ISTA 230

HTML
A Look Under the Hood

HTML

Plain Text

> Where the Sidewalk Ends Shel Silverstein There is a place where the sidewalk ends And before the street begins, And there the grass grows soft and white, And there the sun burns crimson bright, And there the moon-bird rests from his flight To cool in the peppermint wind.

Without any formating or structure, the above paragraph is just that, a paragraph. While a discerning user may recognize the paragraph as a poem by Shel Silverstein, it would be difficult for a computer to do the same.

HTML

Plain Text - Formatted

Where the Sidewalk Ends
Shel Silverstein
There is a place where the sidewalk ends
And before the street begins,
And there the grass grows soft and white,
And there the sun burns crimson bright,
And there the moon-bird rests from his flight
To cool in the peppermint wind.

Were we to break the poem into lines, it becomes somewhat easier to tell that the text is a poem. We could even teach a computer that the first line is the title and the second line is the author. However, this isn't a scalable model as some poems may have titles that span two lines, multiple authors, etc.

HTML

Markup

```
<poem>
    <title>Where the Sidewalk Ends</title>
    <author>Shel Silverstein</author>
    <stanza>
        There is a place where the sidewalk ends
        And before the street begins,
        And there the grass grows soft and white,
        And there the sun burns crimson bright,
        And there the moon-bird rests from his flight
        To cool in the peppermint wind.
    </stanza>
</poem>
```

Markup allows one to provide additional information about a given set of content. In the above example, the markup allows us to clearly indicate what the text is (i.e., a poem), what the title of the poem is, who the author of the poem is, and what lines are related (i.e., part of the same stanza).

Note: The above is not HTML. This is just an example of markup as a concept.

HTML

Markup Languages

- Common Characteristics:
    - Provides additional semantic meaning
      As shown in the previous example, markup languages can provide additional information about a given set of content, making the content more understandable and meaningful.

    - Well-defined set of available elements
      Unlike the previous example, most markup languages have a specific set of what markup elements you can use. Using any undefined elements would result in an invalid document.

    - Document Type Definition (DTD) or schema
      The details of what goes into a DTD would take a course of its own. It is sufficient for this course to know that the DTD provides a list of rules for a given markup language about what elements can be used, how they can be used, and what attributes a given element might have.

HTML

Markup Languages

- Examples of markup languages
    - LaTeX
      LaTeX is a markup language for the TeX typesetting program. It is most-commonly used in academic circles for writing papers and/or books.

    - XML
      The goal of the eXtensible Markup Language (XML) can be used to make documents readable by both machines and computers. As the name implies, XML is very extensible and is used in a number of applications, including Microsoft Office and Apple iWork. There are also a number of markup languages based on XML.

    - HTML

HTML

Hypertext Markup Language (HTML)

- Proposed by Tim Berners-Lee in 1991
  The original proposal can still be viewed by visiting http://lists.w3.org/Archives/Public/www-talk/1991SepOct/0003.html.

- Specified 21 markup elements
- **Used to provide structure, organization, and meaning**

HTML

HTML Specifications

The Internet Engineering Task Force (IETF) is an open community of IT professionals and enthusiasts concerned with the operation and evolution of the Internet. While Tim Berner's Lee first proposed HTML in 1991, it wasn't formally defined and recognized by the IETF until 1993. That said, browser developers were implementing HTML support before it was officially recognized.

- HTML - June 1993
  The original HTML proposal contained 21 markup elements. However, by the time it was formalized, it had grown to 41.
- HTML+ - November 1993
  Added forms, tables, inline images (already implemented in the Mosaic web browser) and a few other useful tags.

HTML

Browser Influence

- Most Popular Browsers
    - Mosaic - Launched 1993
    - Netscape Navigator - Launched 1994
- Invented their own additional HTML tags!

In an effort to advance the evolution of HTML, browser makers would sometimes implement their own tags as a solution to a given problem. For example, Marc Andreessen, the co-author of Mosaic, pushed for the <img> tag for inline images. While others objected to this tag on the grounds that it wasn't general enough, Mosaic was already using it and it has become firmly ingrained in HTML specifications since.

While some would argue that this push for new tags was necessary for the development of HTML, the fear was that without some sort of oversight and canonical specication, web page designers would become overwhelmed by a barage of HTML tags that only worked on some browsers.

