tells us the categories an element belongs to, as well as the types of child elements allowed.

4.4.1 The `p` element

Categories:

Flow content.

Palpable content.

Contexts in which this element can be used:

Where flow content is expected.

Content model:

Phrasing content.

Content attributes:

Global attributes

Tag omission in text/html:

A `p` element's end tag may be omitted if the `p` element is immediately
`h6`, `header`, `hgroup`, `hr`, `main`, `nav`, `ol`, `p`, `pre`, `section`, `table`, or `ul`, ele

Allowed ARIA role attribute values:

Any role value.

Allowed ARIA state and property attributes:

Global aria-* attributes

Any `aria-*` attributes applicable to the allowed roles.

DOM interface:

IDL

```
interface HTMLParagraphElement : HTMLElement {};
```

http://www.w3.org/TR/html5/grouping-content.html#grouping-content

Flow Elements

3.2.4.1.2 Flow content

Most elements that are used in the body of documents and applications are categorized as **flow content**.

- a abbr address area (if it is a descendant of a map element) article aside audio b bdi bdo blockquote br button canvas cite code data datalist del dfn div dl em embed fieldset figure footer form h1 h2 h3 h4 h5 h6 header hr i iframe img input ins kbd keygen label main map mark math meter nav noscript object ol output p pre progress q ruby s samp script section select small span strong sub sup svg table template textarea time u ul var video wbr text

http://www.w3.org/TR/html5/dom.html#flow-content-1

Phrasing Elements

3.2.4.1.5 Phrasing content

Phrasing content is the text of the document, as well as elements that mark up that text at the intra-paragraph level. Runs of phrasing content form paragraphs.

- a abbr area (if it is a descendant of a map element) audio b bdi bdo br button canvas cite code data datalist del dfn em embed i iframe img input ins kbd keygen label map mark math meter noscript object output progress q ruby s samp script select small span strong sub sup svg template textarea time u var video wbr text

Most elements that are categorized as phrasing content can only contain elements that are themselves categorized as phrasing content, not any flow content.

http://www.w3.org/TR/html5/dom.html#phrasing-content-1

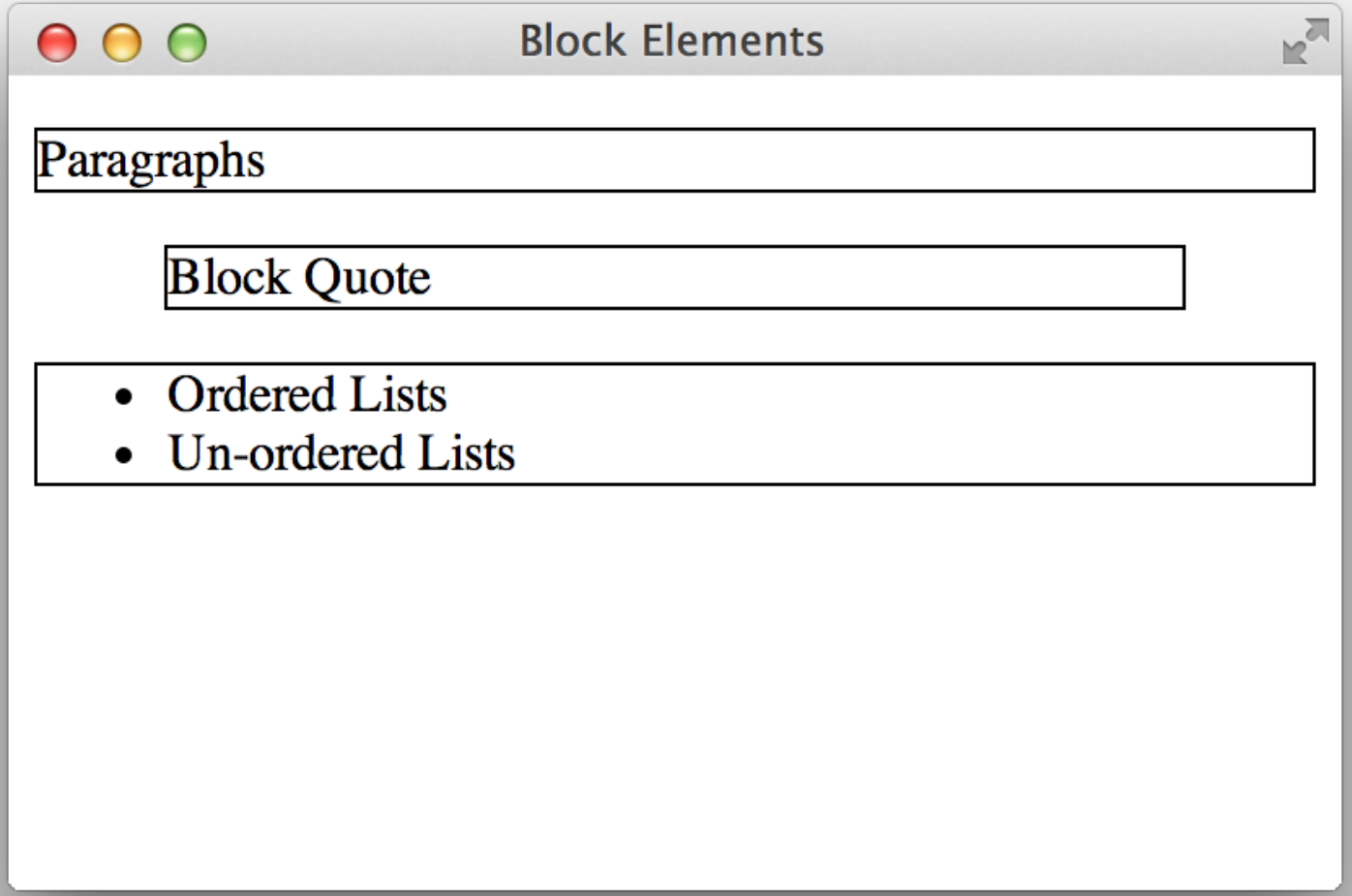
Pre-HTML5

- Pre HTML5, most elements were loosely grouped into "**Block Elements**" and "**Inline Elements**"
- HTML5 strives to better separate the logical structure of a document from the display
- Block Elements were mostly grouped by how they were displayed in a browser, i.e. expanding to fill the available width of a container.
- Browsers will likely continue to render them like this, but it is no longer mandated by the specification

Block Elements

```
<!doctype html>
<html>
<head>
  <title>Block Elements</title>
  <style>
    p, blockquote, ul { border: 1px solid black; }
  </style>
</head>
<body>
  <p>Paragraphs</p>
  <blockquote>Block Quote</blockquote>
  <ul>
    <li>Ordered Lists</li>
    <li>Un-ordered Lists</li>
  </ul>
</body>
</html>
```

