5

MARKING UP TEXT

OVERVIEW

- General block elements
- Breaks
- Lists
- Page organizing elements
- Inline elements
- Generic elements
- ARIA introduction
- Escaping characters

Markup Tips

- It is important to mark up content semantically, in a way that accurately describes the content's meaning or function.
- This ensures your content is accessible in the widest range of viewing environments:
  - Desktop and mobile browsers
  - Assistive reading devices
  - Search engine indexers

Paragraphs

```
<p> </p>
```

Paragraphs are the most rudimentary elements in a text document.

```
<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>

<p>Sans-serif fonts do not have serif slabs; their
strokes are square on the end. Helvetica and Arial are
examples of sans-serif fonts. In general, sans-serif
fonts appear sleeker and more modern.</p>
```

Headings

<h#> </h#>

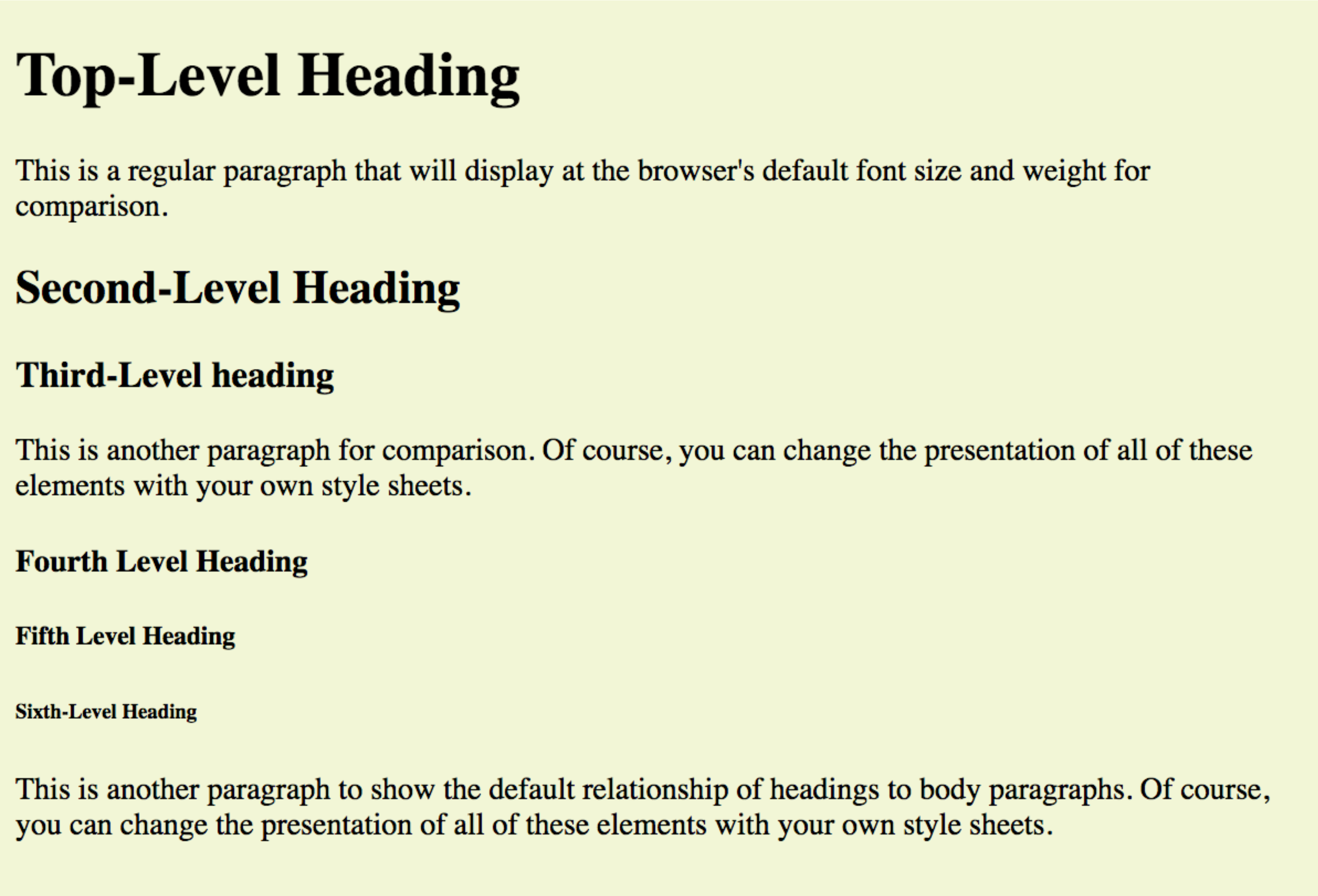
There are six levels of headings (**h1** to **h6**).

```
<h1>Top-Level Heading</h1>
<p>This is a regular paragraph that will display at the browser's
default font size and weight for comparison.</p>

<h2>Second-Level Heading</h2>
<h3>Third-Level heading</h3>
<p>This is another paragraph for comparison. Of course, you can
change the presentation of all of these elements with your own style
sheets.</p>

<h4>Fourth Level Heading</h4>
<h5>Fifth Level Heading</h5>
<h6>Sixth-Level Heading</h6>
<p>This is another paragraph to show the default relationship of
headings to body paragraphs. Of course, you can change the
presentation of all of these elements with your own style sheets.</p>
```