HTML

HTML Specifications

- HTML - June 1993
- HTML+ - November 1993
- HTML 2.0 Proposed - July 1994

Dan Connolly, the author of HTML 2.0 attempted to organize the chaos by pulling together a list of all of the non-standard elements that browser makers had implemented and adding them to the new version of the specification.

HTML

W3C

To help address the issue of loose HTML specifications, Tim Berners-Lee founded the World Wide Web Consortium (W3C).

- World Wide Web Consortium (W3C)
- Founded by Tim Berners-Lee in 1994
- Goal: Standardize web protocols and technologies
- Limitations: Couldn't enforce their recommendations.

While this group was (and still is) successful at defining web protocols and specifications, browser makers weren't necessarily interested in using those standards.

HTML

Browser Wars

- Netscape Navigator - Launched 1994
- Microsoft Internet Explorer - Launched 1995

These two browsers would become the most popular and viewed each other as direct competition. In order to compete, they would continually try to outdo the other by introducing new non-standard HTML elements.

HTML

Bad tags

- < blink >
  < marquee >

In an attempt to one-up the competition, Microsoft and Netscape introduced many non-standard HTML tags, including the two listed above. While viewed as silly by most serious web developers, these tags were abused by novices and people just learning HTML. Additionally, the <blink> tag caused issues for people with epilepsy.

HTML

HTML Specifications

- HTML - June 1993
- HTML+ - November 1993
- HTML 2.0 - November 1995
- HTML 3.2 - January 1997

HTML 3.2 was a combined effort of W3C and browser makers. There was much back and forth, with one browser agreeing to get rid of a non-standard element if another vendor would get rid of one of their non-standard elements.

HTML

HTML 3.2

- W3C working with browser developers
- Incorporated proprietary browser-specific HTML tags...
- Browser developers didn't really stick to it.

HTML

HTML Specifications

- HTML
- HTML+
- HTML 2.0
- HTML 3.2
- HTML 4.0* - December 1997

HTML 4.0 came out with three versions: one for transitional pages (in the process of moving from HTML 3.2 to HTML 4.0) which allowed for deprecated elements, one for strict pages (pure HTML 4.0), and one for framesets, a concept implemented by Netscape and later adopted by other browsers.

HTML

Playing nice with others...

This website is best viewed in

At the height of the browser wars, the tendency of browser makers to implement their own tags and to interpret the use of some standard HTML 4.0 tags had reached the point that web pages would often look completely different depending on the browser the user was viewing the page on.

This led to some developers 'branding' their website with an image that said "Best viewed in _____", specifying what browser they had used to create the site. Others would try to reach a larger audience by developing multiple versions of the same web site, one for each browser they wanted to target.

All in all, the progress of HTML and the Web was quickly turning into a mess!

HTML

Tired of having to worry about how things looked in one browser versus another, a small group of dedicated web developers took up the tasks of convincing browser makers to adhere to standards and educating other developers on standards-compliant web development.

Web Standards Project (WaSP)

- Goals:
  - Educate about standards
  - Work with browser developers to implement standards
  - Ridicule if necessary

While the task seems daunting at best, the group was very successful in their efforts and are responsible for much of the ongoing evolution and progress of web development as a field.

HTML

HTML Specifications

- HTML
- HTML+
- HTML 2.0
- HTML 3.2
- HTML 4.0*
- XHTML 1.0* - January, 2000
- HTML 5 - January, 2008

XHTML 1.0 Strict is what we will be using in this course. It is worth noting that XHTML, in addition to being valid HTML, is also valid XML.

The latest HTML specification, HTML 5, builds off of XHTML 1.0 and will be discussed later in the semester.

HTML

Doctype

Every webpage you create in this course should begin with the following doctype declaration.

HTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
```

HTML

HTML Elements

Example of a standard HTML element
<tagname> Some content here </tagname>

Example of a self-closing HTML element
<tagname />

HTML

HTML Document - HTML Element

After entering the "doctype" for your file, you'll need to insert a start and end "html" tag. Because we're using XHTML 1.0, we'll need to include some attributes in our html start tag. Your document should now look like this.

HTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict...
<html>









</html>
```

In a perfect world, this would be enough. Unfortunately, you have to add more info to the HTML tag for it to be valid XHTML 1.0 Strict.

