HTML5 SKELETON PAGE STRUCTURE

```
<!DOCTYPE html>
<html lang=“en”>

  <head>
    <meta charset="utf-8">
    <title>Title Here</title>
  </head>

  <body>
  Page content here
  </body>

</html>
```

MOZILLA DEVELOPMENT NETWORK

Add MDN to any “how to” search and it will bring up the Mozilla Development Network’s answer

MDN MOZILLA DEVELOPER NETWORK

HTML TAGS

Proper HTML 5 standard compliance:

- **All tags and attributes lowercase**
- **All attribute values quoted**
- **All tags properly nested**
- **All tags properly opened and closed**
- **img tag is only empty tags: <img src=" "> or <img src=" " />**
- **DO NOT USE <br> or <br />**

HTML TAG ATTRIBUTES

There are two very important HTML attributes that every tag supports. Both of these attributes are used as hooks to select elements for styling and scripting.

- **ID Attribute**
- **Class Attribute**
- **In HTML 5 the values for these attributes can contain almost any character. I would suggest you start the value with a letter [A-Za-z] and follow it by any number of letters, digits [0-9], hyphens (-), or underscores (_).**
- **This will make your life a lot easier when it comes to using them for CSS or JavaScript hooks.**
- **https://mathiasbynens.be/notes/html5-id-class**

HTML TAG
ID ATTRIBUTE

<div id=“use_me-once"></div>

The “id” attribute on an HTML tag is available for use on any HTML tag. It is a general attribute that works the same for all tags.

- The “id” attribute is used to identify an element on the page
- It is used as a hook to select an element for styling or scripting
- A single value for an “id” attribute can only be used once per page
- A individual HTML element can only have one value for an “id” attribute

HTML TAG
CLASS ATTRIBUTE

<div class=“sharedstyle"></div>

The “class” attribute on an HTML tag is available for use on any HTML tag. It is a general attribute that works the same for all tags.

- **The “class” attribute is used to classify an element on the page**
- **It is used as a hook to select an element for styling or scripting**
- **A single value for a “class” element can be used many time on a page. This is a way to group elements for styling or scripting.**
- **A individual HTML element can have multiple values for an “class” attribute. Multiple values would be space separated inside the quotes.**
    - <div class="tipsbox greentext"></div>

HTML TAGS
INLINE OR BLOCK

HTML Tags are either block elements or inline elements by default. The standard behavior can be changed with CSS.

Block Tags

- **Fill width of window unless floated or positioned**
- **Can contain other blocks (except <p>) and inline elements**
- **https://developer.mozilla.org/en-US/docs/HTML/Block-level_elements**

Inline Tags

- **Flows along with text content**
- **Contains no block level elements, only other inline or text data**
- **https://developer.mozilla.org/en-US/docs/HTML/Inline_elements**

http://www.impressivewebs.com/difference-block-inline-css/

HTML BLOCK ELEMENTS

- If no width is set, will expand naturally to fill its parent container
- Can have margins and/or padding
- If no height is set, will expand naturally to fit its child elements (assuming they are not floated or positioned)
- By default, will be placed below previous elements in the markup (assuming no floats or positioning on surrounding elements)
- Ignores the vertical-align property

https://developer.mozilla.org/en-US/docs/HTML/Block-level_elements

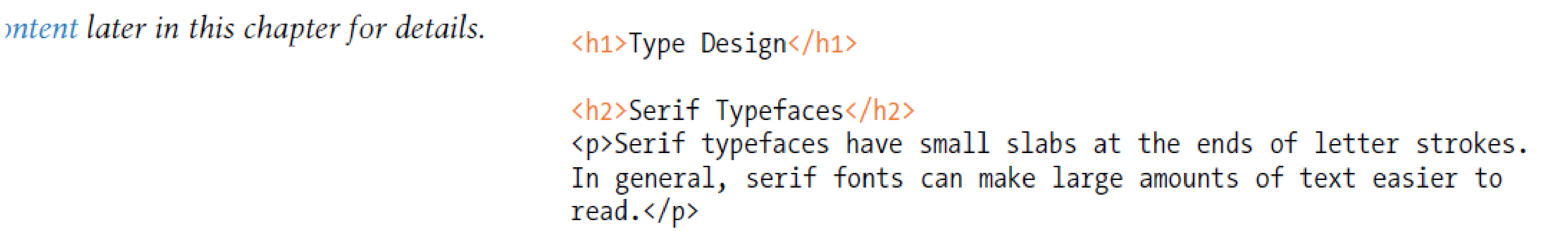
HTML INLINE ELEMENTS

- Flows along with text content, thus
- Will not clear previous content to drop to the next line like block elements
- Is subject to white-space settings in CSS
- Will ignore top and bottom margin settings, but will apply left and right margins, and any padding
- Will ignore the width and height properties
- If floated left or right, will automatically become a block-level element, subject to all block characteristics
- Is subject to the vertical-align property