Headings (cont’d)



h1

Top-Level Heading

This is a regular paragraph that will display at the browser's default font size and weight for comparison.

h2

Second-Level Heading

h3

Third-Level heading

This is another paragraph for comparison. Of course, you can change the presentation of all of these elements with your own style sheets.

h4

Fourth Level Heading

h5

Fifth Level Heading

h6

Sixth-Level Heading

This is another paragraph to show the default relationship of headings to body paragraphs. Of course, you can change the presentation of all of these elements with your own style sheets.

Headings (cont’d)

- Used to create the document outline.
- Help with accessibility and search engine indexing.
- Recommended to start with **`h1`** and add subsequent levels in logical order.
- Don’t choose headings based on how they look; use a style sheet to change them.

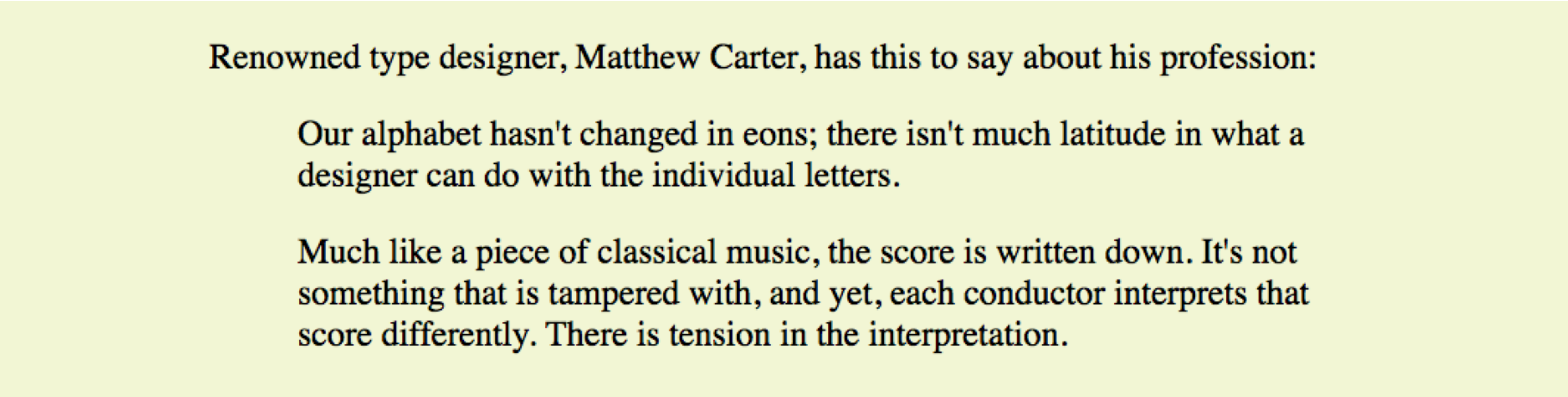
Long Quotations (blockquotes)

`<blockquote> </blockquote>`

```
<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>
```