HTML

HTML Document - HTML Element

Because we're using XHTML 1.0, you'll need to include a xmlns attribute in our html start tag.

HTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict...
<html xmlns="http://www.w3.org/1999/xhtml">









</html>
```

HTML

HTML Document - Title Element

The head section of your web page contains information about the page but does not contain any content that is displayed on the screen. The title element shown below is used to specify the name of the page. This name is displayed at the top of the browser window but does not display on the webpage itself.

HTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict...">
<html xmlns="http://www.w3.org/1999/xhtml">
   <head>
      <title>My Page</title>
   </head>




</html>
```

HTML

HTML Document - Body Element

The body element of the page contains everything that will be displayed as part of the web page.

HTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict...">
<html xmlns="http://www.w3.org/1999/xhtml">
   <head>
      <title>My Page</title>
   </head>
   <body>




   </body>
</html>
```

HTML

HTML Document - Paragraph Element

HTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict...">
<html xmlns="http://www.w3.org/1999/xhtml">
   <head>
      <title>My Page</title>
   </head>
   <body>
      <p>
         This is my content.

      </p>
   </body>
</html>
```

HTML

HTML Document - Line Break Element

HTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict...">
<html xmlns="http://www.w3.org/1999/xhtml">
   <head>
      <title>My Page</title>
   </head>
   <body>
      <p>
         This is my content.<br />
         This is the second line.
      </p>
   </body>
</html>
```

HTML

HTML Document - White Space

It is important to note that any 'white space' - that is, characters such as a space, a tab, or a line break, are treated as a single character when more than one of them occurs in a row. In other words, you can't use multiple spaces between words or elements in the HTML source code to create visual space in the browser.

HTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict...">
<html xmlns="http://www.w3.org/1999/xhtml">
   <head>
      <title>My Page</title>
   </head>
   <body>
      <p>
         This is my content.<br />
         This is the second line.
      </p>
   </body>
</html>
```

Browser

This is my content.
This is the second line.

HTML

HTML Entities

HTML entities allow you to include special characters.

- Used for including special characters
- Format is always an ampersand, then the entity name, then a semi-colon:

  &___;

HTML

HTML Document - HTML Entities

HTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict...">
<html xmlns="http://www.w3.org/1999/xhtml">
   <head>
      <title>My Page</title>
   </head>
   <body>
      <p>
         This is my content.<br />
         This is the second line.<br />
         &copy; 2014
      </p>
   </body>
</html>
```

Browser

This is my content.
This is the second line.
© 2014

HTML

HTML Document - Links

HTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict...">
<html xmlns="http://www.w3.org/1999/xhtml">
   <head>
      <title>My Page</title>
   </head>
   <body>
      <p>
         <a> This is a link </a>
      </p>
   </body>
</html>
```

Browser

This is a link

Only one problem... Where does it link to?

HTML

HTML Attributes

- Attributes provide additional information about a given element
- Attributes are always in the form **attr_name="attr_value"**
- Attribute names are always lowercase

HTML

HTML Attributes

HTML

```
<a href="someURL"> My link text </a>
```

All <a> elements require an 'href' attribute, which is used to specify where the link should go.

HTML

HTML Attributes

HTML

```
<a href="http://www.google.com"> Visit Google </a>
```

This link would take the user to http://www.google.com.

HTML

Absolute vs. Relative URLs

- Absolute URLs
  - For linking to other web sites
  - Always start with 'http' (or some other protocol)
  - Include the domain name

HTML

Absolute vs. Relative URLs

- Relative URLs
    - For linking to web pages on the same web site
    - The link is designed relative to the current web page
    - Do NOT include a protocol
    - Do NOT include the domain name

Absolute URLs should **only** be used for linking to other websites. If you're linking to a page on the same website, you should use relative URLs!

HTML

HTML Document - Links

HTML

```
<a href="http://www.google.com"> Visit Google </a>
```

The link above would direct the user to Google, **a different website than we're already on**, using an absolue URL.

```
<a href="/"> Visit our homepage </a>
```

Because the URL in the 'href' attibute for link above doesn't start with a protocol, we know that the URL is relative. The link above would direct the user to the root of our website. Since no file name was specified, the user would be taken to the file named index.html.

```
<a href="/resources/index.html"> View our resources </a>
```

The link above would direct the user to the resources folder on our website and then display the file index.html.

```
<a href="../files/report.pdf"> View our annual report </a>
```

The link above would direct the user up one folder, and then down into the 'files' folder where it would display the content of the file 'report.pdf'.