https://developer.mozilla.org/en-US/docs/HTML/Inline_elements

ITMD-361

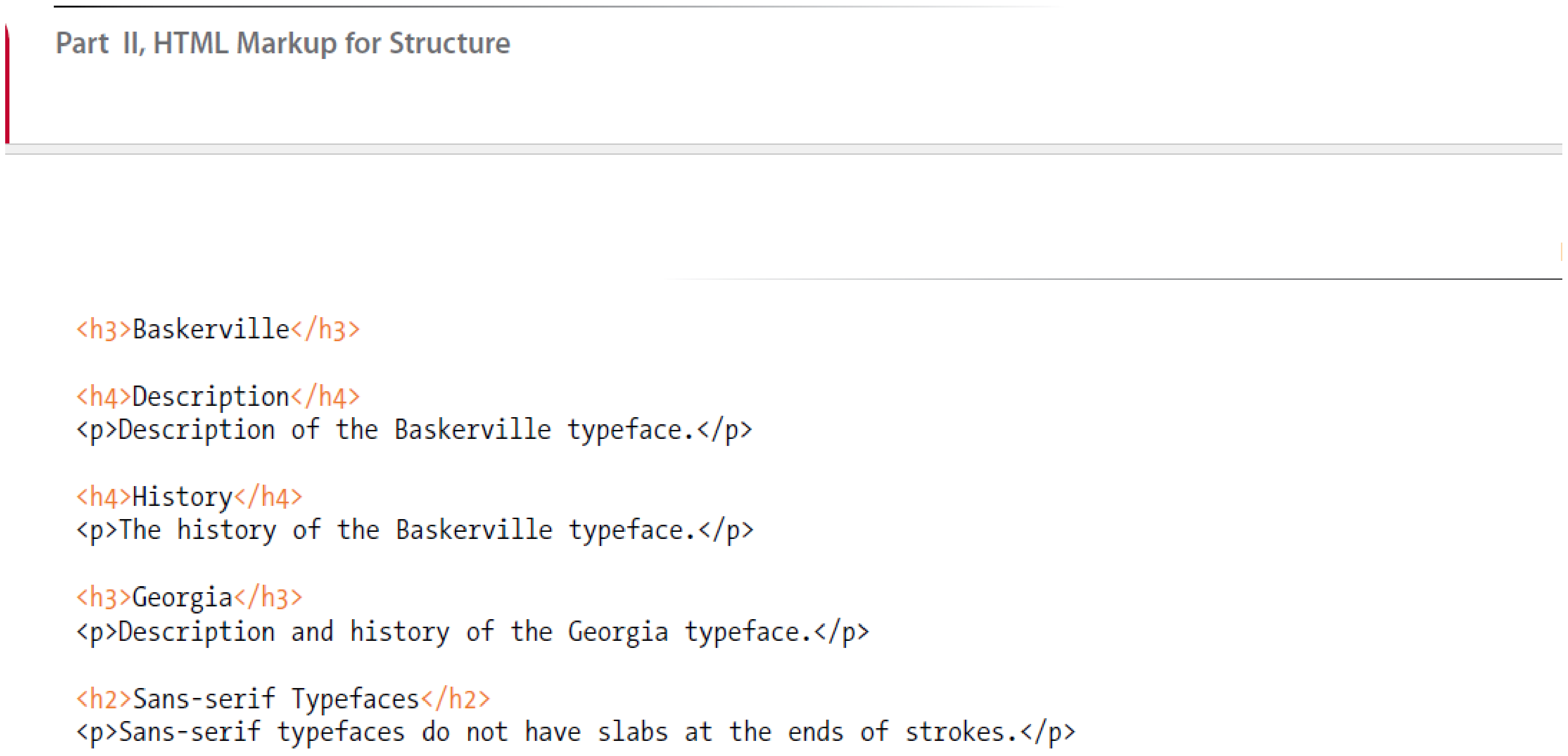
TEXT MARKUP



ntent later in this chapter for details.

```
<h1>Type Design</h1>

<h2>Serif Typefaces</h2>
<p>Serif typefaces have small slabs at the ends of letter strokes.
In general, serif fonts can make large amounts of text easier to
read.</p>
```



Part II, HTML Markup for Structure

```
<h3>Baskerville</h3>

<h4>Description</h4>
<p>Description of the Baskerville typeface.</p>

<h4>History</h4>
<p>The history of the Baskerville typeface.</p>

<h3>Georgia</h3>
<p>Description and history of the Georgia typeface.</p>

<h2>Sans-serif Typefaces</h2>
<p>Sans-serif typefaces do not have slabs at the ends of strokes.</p>
```

The markup in this example would create the following document outline:

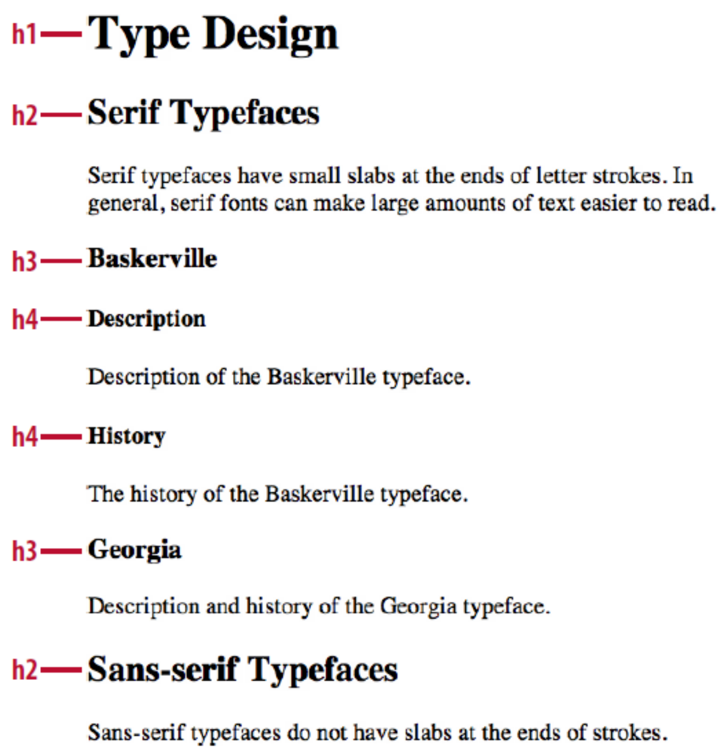
1. Type Design
    1. Serif Typefaces
       + text paragraph
        1. Baskerville
            1. Description
               + text paragraph
            2. History
               + text paragraph
        2. Georgia
           + text paragraph
    2. Sans-Serif Typefaces
       + text paragraph

By default, the headings in our example will be displayed in bold text, starting in very large type for `h1`s, with each consecutive level in smaller text, as shown in Figure 5-1. You can use a style sheet to change their appearance.