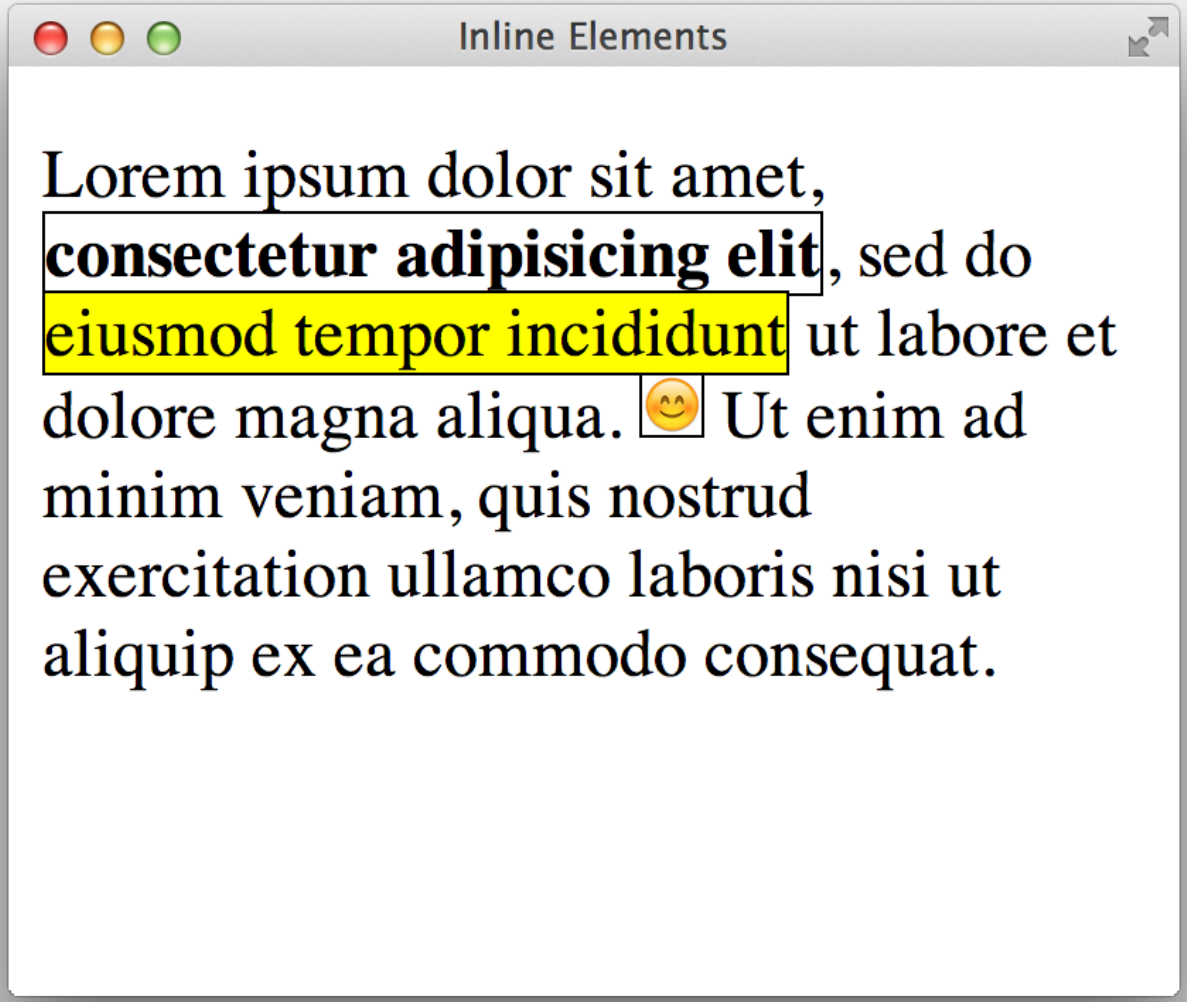
Block Elements



Inline Elements

```
<!doctype html>
<html>
<head>
  <title>Inline Elements</title>
  <style>
    strong, mark, img { border: 1px solid black; }
    img { width: 14px; }
  </style>
</head>
<body>
  <p>Lorem ipsum dolor sit amet, <strong>consectetur adipisicing
elit</strong>, sed do <mark>eiusmod tempor incididunt</mark> ut
labore et dolore magna aliqua. <img src="img/smile.png" alt="smile">
Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris
nisi ut aliquip ex ea commodo consequat.</p>
</body>
</html>
```

Inline Elements



<!doctype …>

- The `<!doctype …>` preamble is *not* an HTML element.
- `<!doctype …>` tells the rendering engine what type of markup to expect
- HTML4.1 Transitional
  - `<!doctype html public "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">`
- HTML5
  - `<!doctype html>`

DOCTYPEs are required for legacy reasons. When omitted, browsers tend to use a different rendering mode that is incompatible with some specifications. Including the DOCTYPE in a document ensures that the browser makes a best-effort attempt at following the relevant specifications.

http://www.w3.org/TR/html5/syntax.html#the-doctype

<html>

- The <html> element is the root element of our element tree
- The HTML Element can only be preceded by whitespace characters and comments
- The HTML Element can only have two children: one <html> element and one <body> element
- From the HTML specification:
  - *An html element's start tag can be omitted if the first thing inside the html element is not a comment.*
  - *An html element's end tag can be omitted if the html element is not immediately followed by a comment.*

<head>

- The <head> element represents a collection of metadata for the Document.

- A <title> tag is the only required child element

```
<head>
  <meta charset="utf-8">
  <base href="http://www.example.com/">
  <title>A New Hope</title>
  <link rel="stylesheet" href="default.css">
  <script src="example.js"></script>
</head>
```

<body>

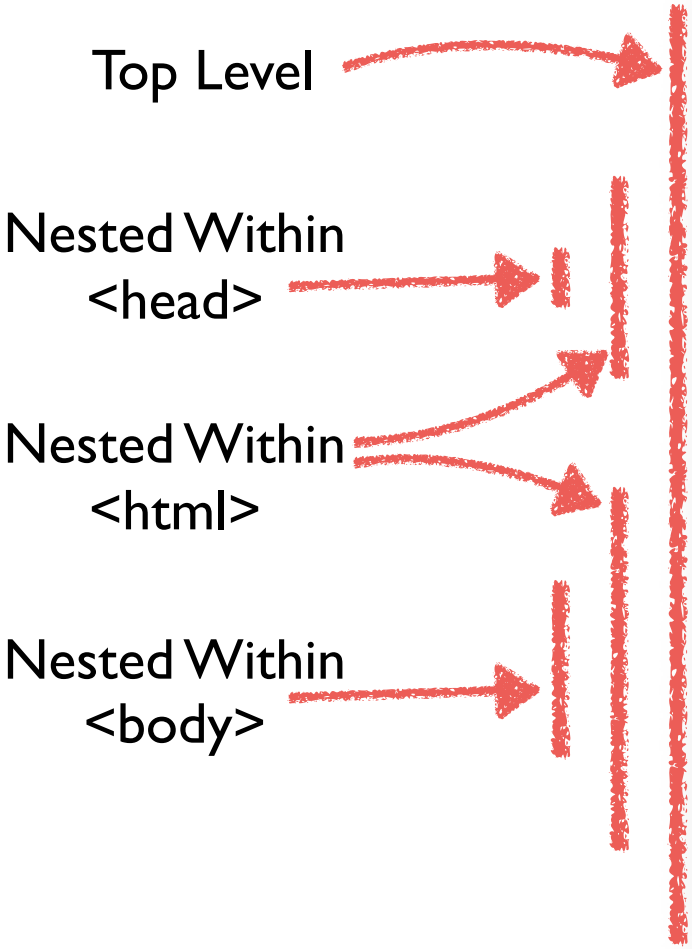
- The <body> element represents the content of the Document.