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

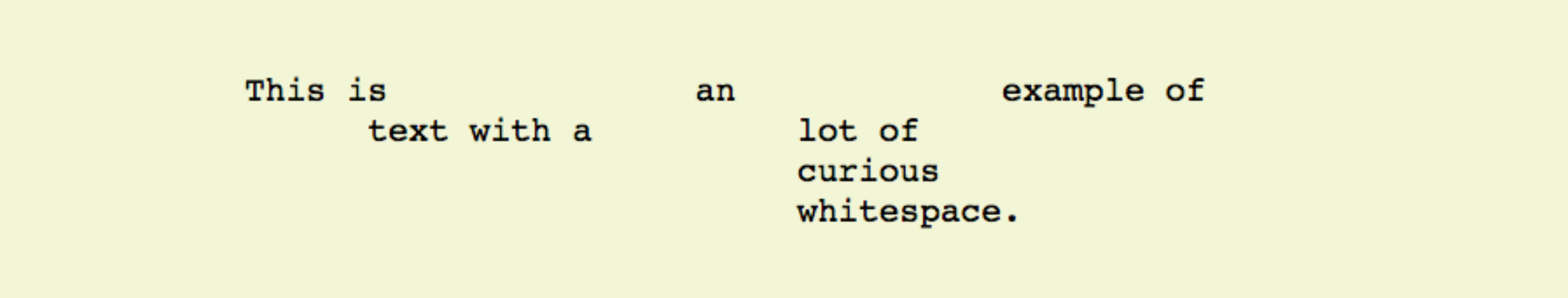
> Our alphabet hasn't changed in eons; there isn't much latitude in what a designer can do with the individual letters.
>
> Much like a piece of classical music, the score is written down. It's not something that is tampered with, and yet, each conductor interprets that score differently. There is tension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

Preformatted Text

`<pre> </pre>`

Preformatted text preserves white space when it is important for conveying meaning. By default, **`pre`** text displays in a constant-width font, such as Courier.

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



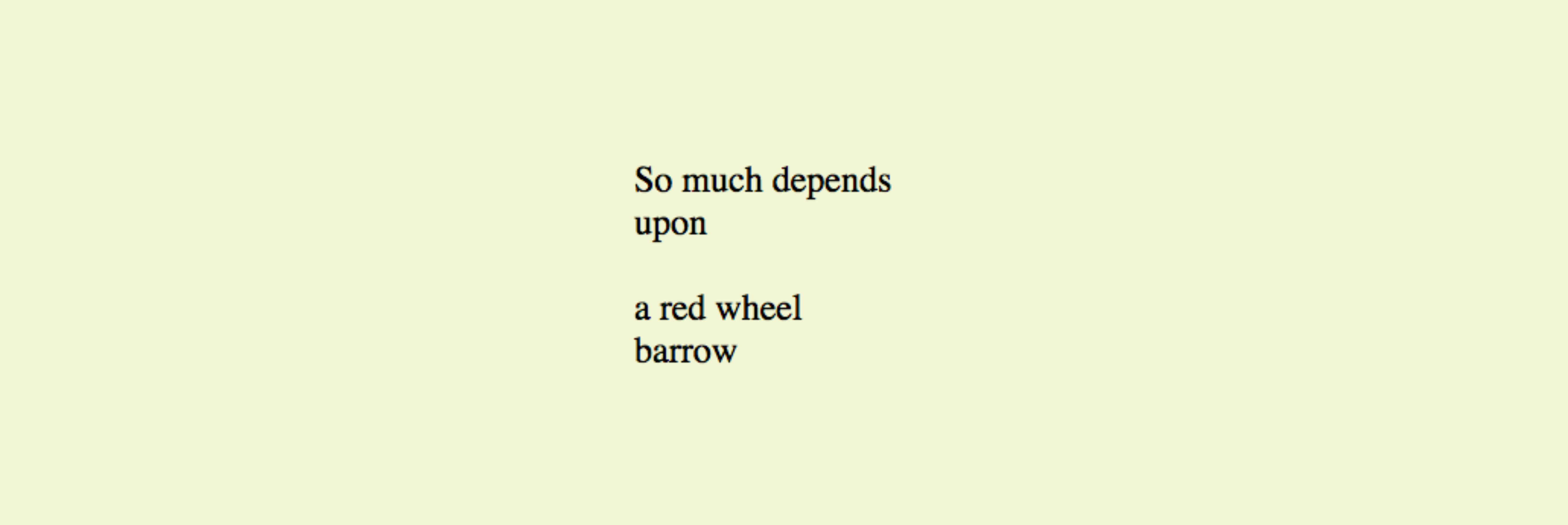
```
This is               an             example of
      text with a          lot of
                          curious
                          whitespace.
```

Line Breaks

`<br>`

The empty **br** element inserts a line break.

```
<p>So much depends <br>upon <br><br>a red wheel <br>barrow</p>
```