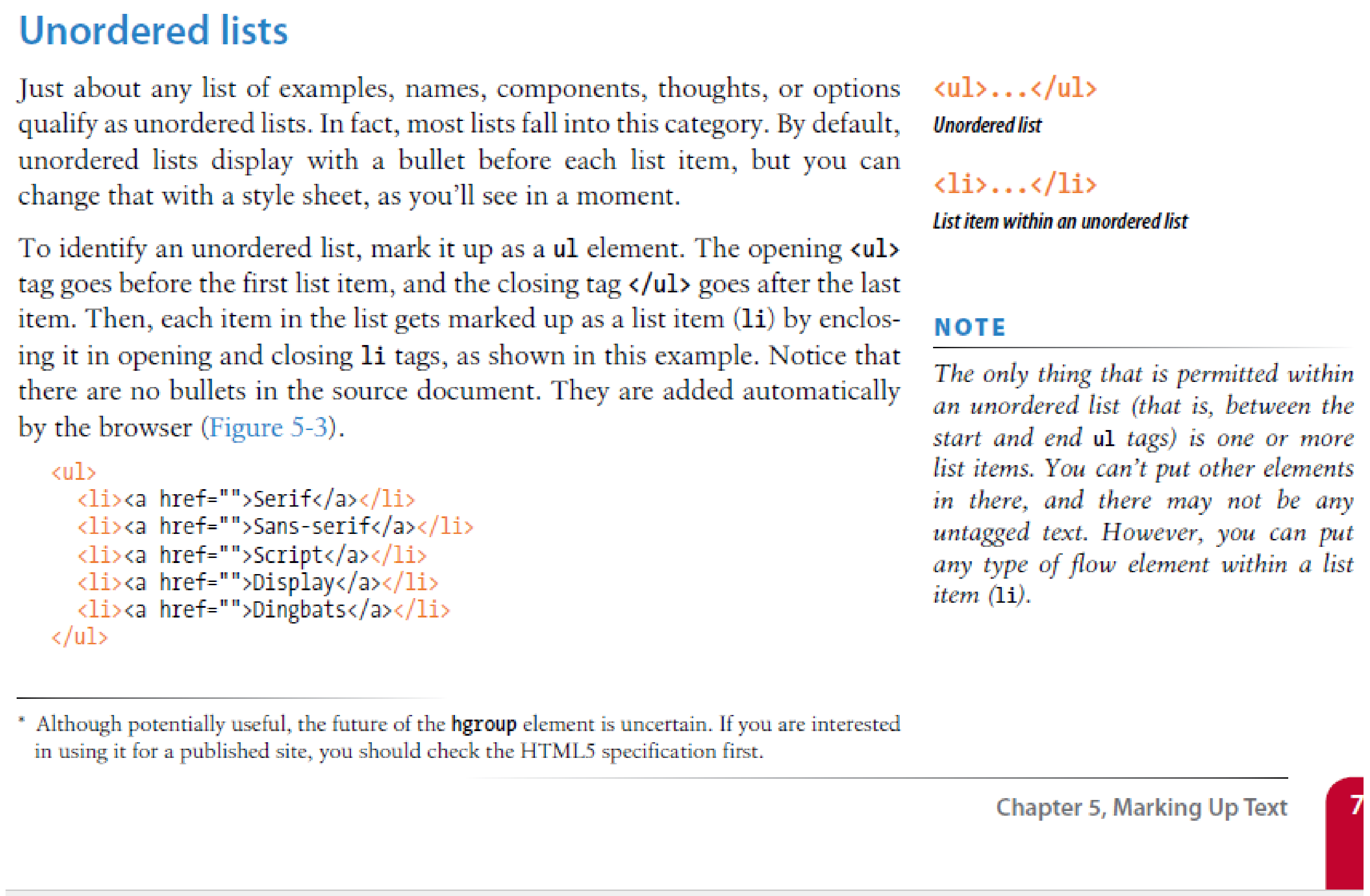
NOTE

All screenshots in this book were taken using the Chrome browser on a Mac unless otherwise noted.

Figure 5-1. The default rendering of four heading levels.



Type Design
Serif Typefaces
Serif typefaces have small slabs at the ends of letter strokes. In general, serif fonts can make large amounts of text easier to read.
Baskerville
Description
Description of the Baskerville typeface.
History
The history of the Baskerville typeface.
Georgia
Description and history of the Georgia typeface.
Sans-serif Typefaces
Sans-serif typefaces do not have slabs at the ends of strokes.



Unordered lists

Just about any list of examples, names, components, thoughts, or options qualify as unordered lists. In fact, most lists fall into this category. By default, unordered lists display with a bullet before each list item, but you can change that with a style sheet, as you'll see in a moment.

To identify an unordered list, mark it up as a **ul** element. The opening **<ul>** tag goes before the first list item, and the closing tag **</ul>** goes after the last item. Then, each item in the list gets marked up as a list item (**li**) by enclosing it in opening and closing **li** tags, as shown in this example. Notice that there are no bullets in the source document. They are added automatically by the browser (Figure 5-3).

```
<ul>
  <li><a href="">Serif</a></li>
  <li><a href="">Sans-serif</a></li>
  <li><a href="">Script</a></li>
  <li><a href="">Display</a></li>
  <li><a href="">Dingbats</a></li>
</ul>
```

`<ul>...</ul>`

Unordered list

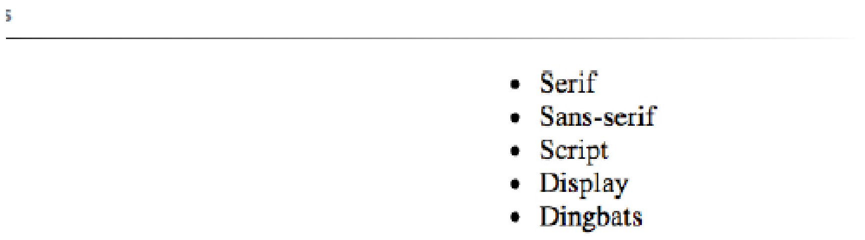
`<li>...</li>`

List item within an unordered list

NOTE

The only thing that is permitted within an unordered list (that is, between the start and end **ul** *tags) is one or more list items. You can't put other elements in there, and there may not be any untagged text. However, you can put any type of flow element within a list item (*`li`*).*

* Although potentially useful, the future of the **hgroup** element is uncertain. If you are interested in using it for a published site, you should check the HTML5 specification first.