- Basically this holds everything you see.

```
<body>
  <img src="http://goo.gl/5LEYnp" alt="regrets">
  <form action="search.php" method="post">
    <input type="text" name="search">
    <input type="submit" value="Find Droids">
  </form>
</body>
```

Nested Elements

- We've already seen a lot of nested elements

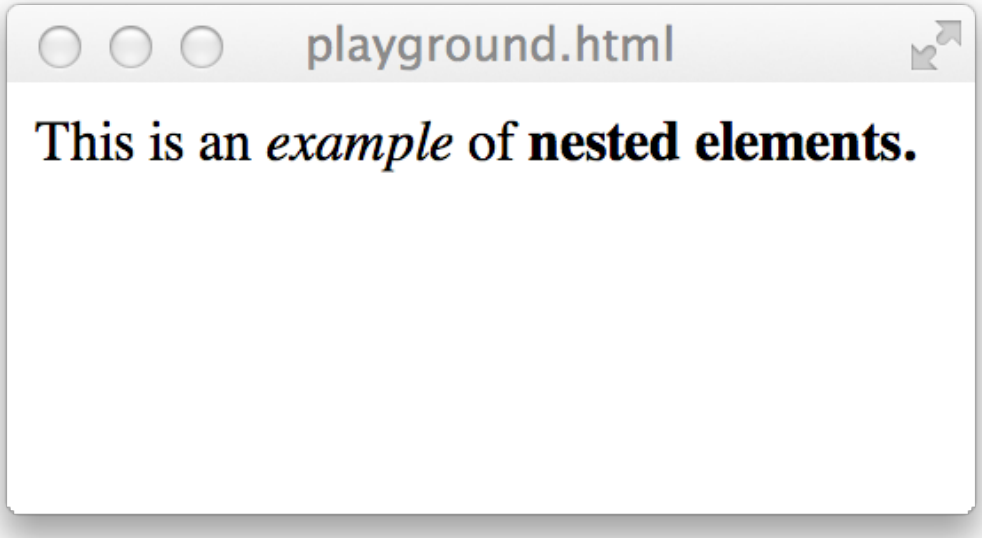



```
<!doctype html>
<html lang="en">
  <head>
    <title>Validator!</title>
  </head>

  <body>
    <p>
      Let's Test it
    </p>
  </body>
</html>
```

Nested Elements

```
<p>This is an <em>example</em> of
<strong>nested elements</strong>.</p>
```



Nested Elements

```
<p>This is an <em>example</em> of
<strong>nested elements</strong>.</p>
```

- Nested element tags must be balanced in a last-in, first-out manner.
- A close tag must always balance the most recent still-open tag.
- After the word “example” if we want to close something, we must close the `<em>` tag first.

Nested Elements

- What does the browser do if the elements are incorrectly nested?

```
<p>This is an <em>example</p> of
<strong>incorrectly nested elements</em>.</strong>
```

Links

```
Come visit the <a href="http://www.arizona.edu">University of Arizona</a> campus.
```

- `<a>` Anchor tag
- Used to define a link to another document, or location in the same document.

Links

```
<a href="http://www.arizona.edu">University of Arizona</a>
```

- `href` attribute defines what to link to.
  - This is the *Hyper* in HyperText
- Must contain a valid URL
  - Universal Resource Locator

URL

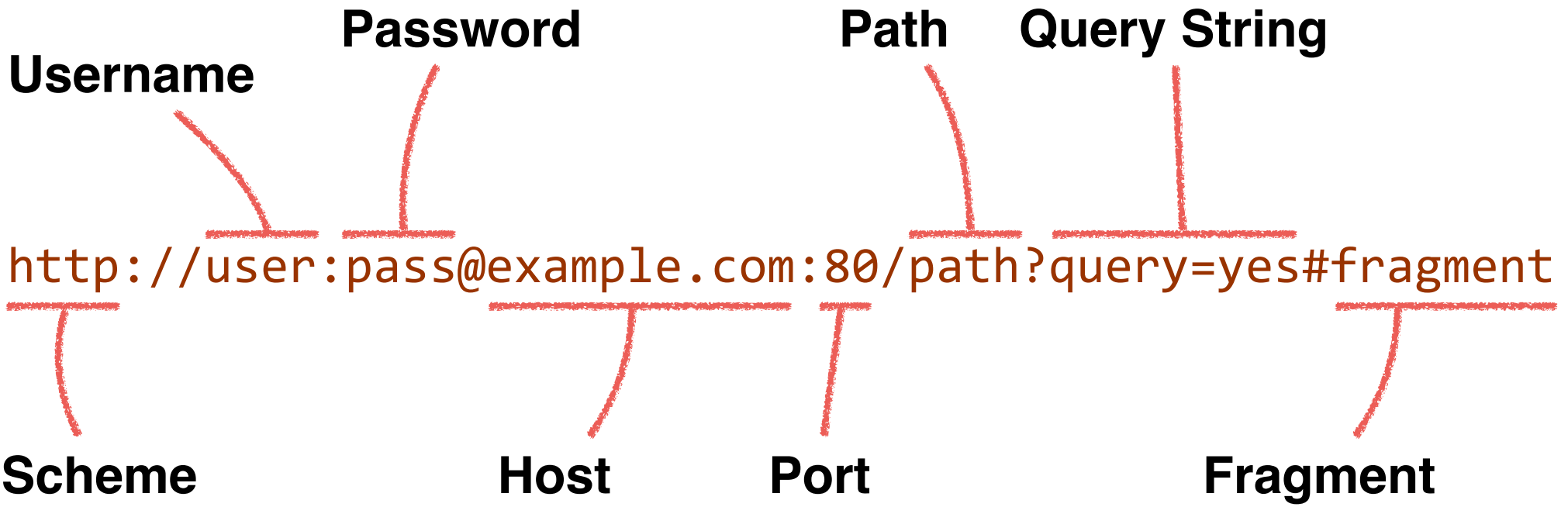
- A basic absolute URL

```
http://www.arizona.edu
```

- A basic relative URL

```
../images/image.png
```

URL


Username
Password
Path
Query String
http://user:pass@example.com:80/path?query=yes#fragment
Scheme
Host
Port
Fragment