HTML

HTML Document - Email Links

```
<a href="mailto:jmspargu@email.arizona.edu"> Email me! </a>
```

Links can be directed to email addresses by including the text 'mailto:' at the beginning of the 'href' attribute, followed by an email address. On most computers, this will result in an email client (such as Microsoft Outlook or Mozilla Thunderbird) opening, ready to compose a message.

HTML

HTML Document - Fragment Identifiers in Links

```
<a href="somepage.html#middle"> Jump to the middle of some page! </a>
```

So what exactly do fragment identifiers link to?

HTML

ID attribute

- Can be specified for all HTML elements
- ID value must start with a letter and be made up of the following:
    - letters
    - numbers
    - underscores
    - dashes

HTML

ID attribute

- Useful for providing additional semantic information
- An 'id' value can only be used once per web page!

It is important to note that an 'id' can only be used once per page but can be used on different pages without any problem. For example, one could have multiple HTML files, each with an <h1> element with an 'id' attribute of 'logo'. However, one could **NOT** have multiple <h1> elements in a single HTML file with the same 'id' attribute of 'logo'.

HTML

ID attribute

- This is invalid!

HTML

```
<p id="user">John</p>
<p id="user">Sue</p>
<p id="user">Bill</p>
```

HTML

ID attribute

- This is valid!

HTML

```
<p id="user1">John</p>
<p id="user2">Sue</p>
<p id="user3">Bill</p>
```

HTML

HTML Links - Fragment Identifiers

HTML

```
<body>

   <p>A little content here...</p>

   ...

   <p id="specialAnnouncement">A special announcment here!</p>

</body>
```

Allows us to create URLs that link directly to specific parts of a page:
http://www.example.com/filename.html#specialAnnouncement

Using 'id' attributes, you can specify the identifier for a specific element. You can then use this identifier in URLs to tell the browser that it should display that specific element at the top of the page when the user visits that URL.

HTML

HTML Links - Anchors and Fragment Identifiers

We can also use fragment identifiers to move around the same web page:

HTML

```
<body id="top">

   <p>A little content here...</p>

   <p>And a little more content here...</p>

   ...

   <a href="#top">Jump to the top of the page</a>
</body>
```

In the example above, clicking on the link that says "Jump to the top of the page" would tell the browser that it should scroll up the page until the HTML element with an 'id' attribute whose value matches 'top' is displayed.

HTML

HTML - Headings

- Six levels, h1 through h6
- h1 indicates the highest importance
- h6 indicates the least importance

HTML

HTML - Headings

- Don't skip levels!
    - Using <h1>, <h2>, and then <h4> is technically valid. However, it will cost you points in this course.

Heading tags can be used to create hierarchy. This information can be semantically helpful in determining how your web page is structured and what the relationship is between certain elements.

HTML

HTML - Headings

HTML

```
<body>
   <h1>Newspaper Name</h1>

      <h2>Section Name</h2>
      <p> Welcome to the Business section!</p>

         <h3>Article Name</h3>
         <p>
            This is the content of my article...
         </p>
</body>
```

In the above example, the <h1> element designates the name of the newspaper, the <h2> element is the section of the newspaper, and the <h3> is the title of the article. Using the appropriate heading levels, one can easily determine the hierarchy of the various elements.

HTML

HTML - Comments

HTML

```
<!-- This is an HTML comment. "I don't show up!" -->
```

HTML

```
<!--
   This is another HTML comment.
   "I can span multiple lines."
-->
```

HTML comments are useful for documenting how your website is structured. They are not displayed to users but are useful for other web designers as well as for yourself when you return to editing a page.

HTML

HTML - Comments

HTML

```
<body>
   <!-- Start newspaper content. -->
   <h1>Newspaper Name</h1>

      <h2>Section Name</h2>
      <p> Welcome to the Business section!</p>

         <h3>Article Name</h3>
         <p>
            This is the content of my article...
         </p>

   <!-- End newspaper content. -->
</body>
```

HTML comments can be used to specify where sections of your website start and stop. They can also be used to explain any HTML that might be confusing otherwise.