- Serif
- Sans-serif
- Script
- Display
- Dingbats

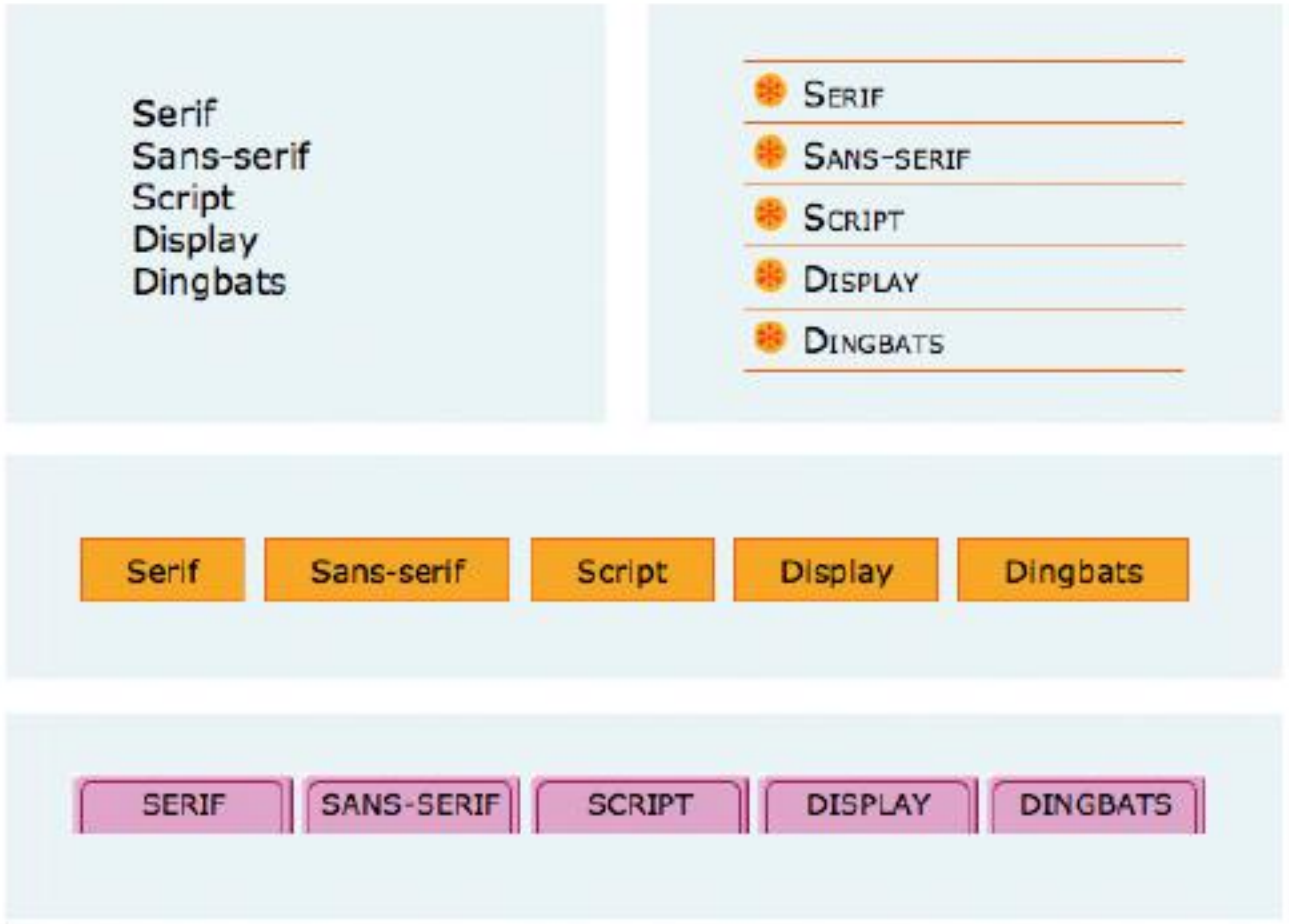
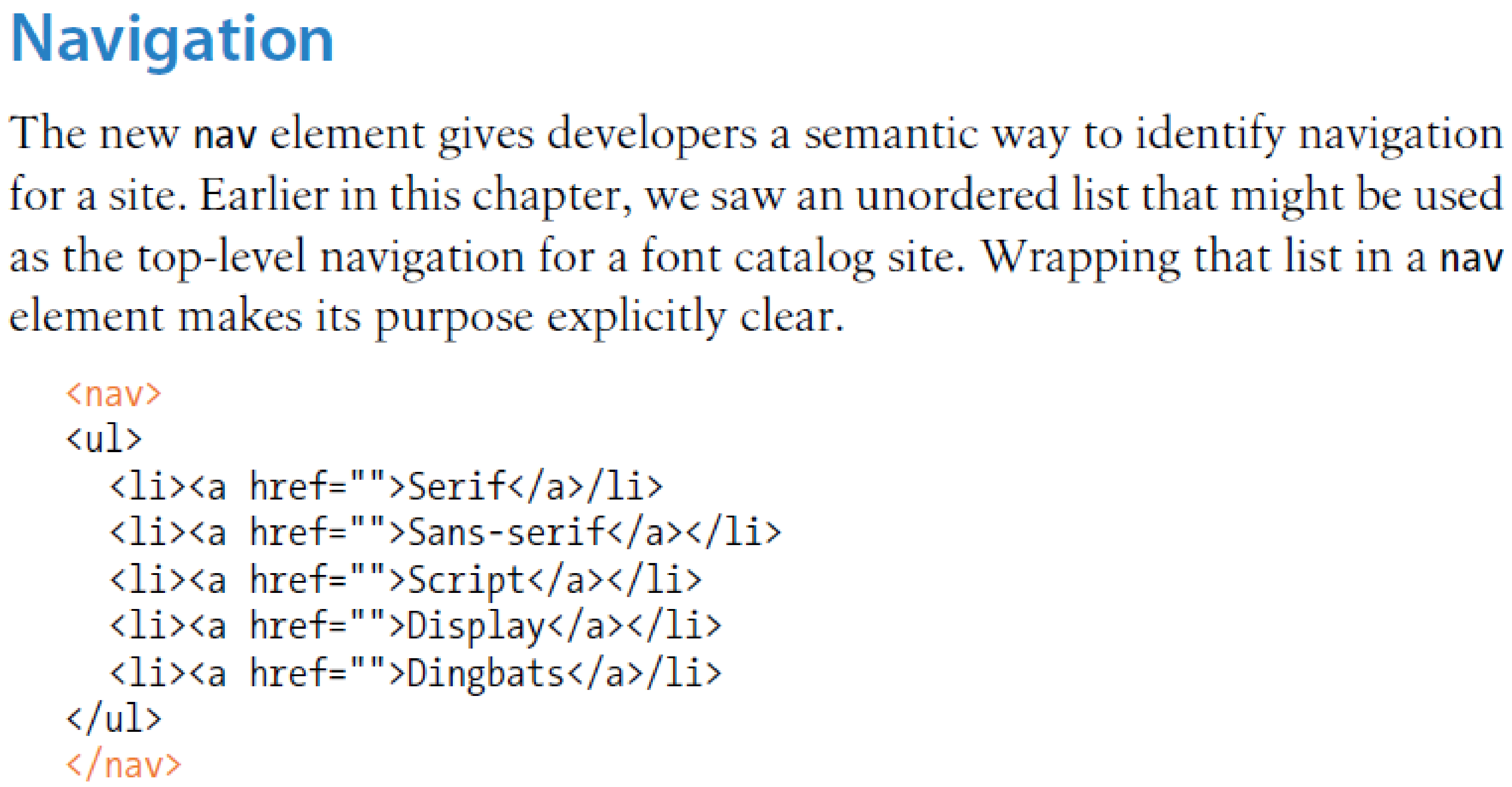