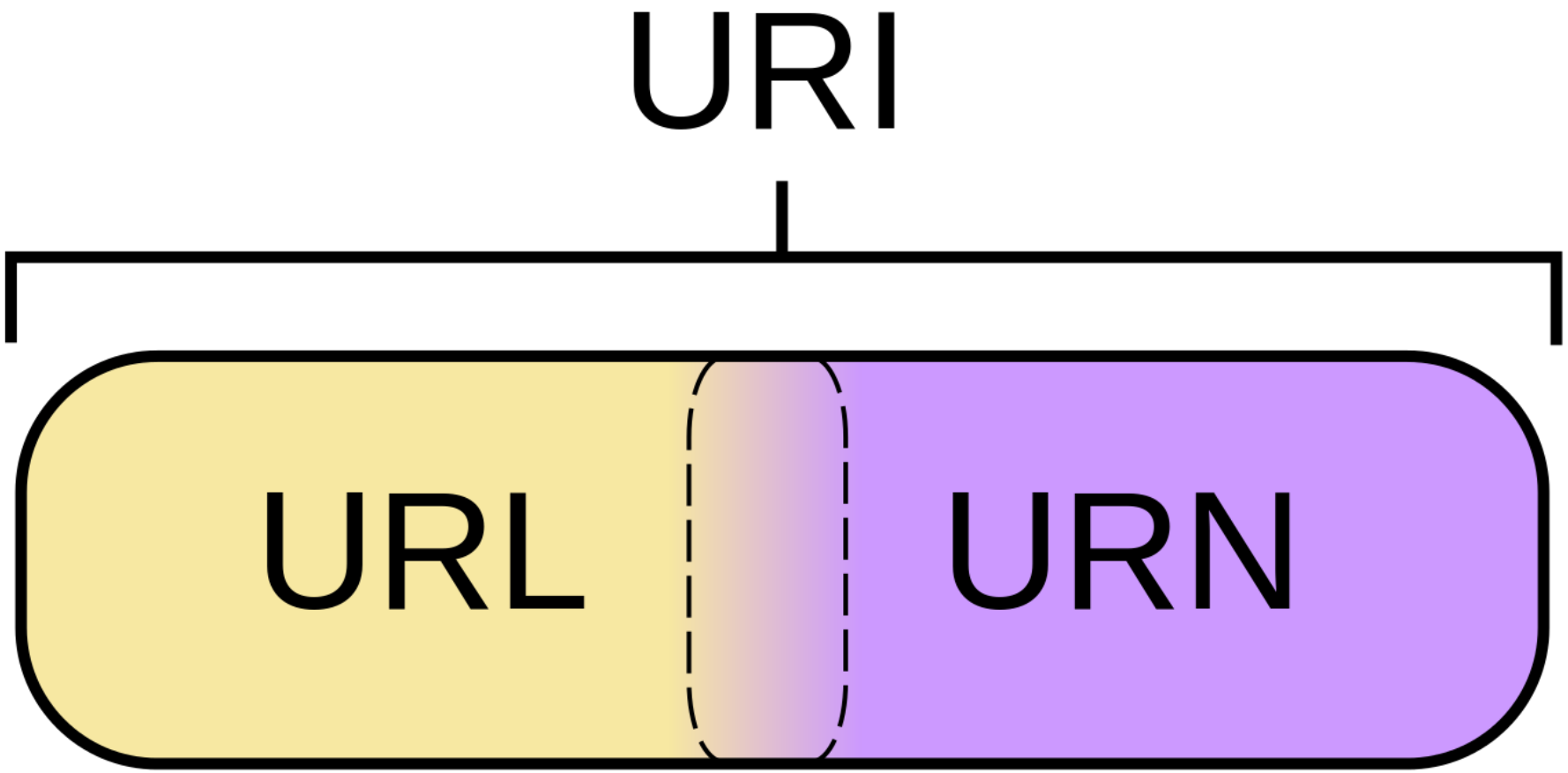
URL

- Most of these parts are null most of the time
- The following are all valid URLs

```
https://example.com
/path/to/something.html
mailto:fischerm@email.arizona.edu
foo
//ajax.googleapis.com/libs/jquery.min.js
../somepage.php?key=123
anotherpage.html#figure1
#droids
```

URI, URL, URN

- URI - Universal Resource Identifier
- URL - Universal Resource Locator
- URN - Universal Resource Name
- These are NOT interchangeable. Each has a different meaning, although there can be significant overlap
- We’re almost always going to use URLs unless otherwise explicitly mentioned



http://en.wikipedia.org/wiki/File:URI_Euler_Diagram_no_lone_URIs.svg

URI

```
The generic URI syntax consists of a hierarchical sequence of
components referred to as the scheme, authority, path, query, and
fragment.

   URI         = scheme ":" hier-part [ "?" query ] [ "#" fragment ]

   hier-part   = "//" authority path-abempty
               / path-absolute
               / path-rootless
               / path-empty
```

http://tools.ietf.org/html/rfc3305

http://tools.ietf.org/html/std66

Links

- Let's assume we're currently on the following page:
  - http://www.cs.arizona.edu/classes/cs337/spring16/
- There are two links on this page that are functionally equivalent:
  - href="http://www.cs.arizona.edu/classes/cs337/spring16/syllabus.html"
  - href="syllabus.html"
- The first one is an *"absolute"* URL and the second is a *"relative"* URL. Clicking on either of them from that page produces the same result.
- Is one preferable to the other? What circumstances would make you choose one way over another?

Anchor Fragments

http://sia.uits.arizona.edu/eds_attributes#eduPersonAffiliation

- The last part of a URL, following a # symbol targets a fragment within a page
- Can target an element with a matching `id` attribute

```
<a id="eduPersonAffiliation">…</a>
```

URL Schemes

- The Scheme tells the browser how to access the resource.
- `file:///` loads the file directly from the local filesystem
- `http://` initiates an HTTP connection over TPC/IP
- `https://` establishes a secure connection over SSL, then communicates via HTTP
- `email:` hands off control to an email client
- `tel:` hands off control to a phone client

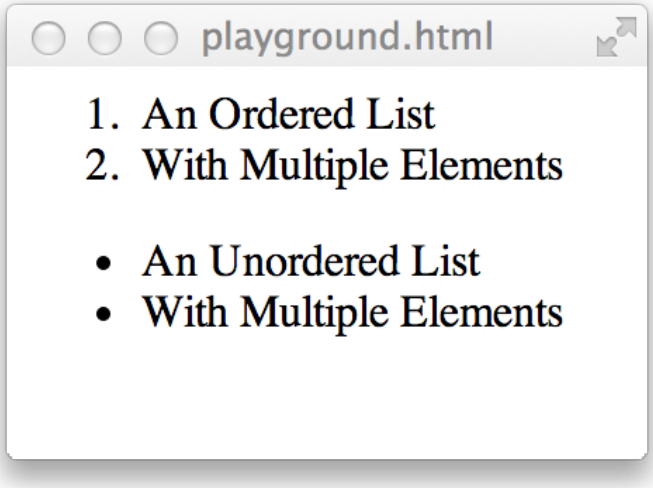
More Elements

Ordered and Unordered Lists

- `<ol>` Ordered List
- `<ul>` Unordered List
- `<li>` List Element - Used for both types of lists
- Closing Tag for `<li>` may be omitted

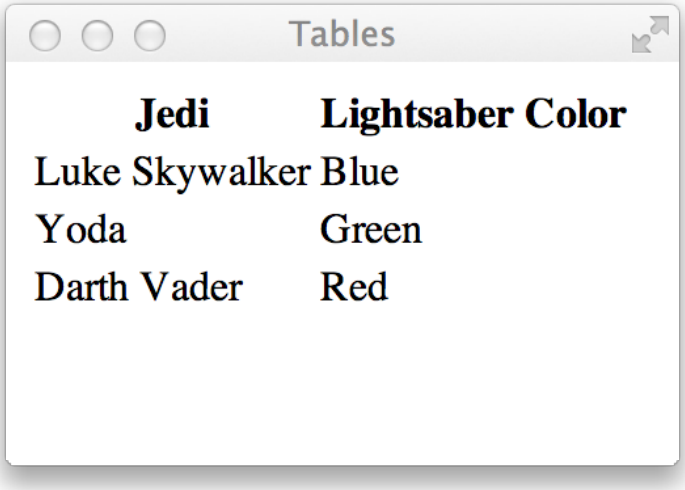
```
<ol>
  <li>An Ordered List</li>
  <li>With Multiple Elements</li>
</ol>

<ul>
  <li>An Unordered List
  <li>With Multiple Elements
</ul>
```