HTML

HTML - Comments (a word of caution)

HTML

```
<body>
   <!-- Start newspaper content. --  >
   <h1>Newspaper Name</h1>

      <h2>Section Name</h2>
      <p> Welcome to the Business section!</p>

         <h3>Article Name</h3>
         <p>
            This is the content of my article...
         </p>

   <!-- End newspaper content. -->
</body>
```

Make sure you close your comments correctly!

If you don't, you may make your entire website into one large comment (as is the case in the example above).

HTML

HTML Formatting

- What is purpose of HTML?

To provide structure, organization, and meaning!

HTML

HTML Formatting

The old (wrong) way...

HTML

```
<i> This is some italicized text!</i>
```

In previous versions of HTML, one could use the <i> element to italicize text. While this was functional, it clearly was an HTML element that was used for display purposes, not to provide structure, organization, or meaning.

The new (right) way...

HTML

```
<em> This is some emphasized text!</em>
```

Browser

This is some italicized text!
This is some emphasized text!

While both examples above have the same effect on most browsers, the second is the proper way of using HTML, specifying that the text should be *emphasized*, not italicized.

HTML

HTML Formatting

The old (wrong) way...

HTML

```
<b> This is some bold text!</b>
```

The new (right) way...

HTML

```
<strong> This is some strongly emphasized text!</strong>
```

Browser

This is some bold text!
This is some strongly emphasized text!

Similarly, both the <b> and elements are displayed the same on most modern browsers, the second example is the proper way of using HTML, specifying that the text should be **strongly emphasized**, not bolded.

HTML

HTML Formatting

Why is this important?

Not all users are viewing your page.
Some may be listening.

 may be interpreted as "make italicized" or as "read louder".

 may be interpreted as "make bolded" or as "read even louder".

On the other hand, a screen reader (used by someone with vision impairment) may not know how to interpret 'italized' text.

HTML

Images

HTML

```
<img src="myImage.jpg" alt="About my image" />
```

- Two required attributes:
  - src - location of the image file
  - alt - alternate text for visually-impared users

The 'src' attribute should be used simliarly to the 'href' attribute for links. Specifically, you should use the same rules regarding images from external websites vs. images from your website. While there are rare cases that merit using an image from another website, you'll want to use relative images most of the time.

HTML

Supported Image Types

- JPG
- GIF
- PNG

While there may be other image formats available, they are not necessarily supported by modern browsers and should not be used when one of the three formats above is available instead.

HTML

JPEG - Joint Photographic Experts Group

- File extension: .jpg or .jpeg
- Best used for photographs or photo-like images
- Pixilation on some graphics and text

HTML

GIF - Graphics Interchange Format

- File extension: .gif
- Better for graphic images and text images (see comparison of jpg/gif image below)

HTML

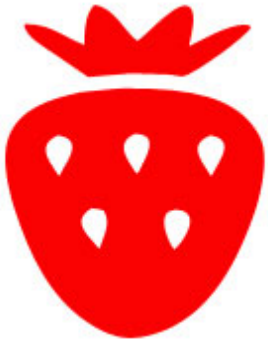
GIF - Graphics Interchange Format

- File extension: .gif
- Better for graphic images and text images
- Provides limited transparency