Figure 5-4. With style sheets, you can give the same unordered list many different looks.

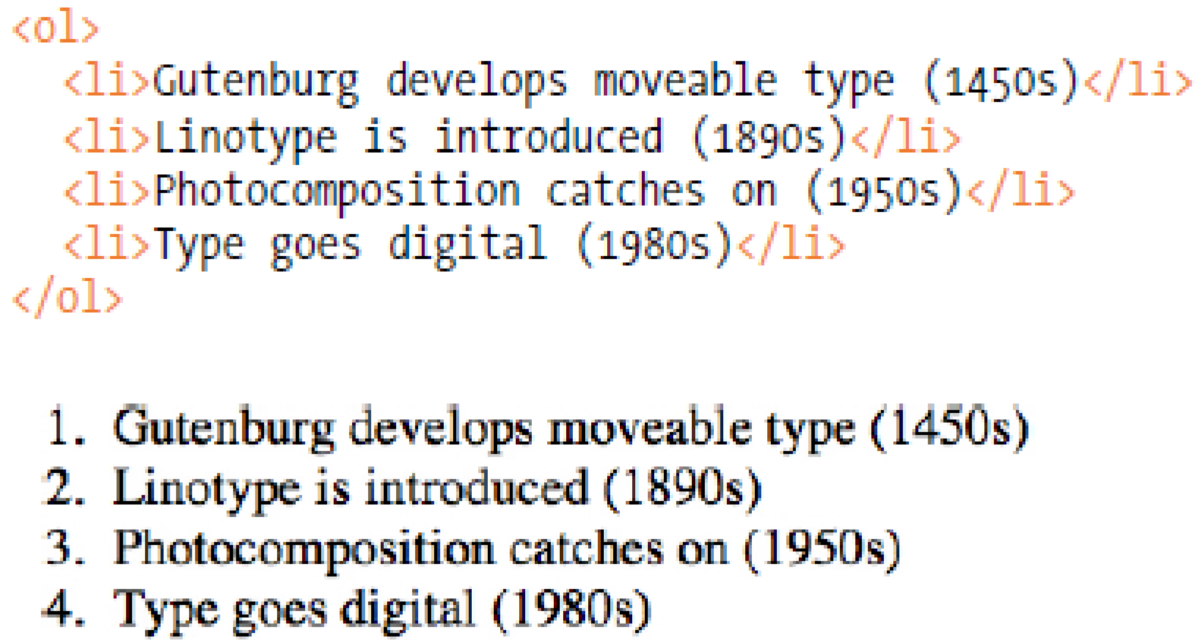


Navigation

The new **nav** element gives developers a semantic way to identify navigation for a site. Earlier in this chapter, we saw an unordered list that might be used as the top-level navigation for a font catalog site. Wrapping that list in a **nav** element makes its purpose explicitly clear.

```
<nav>
<ul>
  <li><a href="">Serif</a>/li>
  <li><a href="">Sans-serif</a></li>
  <li><a href="">Script</a></li>
  <li><a href="">Display</a></li>
  <li><a href="">Dingbats</a>/li>
</ul>
</nav>
```

Ordered Lists



```
<ol>
  <li>Gutenburg develops moveable type (1450s)</li>
  <li>Linotype is introduced (1890s)</li>
  <li>Photocomposition catches on (1950s)</li>
  <li>Type goes digital (1980s)</li>
</ol>
```

1. Gutenburg develops moveable type (1450s)
2. Linotype is introduced (1890s)
3. Photocomposition catches on (1950s)
4. Type goes digital (1980s)