Tables

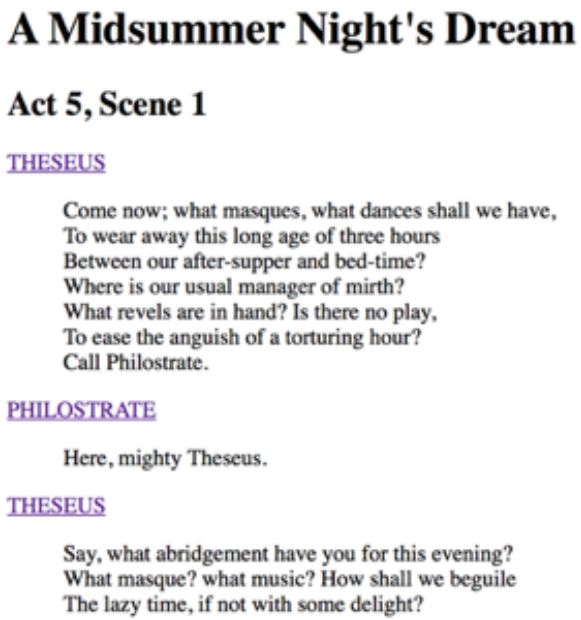
- `<table>` begins a table
- `<th>` table header
- `<tr>` table row
- `<td>` table data




```
<table>
  <tr>
    <th>Jedi</th>
    <th>Lightsaber Color</th>
  </tr>
  <tr>
    <td>Luke Skywalker</td>
    <td>Blue</td>
  </tr>
  <tr>
    <td>Yoda</td>
    <td>Green</td>
  </tr>
  <tr>
    <td>Darth Vader</td>
    <td>Red</td>
  </tr>
</table>
```

Headings

- `<h1>` 1st level heading
  - Biggest
- `<h6>` 6th level heading
  - Smallest
- `<h1> <h2> <h3> <h4> <h5> <h6>`



A Midsummer Night's Dream

Act 5, Scene 1

THESEUS

Come now; what masques, what dances shall we have,
To wear away this long age of three hours
Between our after-supper and bed-time?
Where is our usual manager of mirth?
What revels are in hand? Is there no play,
To ease the anguish of a torturing hour?
Call Philostrate.

PHILOSTRATE

Here, mighty Theseus.

THESEUS

Say, what abridgement have you for this evening?
What masque? what music? How shall we beguile
The lazy time, if not with some delight?

Generic Block and Inline Elements

- div is a generic block element. It takes up all available horizontal space.
- span is the generic inline element. It takes up only as much horizontal space as its content.

Generic Block and Inline Elements

```
<div>
  <div>
    Pony ipsum dolor sit amet fluttershy
    hoof eu Daring Do Sweetie Drops.
  </div>
  <div>
    Cloud <span>officia Owlowiscious pie</span>
    muffin Angel horn enim gryphon honesty.
  </div>
  <div>
    <img src="pony.png">
    <span>Aliqua pegasus alicorn eu.</span>
  </div>
</div>
```

- Without styling it's very difficult to discern the document structure with just div and span elements.

Generic Block and Inline Elements

```
<div class="blog-post">
  <div class="title">
    Pony ipsum dolor sit amet fluttershy
    hoof eu Daring Do Sweetie Drops.
  </div>
  <div class="content">
    Cloud <span>officia Owlowiscious pie</span>
    muffin Angel horn enim gryphon honesty.
  </div>
  <div class="picture">
    <img src="pony.png">
    <span class="caption">Aliqua pegasus alicorn eu.</span>
  </div>
</div>
```

- CSS classes are required to give the document some structure.

Semantic Tags

- HTML5 Placed a new emphasis on semantics
- The meaning of a document should be understandable from the DOM, and not need visual styles to understand.

Semantic Tags

```
<article>
  <header>
    Pony ipsum dolor sit amet fluttershy
    hoof eu Daring Do Sweetie Drops.
  </header>
  <details>
    Cloud <em>officia Owlowiscious pie</em>
    muffin Angel horn enim gryphon honesty.
  </details>
  <figure>
    <img src="pony.png">
    <figcaption>Aliqua pegasus
    alicorn eu.</figcaption>
  </figure>
</article>
```

Semantic Tags

- The `<article>` element represents a complete, or self-contained, composition in a document. This could be a forum post, a magazine or newspaper article, a blog entry, etc.
- The `<header>` element represents introductory content for its nearest ancestor.
- The `<figure>` element represents some flow content, optionally with a caption, used to annotate illustrations, diagrams, photos, code listings, etc.

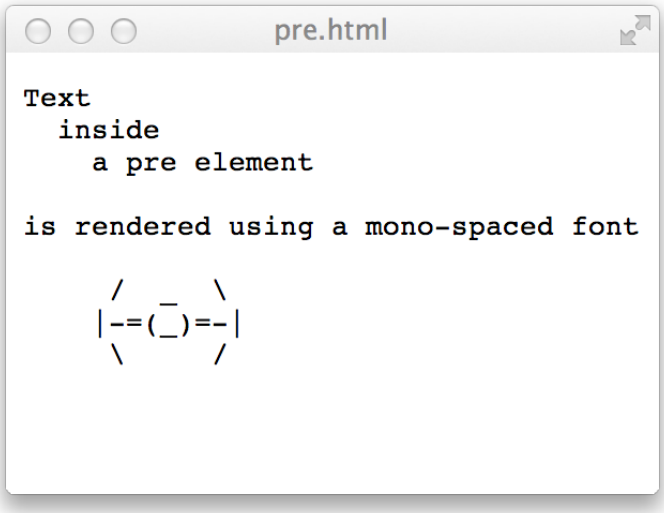
Styling with Tags

- The `<pre>` tag represents a block of preformatted text, in which structure is represented by typographic conventions rather than by elements.
- Whitespace and newlines are rendered as-is

```
<pre>
Text
  inside
    a pre element

is rendered using a mono-spaced font

     /   _   \
    |-=(_)=-|
     \       /
</pre>
```