HTML

GIF - Graphics Interchange Format

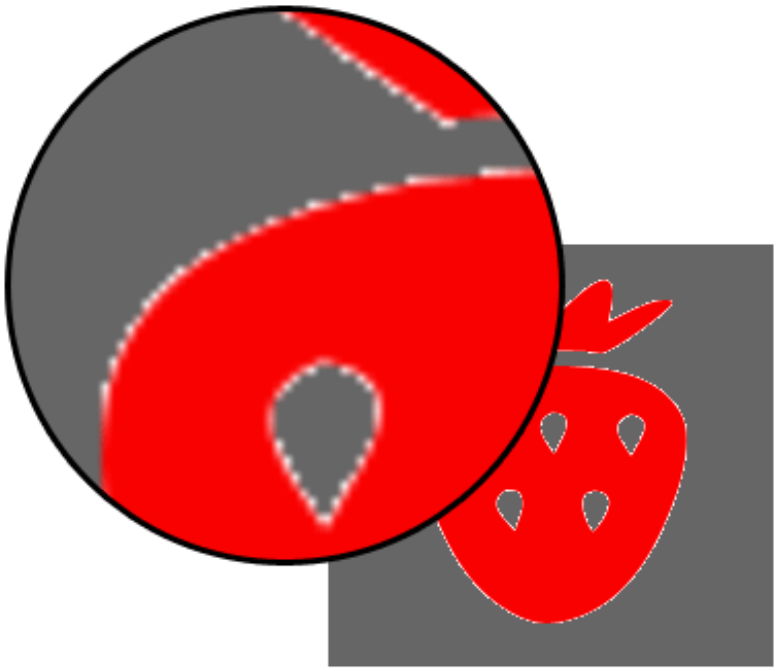
- File extension: .gif
- Better for graphic images and text images
- Provides limited transparency



In the GIF image of the strawberry, we can specify that the white area around the strawberry is transparent, allowing the background color of anything underneath it to show through (see next slide).

HTML

GIF - Graphics Interchange Format



While the transparency will work in some instances, there may be slight pixelation around the edges of the image. For this reason, GIF is often not the preferred format for transparent images.

HTML

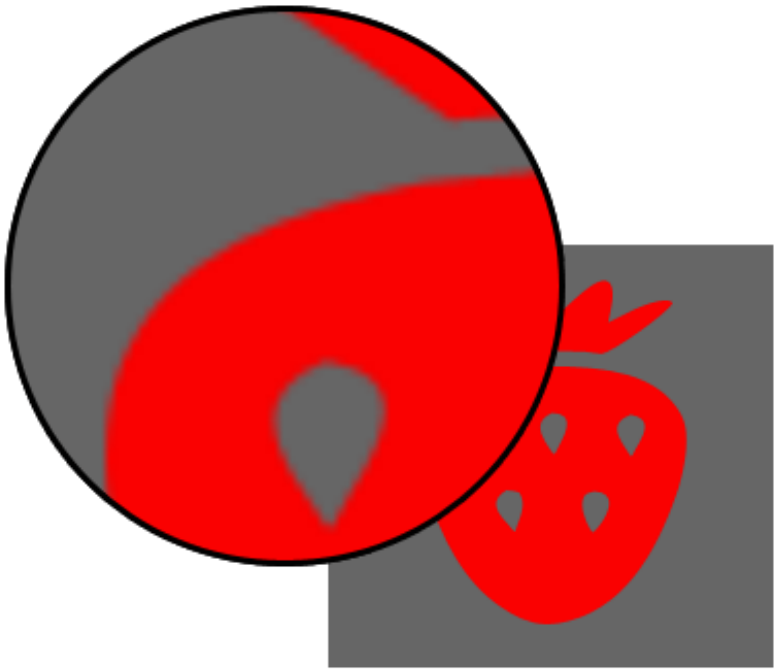
PNG - Portable Network Graphic

- File extension: .png
- Better for graphic images and text images
- Provides smaller file sizes than GIF (typically)
- Displays more quickly on the screen

HTML

PNG - Portable Network Graphic

- Provides better transparency



PNG images don't usually suffer from the same sort of pixilation related to transparency and GIF images.

HTML

PNG - Portable Network Graphic

- Provides alpha transparency
  This allows us to not only specify that certain areas are completely transparent but that other areas are semi-transparent.

HTML

GIF v. PNG

- If PNG is so much better, why do we still use GIF?

GIF images support animation. While the use of GIF animations often can lead to an unprofessional feel, there are instances where they come in handy. For example, many of the 'loading' images used on websites are animated GIF images.

HTML

Where to get images

- Make them - Adobe Photoshop, GIMP
- Google Images - Advanced Search
    - http://www.google.com/imghp?hl=en&tab=wi
    - Use advanced search to specify format
    - When searching for images on Google, it's important to be aware of copyright infringement. Unless otherwise stated, you can't just use someone else's images on your website. In the instance that no copyright statement or statement of use is available, your best bet is to assume that you **do not** have permission to use the image.
- Flickr - Advanced Search
    - http://www.flickr.com/search/advanced/
    - Creative Commons
    - In the advanced search on Flickr, you can specify that you only want to search for images that have a Creative Commons license. Within Creative Commons, you can search for images that are free to use without attribution, that are free to use commercially, and so on.
- Morgue File
    - http://www.morguefile.com/
    - Free to use/edit without attribution
      These are images that have either fallen into public domain or images that have been shared for corporate or public use without any payment or attribution requirements.