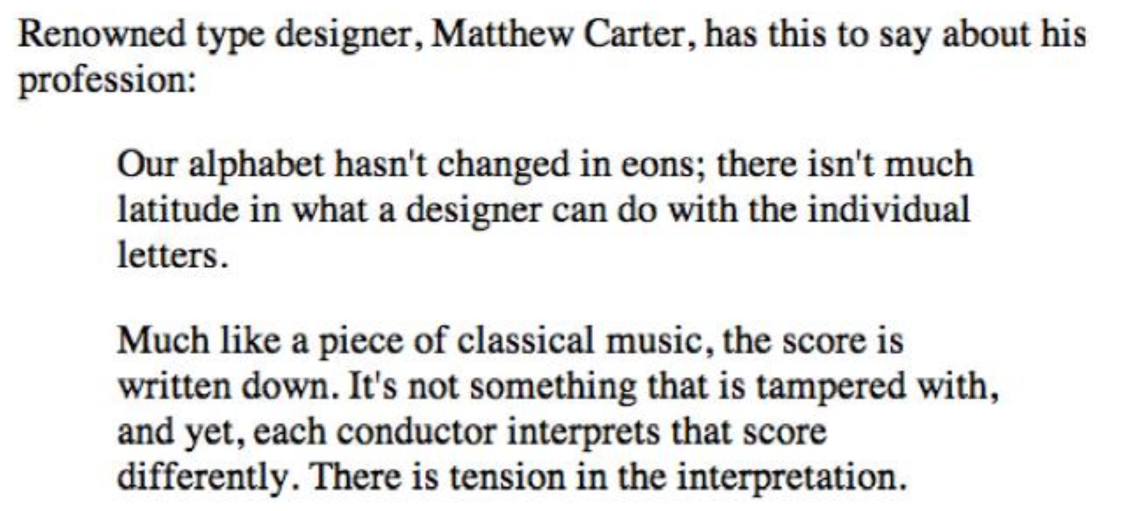
Figure 5-5. The default rendering of an ordered list. The numbers are added automatically by the browser.

```
<p>Renowned type designer, Matthew Carter, has this to say about his
profession:</p>

<blockquote>
  <p>Our alphabet hasn't changed in eons; there isn't much latitude in
what a designer can do with the individual letters.</p>

  <p>Much like a piece of classical music, the score is written
down - it's not something that is tampered with - and yet, each
conductor interprets that score differently. There is tension in
the interpretation.</p>
</blockquote>
```

Figure 5-7 shows the default rendering of the `blockquote` example. This can be altered with CSS.



Renowned type designer, Matthew Carter, has this to say about his profession:

> Our alphabet hasn't changed in eons; there isn't much latitude in what a designer can do with the individual letters.
>
> Much like a piece of classical music, the score is written down. It's not something that is tampered with, and yet, each conductor interprets that score differently. There is tension in the interpretation.

Figure 5-7. The default rendering of a `blockquote` *element.*

```
<pre>
This is                 an              example of
      text with a          lot of
                           curious
                           whitespace.
</pre>
```

Chapter 5, Marking Up

ontent Elements

```
<p>
This is                 an              example of
      text with a          lot of
                           curious
                           whitespace.
</p>
```

```
This is                 an              example of
      text with a          lot of
                           curious
                           white space.
```

This is an example of text with a lot of curious white space.

Addresses

Last, and well, least, is the **address** element that is used to create an area for contact information for the author or maintainer of the document. It is generally placed at the end of the document or in a section or article within a document. An **address** would be right at home in a **footer** element.

It is important to note that the **address** element should *not* be used for any old address on a page, such as mailing addresses. It is intended specifically for author contact information (although that could potentially be a mailing address). Following is an example of its intended use. The "a href" parts are the markup for links...we'll get to those in Chapter 6, Adding Links.

```
<address>
Contributed by <a href="../authors/robbins/">Jennifer Robbins</a>,
<a href="http://www.oreilly.com/">O'Reilly Media</a>
</address>
```