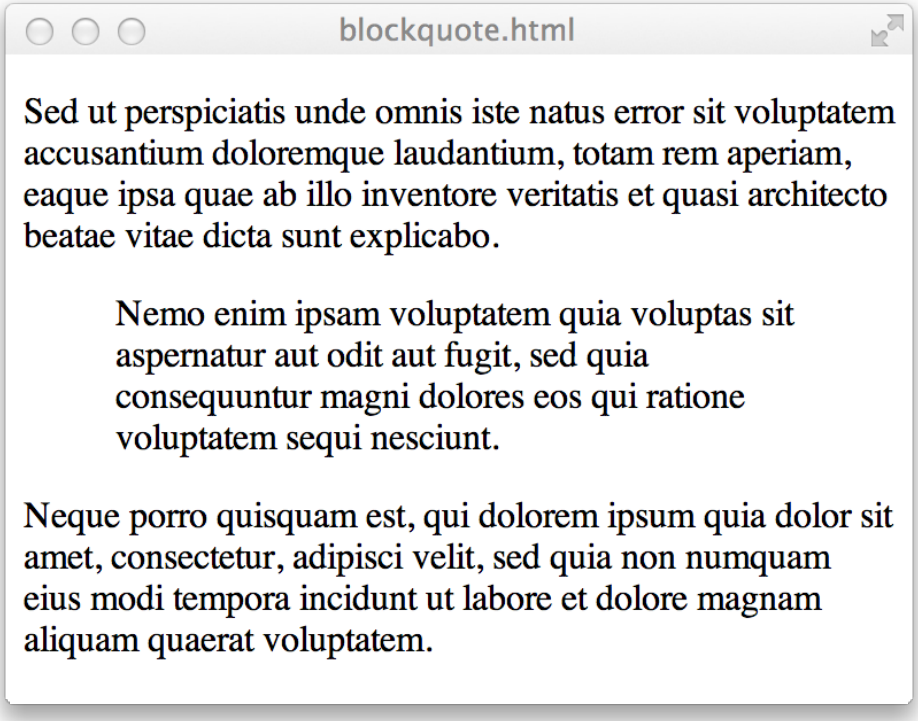
Styling with Tags

- The `<blockquote>` renders as indented text

```
<p>
Sed ut perspiciatis unde omnis iste natus error sit
voluptatem accusantium doloremque laudantium, totam
rem aperiam, eaque ipsa quae ab illo inventore
veritatis et quasi architecto beatae vitae dicta sunt
explicabo.
</p>

<blockquote>
Nemo enim ipsam voluptatem quia voluptas sit
aspernatur aut odit aut fugit, sed quia consequuntur
magni dolores eos qui ratione voluptatem sequi
nesciunt.
</blockquote>

<p>
Neque porro quisquam est, qui dolorem ipsum quia dolor
sit amet, consectetur, adipisci velit, sed quia non
numquam eius modi tempora incidunt ut labore et dolore
magnam aliquam quaerat voluptatem.
</p>
```



Styling with Tags

- Both pre and blockquote are pretty old elements.
- Defined in HTML 1.0
- Are these pure structure elements?
- Are the header tags (`<h1>` … `<h6>`) similar?
- What other tags typically define both structure and style?

http://www.w3.org/MarkUp/draft-ietf-iiir-html-01.txt

Images

Images

```
<figure>
  <img src="img/droid.jpg" alt="A Droid">
  <figcaption>
    https://www.flickr.com/photos/dunechaser/6987810377
  </figcaption>
</figure>
```

- Something other than text!
- The img tag is a void element, so it has no closing tag
- By default images are displayed at their native pixel size

Images

- Images can be resized with CSS, or with `width` and `height` attributes.
- Resized images are not resampled. The full image is sent to the browser no matter what size the image is ultimately displayed at.
- Assigning just `width` or `height` will scale the image and preserve the aspect ratio. (`width:height`)
- Examples!

Images

- The `alt` attribute should always be present, and should describe the image as best you can.
- Accessibility should be thought about from the very start of an HTML project, and not at the very end.
- If an image provides no useful information (a spacer image, or background gradient) an empty `alt` attribute should be used: `alt=""`

Images

- Three widely supported Image formats
  - GIF - Graphics Interchange Format
  - JPEG - Joint Photographic Experts Group
  - PNG - Portable Network Graphics
- HTML Specification does not mandate support for any particular format

GIF

- 256 distinct colors. Each GIF can have its own color pallet.
- One color can be designated as transparent.
- Can contain multiple frames for animation.
- Lossless compression, but limited format.

JPEG

- Millions of colors
- Lossy compression
  - Higher quality, less compression, larger file size
  - Smaller file size, higher compression, less quality
- Designed to be good at compressing photographs.
- No transparency

PNG

- Lossless compression
- No animation
- Several bit depth variants
  - PNG-8: 256 colors
  - PNG-24: 16 Million colors (3 8-bit channels)
  - PNG-32: 16 Million colors + 8-bit transparency
    - Allows for smooth anti-aliased transparency

Images

	GIF	JPEG	PNG
Photograph		✔	
Animated	✔		
Icon or Drawing	✔		✔
Transparency	✔		✔

Images

- You can spend a lot of time working on images
- Good discussion of images in *Head First HTML and CSS*

Character References

- The `&` and `<` characters are special, and are allowed in some places but not others
- `<p title="<">` … ok
- `<p>a<b</p>` … problem
- `<p>a&b</p>` … problem
- Solution: Use `&` and `<` instead
  - `<p>a<b</p>`

Character References

- Used to escape certain characters
- Three different styles
  - & name ; `©` ©
  - &# decimal-digits ; `★` ★
  - &#x hex-digits ; `∞` ∞
- Mostly we'll use `&` `<` `>` `"` `'` ` `

http://www.w3.org/TR/html5/syntax.html#named-character-references

Document Object Model

DOM Tree

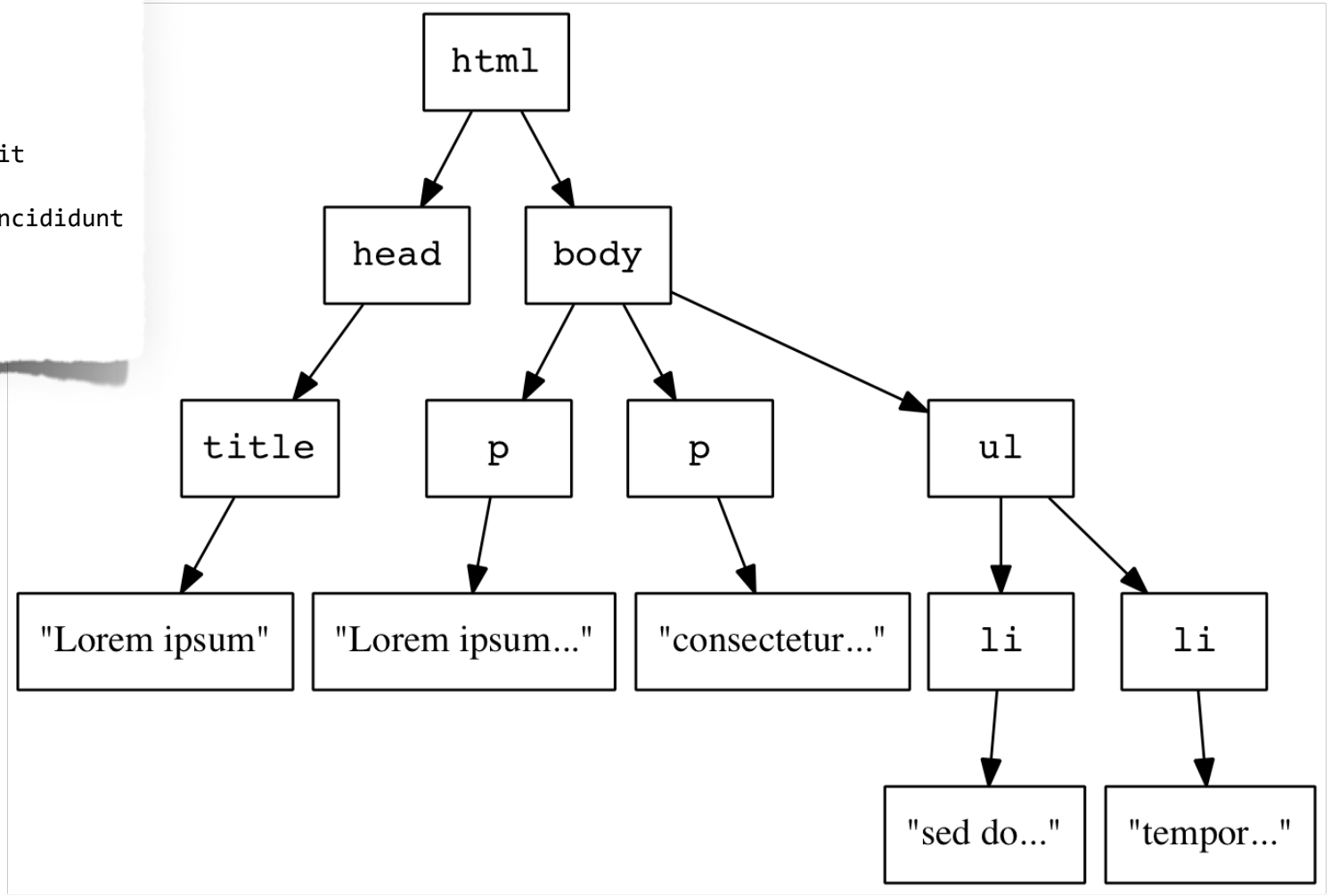
```
<!doctype html>
<html>
<head>
  <title>Lorem Ipsum</title>
</head>

<body>
  <p>
   Lorem ipsum dolor sit amet
  </p>
  <p>consectetur adipisicing elit
  <ul>
    <li>sed do eiusmod tempor incididunt
    <li>tempor incididunt
  </ul>
</body>
</html>
```