HTML

Images

HTML

```
<img src="myImage.jpg" alt="About my image" title="About my image" />
```

The 'title' attribute can also be used to provide a 'mouse-over' effect. When the user moves their mouse over the image, the value of the 'title' attribute will be displayed next to their cursor.

HTML

Horizontal Rules

HTML

```
<hr />
```


Horizontal rules can be used to provide an indication of a break in the content. By default, this displays as a horizontal line across the page.

HTML

Preformatted Text

HTML

```
    <pre> Some content   goes       here....</pre>
```

- Preserves all whitespace and carriage returns in the source markup

HTML

Preformatted Text

HTML

```
<p>
   This    is    some    text.
      This is some indented text.
   This    is    some    other    text.
</p>
```

Browser

This is some text. This is some indented text. This is some other text.

As previously mentioned, multiple spaces, tabs, and line breaks in the HTML source code are treated as a single space when displayed to the user. In the example above, the white space in the <p>, including the line breaks, would not be displayed to the user.

HTML

Preformatted Text

HTML

```
<pre>
   This    is    some    text.
      This is some indented text.
   This    is    some    other    text.
</pre>
```

Browser

```
This    is    some    text.
   This is some indented text.
This    is    some    other    text.
```

When using the <pre> tag, all of the spaces, tabs, and line breaks in the element are displayed to the user. Additionally, the font used is a monospaced font, ensuring that all characters take up exactly the same width.

The <pre> element is one that can easily be abused. It should only be used for elements where it is important for the text to appear exactly as typed. For example, indentation and line breaks are important elements some programming languages. When trying to show a sample of these programming languages, one could use a <pre> element to ensure that the white space is displayed.

The <pre> tag would **NOT** be the optimal choice for displaying content that has lots of line breaks. While the element would allow you to skip the step of entering multiple
 elements, it would also maintain multiple spaces and tabs, which is not typically needed.

HTML

Element types

- Two types of elements:
  - Block elements
  - Inline elements

All HTML elements fall into one of these two categories. Elements cannot be both inline and block. They are either one or the other.

HTML

Block elements

- Have a line break before and after them automatically
- Can contain inline elements
- Can sometimes contain block elements*

* Unless otherwise stated, you can assume that any block elements that we discuss **cannot** contain other block elements.

HTML

Block elements - Some examples

- ...
- <h1> ... </h1>
- <h2>, <h3>, <h4>, <h5>, <h6>

All of the elements listed above are block elements and **cannot** contain other block elements.

HTML

Inline elements

- Do not have a line break before or after
  (**
** is the exception)
- Can contain other inline elements
- **Must be contained by a block element**
- **Cannot contain block elements**

HTML

Inline elements - Some examples

- ...
- ...
- ...
-

HTML

Block elements

HTML

```
<p>
    This is <em>an example of some emphasized text</em>.
</p>
```

Browser

This is *an example of some emphasized text*.

HTML

Inline elements

HTML

```
<p>
   This is <em>an example of some emphasized text</em>.
</p>
```

Browser

This is *an example of some emphasized text*.

HTML

Inline elements

HTML

```
<p>
   This is <em>an example of some emphasized text</em>.
</p>
```

Browser

This is *an example of some italic text*.

HTML

Semantically-neutral HTML elements

The purpose of HTML is to provide:

- Structure
- Organization
- Meaning

Of the elements that we've discussed this far, the one block element that would be used for most content would be a <p> element. That said, not everything is a paragraph. We need something more generic...

HTML

Semantically-neutral HTML elements

- Semantically-neutral (generic) HTML elements
- &
- Two purposes:
    - Provide additional structure/meaning
    - Group related elements

These two elements provide us with elements that can be used when there isn't an HTML element that matches our need.