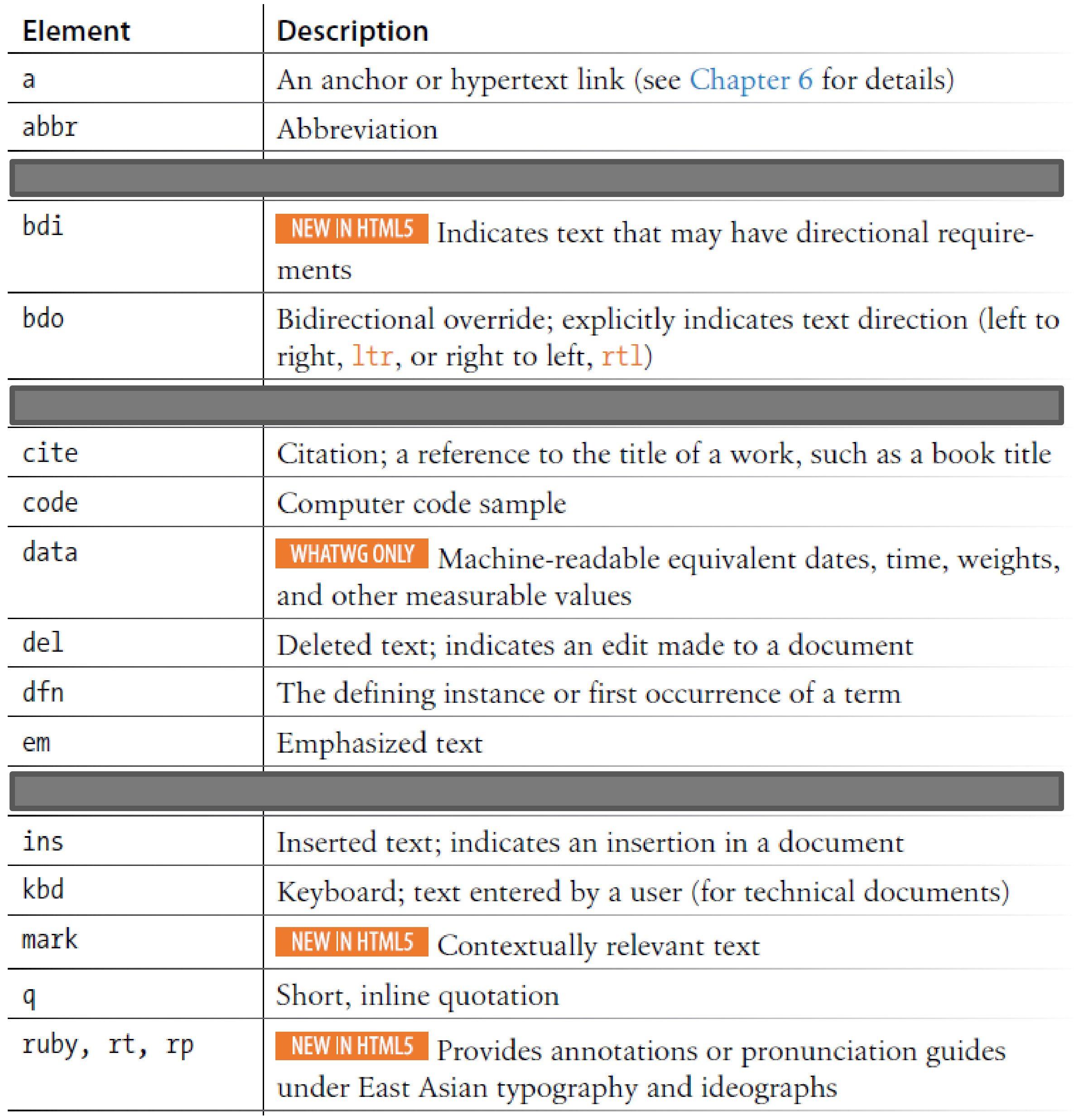
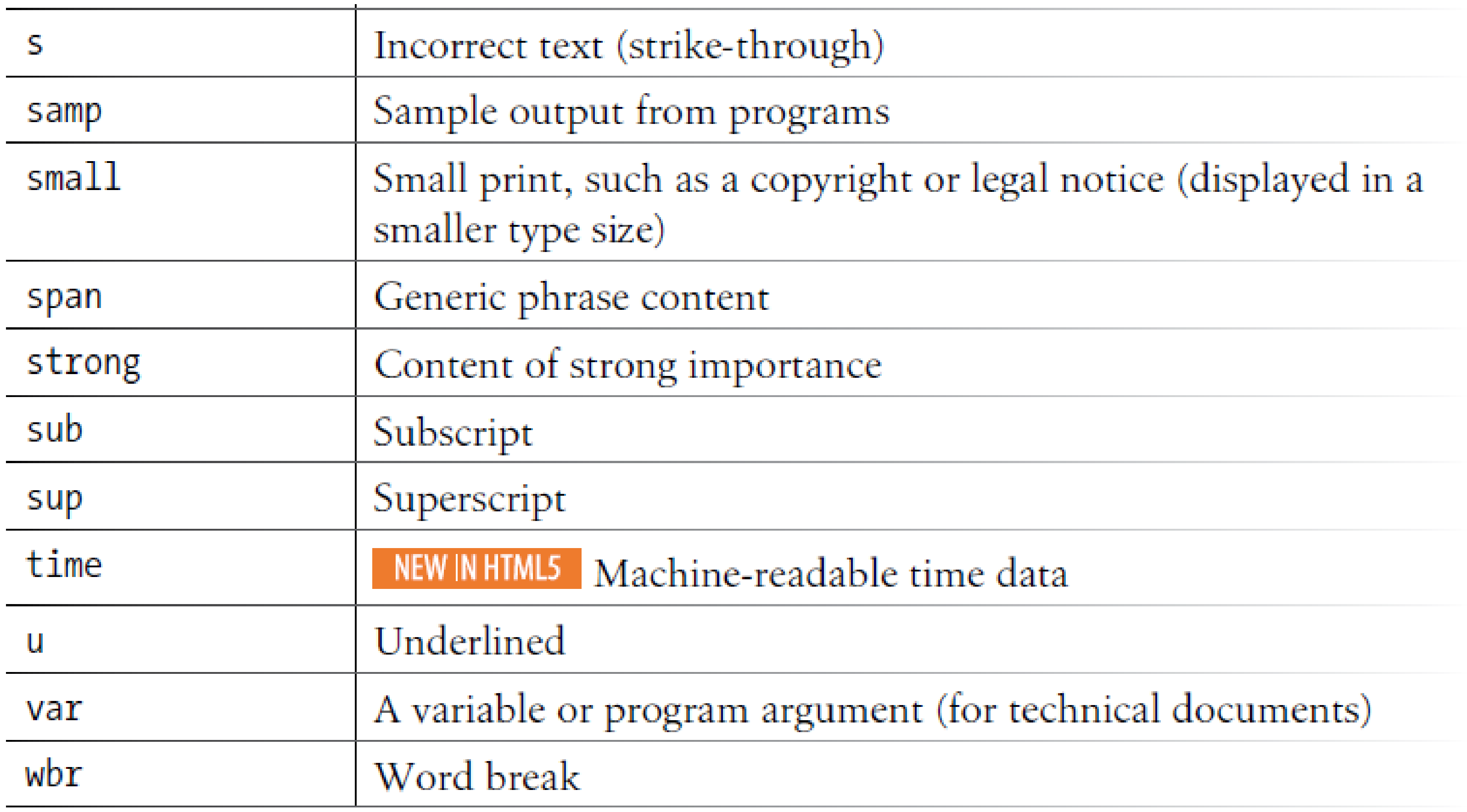