DOM Tree

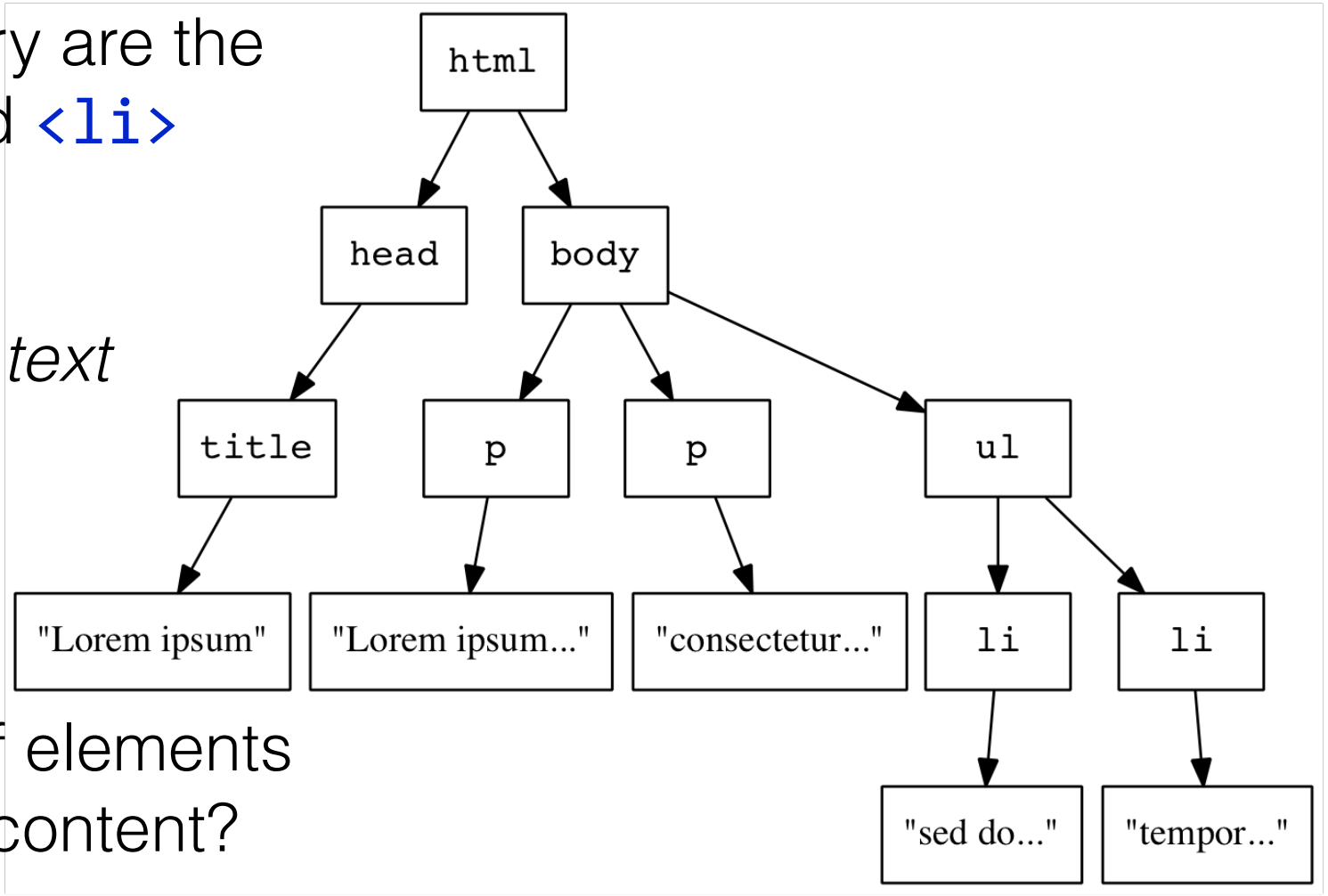
```
<!doctype html>
<head>
  <title>Lorem Ipsum</title>
</head>

<body>
  <p>
   Lorem ipsum dolor sit amet
  </p>
  <p>consectetur adipisicing elit
  <ul>
    <li>sed do eiusmod tempor incididunt
    <li>tempor incididunt
  </ul>
</body>
</html>
```



DOM Tree

- What category are the `<p>` `<ul>` and `<li>` elements?
- What are the *text* elements?
- Would *all* leaf elements be phrasing content?



Misc Details

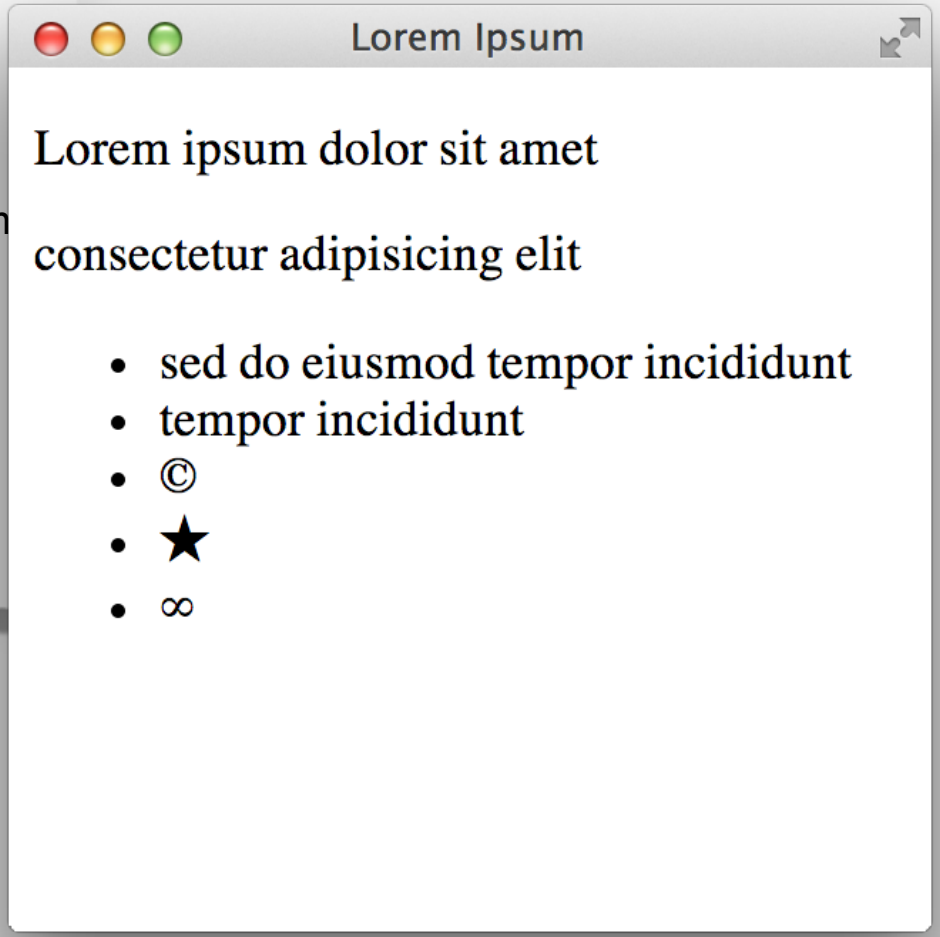
- HTML Tags and attribute names are **not case sensitive**
- Comments: `<!-- ••• -->`
  - Cannot nest comments. No inline comments
- Whitespace is mostly ignored. Multiple whitespace characters are condensed to a single space when rendered
- Text nodes and attribute values must be a tab, newline, form-feed, carriage-return or unicode characters ≥ than U+0020 (space)