HTML

- Generic block element
- Can contain block or inline elements

HTML

HTML

```
<!-- Paragraph for holding an image (not really a paragraph). -->
<p>
   <img src="/images/slides/html/logo.png" alt="Company logo" />
</p>
```

```
<!-- div - a better containing block -->
<div>
   <img src="/images/slides/html/logo.png" alt="Company logo" />
</div>
```

The element provides us with a block element that can be used to contain content that doesn't semantically fit inside of another HTML element.

HTML

HTML

```
<h1>Article Title</h1>
<h2>Author</h2>
<p>
   This is the content of my article.
</p>
```

```
<!-- All article-related elements are grouped together -->
<div>
   <h1>Article Title</h1>
   <h2>Author</h2>
   <p>
      This is the content of my article.
   </p>
</div>
```

The element can also be used to group related elements in a single container.

HTML

- Generic inline element
- Can contain inline elements

HTML

HTML

```
<p>
   My favorite book is
   Lord of the Flies.
</p>
```

In the above example, we want to find a way to distinguish the book title from the rest of the text. Unfortunately, there is not a <book> element that we can use...

```
<p>
   My favorite book is
   <span id="title">Lord of the Flies</span>.
</p>
```

We can use the generic element to encapsulate the book title. Using the 'id' attribute, we can provide some additional semantic information about the element as well.

HTML

Guidelines for div/span

- Use only when there is no alternative semantic HTML element.

HTML

```
<!-- There's no need to do this... -->
<div>
   This is a paragraph in my article.
</div>
```

```
<!-- ... when this will suffice. -->
<p>
   This is a paragraph in my article.
</p>
```

HTML

Invalid HTML Markup - Incorrect

HTML

```
<body>

   <a href="myLink.html"> Click here to see another page!</a>


</body>
```

The above example is invalid because the <a> element (an inline element) is not contained by a block element.

HTML

Valid HTML Markup - Correct

HTML

```
<body>
   <p>
      <a href="myLink.html"> Click here to see another page!</a>
   </p>

</body>
```

HTML

Invalid HTML Markup - Incorrect

HTML

```
<body>
   <p>
      <h1>Heading</h1>
      This is my content
   </p>
</body>
```

The above example is invalid because the <h1> element (a block element) is contained by a <p> element. <p> elements are a block element which **cannot** contain other block elements.

HTML

Valid HTML Markup - Correct

HTML

```
<body>
   <h1>Heading</h1>
   <p>
      This is my content
   </p>
</body>
```

HTML

Invalid HTML Markup - Incorrect

HTML

```
<body>
   <p>
      <img src="myimage.jpg" />
   </p>

</body>
```

The above example is invalid because the <img> element has two required attributes: a 'src' attribute and an 'alt' attribute. The 'alt' attribute is missing!

HTML

Valid HTML Markup - Correct

HTML

```
<body>
   <p>
      <img src="myimage.jpg" alt="Brief description of my image" />
   </p>

</body>
```

HTML

Invalid HTML Markup - Incorrect

HTML

```
<body>

   <h1>Fun & Games</h1>


</body>
```

The above example is invalid because the & used in the heading isn't specified as an HTML entity. Because & is a special character, it needs to be specified as '&'.

HTML

Valid HTML Markup - Correct

HTML

```
<body>

   <h1>Fun &amp;amp; Games</h1>


</body>
```

HTML

Invalid HTML Markup - Incorrect

HTML

```
<body>
   <p>
      <img src="myimage.jpg" alt="Brief description of my image">
   </p>

</body>
```

The above example is invalid because the <img> element is a self-closing element. In the case above, the element is not properly closed.

HTML

Valid HTML Markup - Correct

HTML

```
<body>
   <p>
      <img src="myimage.jpg" alt="Brief description of my image" />
   </p>

</body>
```

HTML

Checking Your Markup

- W3C Markup Validation Service
  - http://validator.w3.org/#validate_by_input
  - Scans your markup and reports any errors
  - Error messages can be somewhat obtuse...

HTML

Checking Your Markup

- Some vague errors **(and their possible meanings)**
    - **"Missing one of <*element*>"**
      This usually happens when you have an inline element that isn't contained by a block element.
    - **"end tag for <element> omitted, but OMITTAG NO was specified"**
      This usually happens when you forget to close an element somewhere on your page.
    - **"reference not terminated by REFC delimiter"**
      This usually happens when you use an HTML entity but forget the semicolon at the end.