Table 5-1. Text-level semantic elements

Element	Description
a	An anchor or hypertext link (see Chapter 6 for details)
abbr	Abbreviation
[redacted]	[redacted]
bdi	NEW IN HTML5 Indicates text that may have directional requirements
bdo	Bidirectional override; explicitly indicates text direction (left to right, `ltr`, or right to left, `rtl`)
[redacted]	[redacted]
cite	Citation; a reference to the title of a work, such as a book title
code	Computer code sample
data	WHATWG ONLY Machine-readable equivalent dates, time, weights, and other measurable values
del	Deleted text; indicates an edit made to a document
dfn	The defining instance or first occurrence of a term
em	Emphasized text
[redacted]	[redacted]
ins	Inserted text; indicates an insertion in a document
kbd	Keyboard; text entered by a user (for technical documents)
mark	NEW IN HTML5 Contextually relevant text
q	Short, inline quotation
ruby, rt, rp	NEW IN HTML5 Provides annotations or pronunciation guides under East Asian typography and ideographs



`s`	Incorrect text (strike-through)
`samp`	Sample output from programs
`small`	Small print, such as a copyright or legal notice (displayed in a smaller type size)
`span`	Generic phrase content
`strong`	Content of strong importance
`sub`	Subscript
`sup`	Superscript
`time`	NEW IN HTML5 Machine-readable time data
`u`	Underlined
`var`	A variable or program argument (for technical documents)
`wbr`	Word break

SPECIAL CHARACTERS

Some special characters need to be escaped or encoded in your HTML.

They are either escaped by their entity number or entity name.

All character references begin with a & and end with a ;

Must always escape the ampersand (&) since it signifies the start of a character reference

Example, Copyright symbol

- &copy;
- &#169;

http://dev.w3.org/html5/html-author/charref

Non-breaking Spaces

One interesting character to know about is the non-breaking space (**&nbsp;**). Its purpose is to ensure that a line doesn't break between two words. So, for instance, if I mark up my name like this:

```
Jennifer&nbsp;Robbins
```

I can be sure that my first and last names will always stay together on a line.

Table 5-2. Common special characters and their character references

Character	Description	Name	Number
	Character space (nonbreaking space)	&nbsp	&#160;
&	Ampersand	&amp;	&#038;
‘	Apostrophe	&apos;	&#039;
<	Less-than symbol (useful for displaying markup on a web page)	&lt;	&#060;
>	Greater-than symbol (useful for displaying markup on a web page)	&gt;	&#062;
©	Copyright	&copy;	&#169;
®	Registered trademark	&reg;	&#174;
™	Trademark	&trade;	&#8482;
£	Pound	&pound;	&#163;
¥	Yen	&yen;	&#165;
€	Euro	&euro;	&#8364;
–	En-dash	&ndash;	&#8211;
—	Em-dash	&mdash;	&#8212;
‘	Left curly single quote	&lsquo;	&#8216;
’	Right curly single quote	&rsquo;	&#8217;
“	Left curly double quote	&ldquo;	&#8220;
”	Right curly double quote	&rdquo;	&#8221;
•	Bullet	&bull;	&#8226;
…	Horizontal ellipsis	&hellip;	&#8230;

HTML LINKS

HTML LINKS

- Anchor element "a" is used to create links to internal or external pages.
- Attribute "href" determines where the link goes when clicked.
- Attribute "title" is the link title. Used for tooltip and assistive devices.
- Can add attribute target="_blank" to have link open in new tab/window
- href can be fully qualified domain and protocol or relative to current page on your server.

HTML LINKS

- **A fully qualified href would have protocol and domain**
  - Example, href="http://www.iit.edu/"
- **A Little about pathing for relative links.**
  - If the document exists in the same folder
    - href="page2.html"
  - Document exists in subfolder
    - href="folder/page2.html"
  - Document exists in folder one level above file
    - href="../page2.html"
  - Link to a document and folder structure relative to root
    - href="/folder1/folder2/page.html"
- **Pathing explained in the book starting on page 108 please read and make sure you understand. This is a common error.**

HTML IMG

HTML IMG

- Images should be .png, .jpg or .jpeg, .gif format
- We will discuss image formats in more detail in a later class.
- .png and .gif support transparency
- .jpg no transparency
- .jpg should be 24bit RGB
- .png can be 8bit index or 24bit RGB
- .gif will be 8bit index
- Typically .jpg is better for photos and .png is better for logos and graphics. Try to not use .gif unless necessary.

HTML IMG

Image Tag

- **Element is img, <img>**
- **Empty element. No inner text node.**
- **Needs to have a src attribute, <img src=" ">**
- **Needs alt attribute to validate and for accessibility**
  - Used to provide a textual description of the image
  - <img src=" " alt="red car">
- **Can provide a width and height attribute**
  - Value is in pixels
  - <img src=" " width="200” height="100">
- **Providing the dimensions has some benefits**
  - Image reserves the proper space as it loads
  - Allows you to scale an image although not ideal solution

HTML IFRAME

- An iframe element is a embedded window to other content.
- This other content can exist on another server or the same server.
- It is treated like a whole html page embedded in the part of the page the iframe defines
- All the content in the iframe is completely separated from your page content. You can not style or interact with JavaScript any of the content of an iframe.
- Typically used for embedding content, like youtube videos.
- <iframe src="path to content" width="600" height="400"></iframe>

ITMD-361

HTML TABLES

HTML TABLES

HTML tables have historically been used for reasons other than their intended purpose.

Tables should be used to display tabular data

Tables should not be used to structure layout

- In the past many web sites were built using tables to structure the layout.
- This involved many nested tables
- This is considered a bad practice these days and would be looked at poorly by most web design companies.
- Page layout should be done all in CSS in modern web pages

HTML TABLES

- Main table defined and wrapped with <table> tag
- Table rows are defined and wrapped with <tr> tag
- Columns or cells are defined by <td> tags
- Basic structure
- Should be styled and sized with css – no style by default

```
<table>
  <tr>
    <td>R1 C1</td>
    <td>R1 C2</td>
  </tr>
  <tr>
    <td>R2 C1</td>
    <td>R2 C2</td>
  </tr>
</table>
```

R1 C1	R1 C2
R2 C1	R2 C2