Rendering

That Thing Browsers Do

```
<!doctype html>
<html>
<head>
  <title>Lorem Ipsum</title>
</head>

<body>
  <p>
   Lorem ipsum dolor sit amet
  </p>
  <p>consectetur adipisicing elit
  <ul>
    <li>sed do eiusmod tempor incididun
    <li>tempor incididunt
    <li>&copy;
    <li>&#9733;
    <li>&#x221E;
  </ul>
</body>
</html>
```



A quick aside… Lorem Ipsum

- Nonsense text to fill up space. Its latin-ish…
- Great to block out sections of a design.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

http://www.lipsum.com

Lorem Ipsum

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation

Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident

Bob Ross Ipsum

Little trees and bushes grow however makes them happy. I sincerely wish for you every possible joy life could bring. If you don't think every day is a good day - try missing a few. You'll see. And that's when it becomes fun - you don't have to spend your time thinking about what's happening - you just let it happen. Of course he's a happy little stone, cause we don't have any other kind.

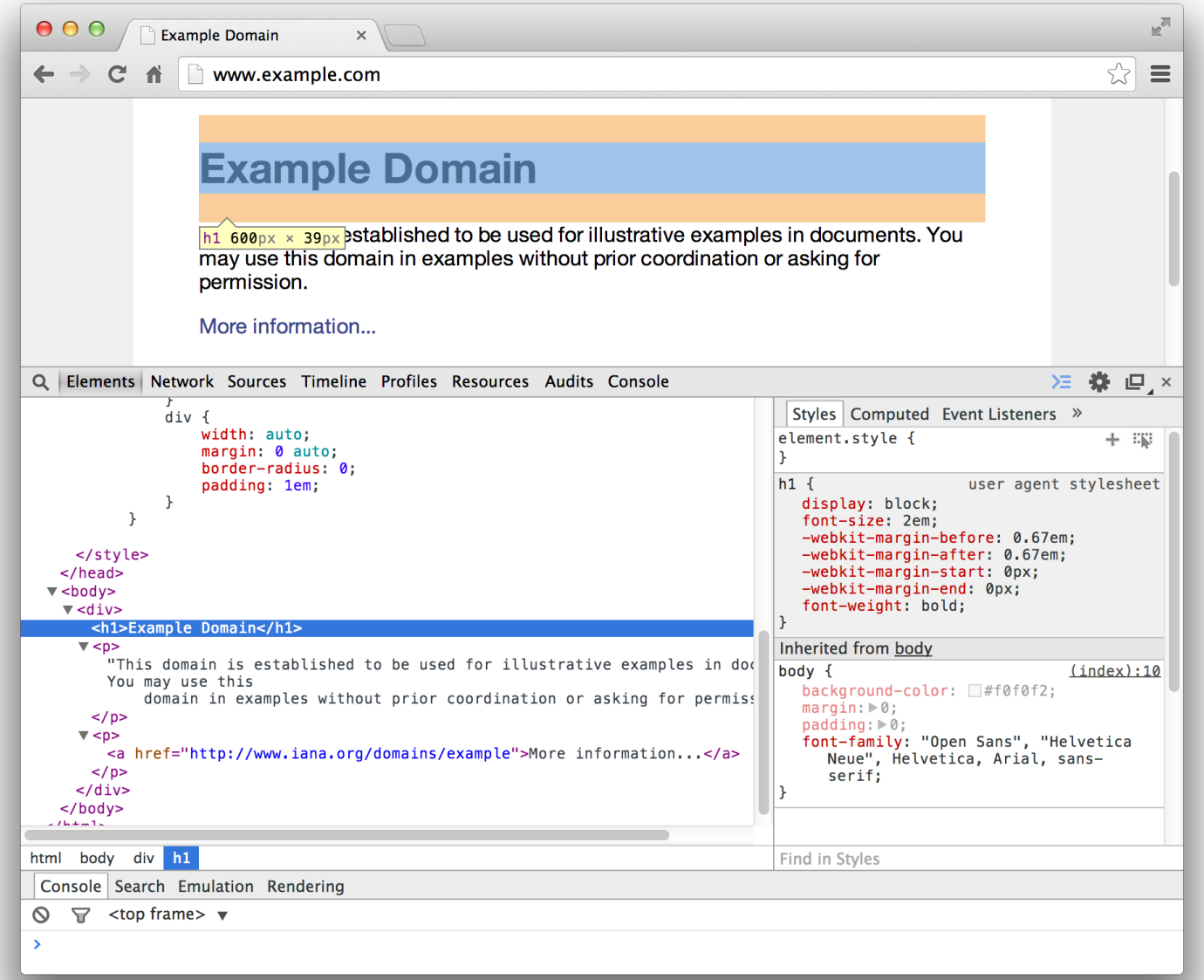
http://www.bobrosslipsum.com

- Riker Ipsum - http://www.rikeripsum.com
- Hodor Ipsum
- Samuel L Ipsum (nsfw)

.. back to rendering

- A ***rendering engine*** takes HTML and turns it into something you can see (or hear)
- Gecko - Firefox 
- KHTML - Konqueror 
- Trident - Internet Explorer 
- WebKit (based on KHTML) - Safari 
- Blink (based on WebKit) - Chrome 

Chrome Developer Tools



Validation and Errors

Validation

```
<!doctype html>
<html lang="en">
  <head>
    <title>Validator!</title>
  </head>

  <body>
    <p>
      Let’s Test it
    </p>
  </body>
</html>
```

http://validator.w3.org/#validate_by_input

Errors

- Check errors in validator
- Check for errors in browser developer tools.
- Examples

- Why are browsers so permissive in handling errors?
- What risks are posed by being so forgiving?
- Do you think browsers are as forgiving with CSS and JavaScript as they are with HTML?
- What other types of software is as permissive?

HTML Review

- `<!doctype html>` starts a document. NOT an element
- Structure of an HTML element
- Attributes
- Open / Closing tags. Optional Tags
- “Block/Inline” elements
- Flow / Phrasing elements

- `<html> <head> <body>`
- Nested Elements
- Links / Anchors / Fragments / Targets
- URLs
- Lists
- Tables
- Headings

- `<pre>`
- `<blockquote>`
- `<img>`
  - GIF, JPEG, PNG, Transparency
  - Alt attribute
- Character References
  - `& < > " '  `

- DOM Tree
- Comments
- Whitespace
- Validation