So much depends
upon

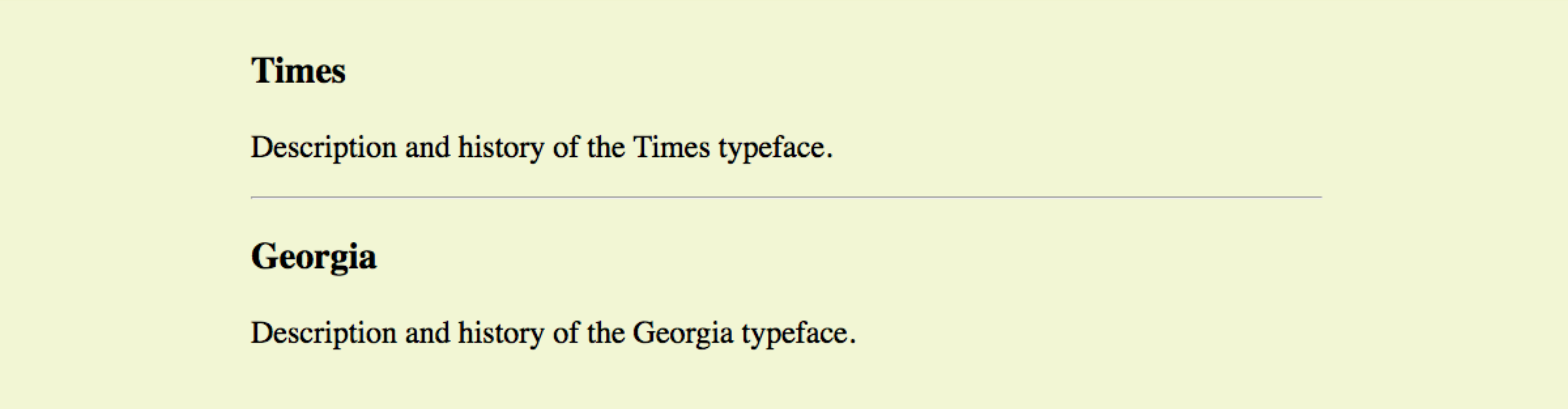
a red wheel
barrow

Thematic Breaks (Horizontal Rules)

<hr>

Indicates one topic has completed and another one is beginning. Browsers display a horizontal rule (line) in its place:

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



Times

Description and history of the Times typeface.

Georgia

Description and history of the Georgia typeface.

Lists

There are three types of lists in HTML:

- Unordered lists
- Ordered lists
- Description lists

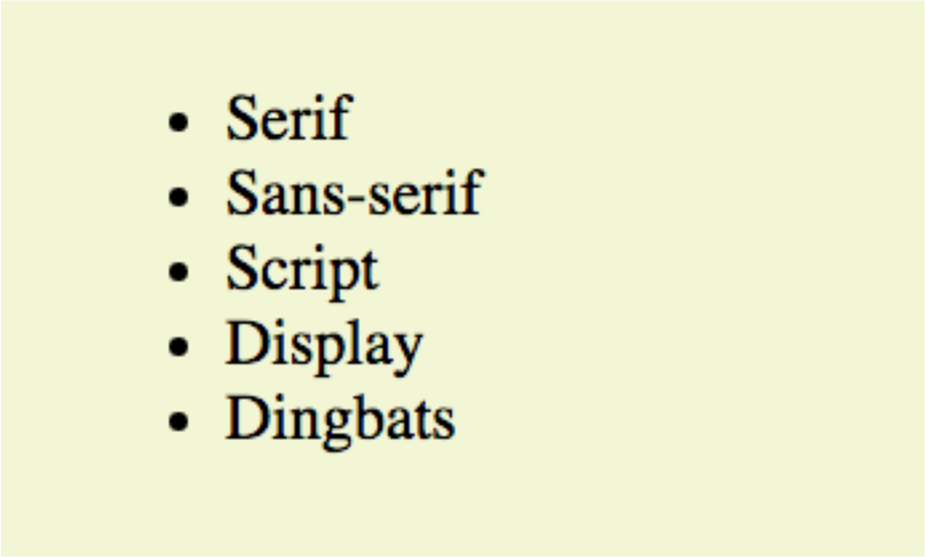
Unordered Lists

In unordered lists items may appear in any order (examples, names, options, etc.). Most lists fall into this category.

`<ul> </ul>`	Defines the whole list
`<li> </li>`	Defines each list item

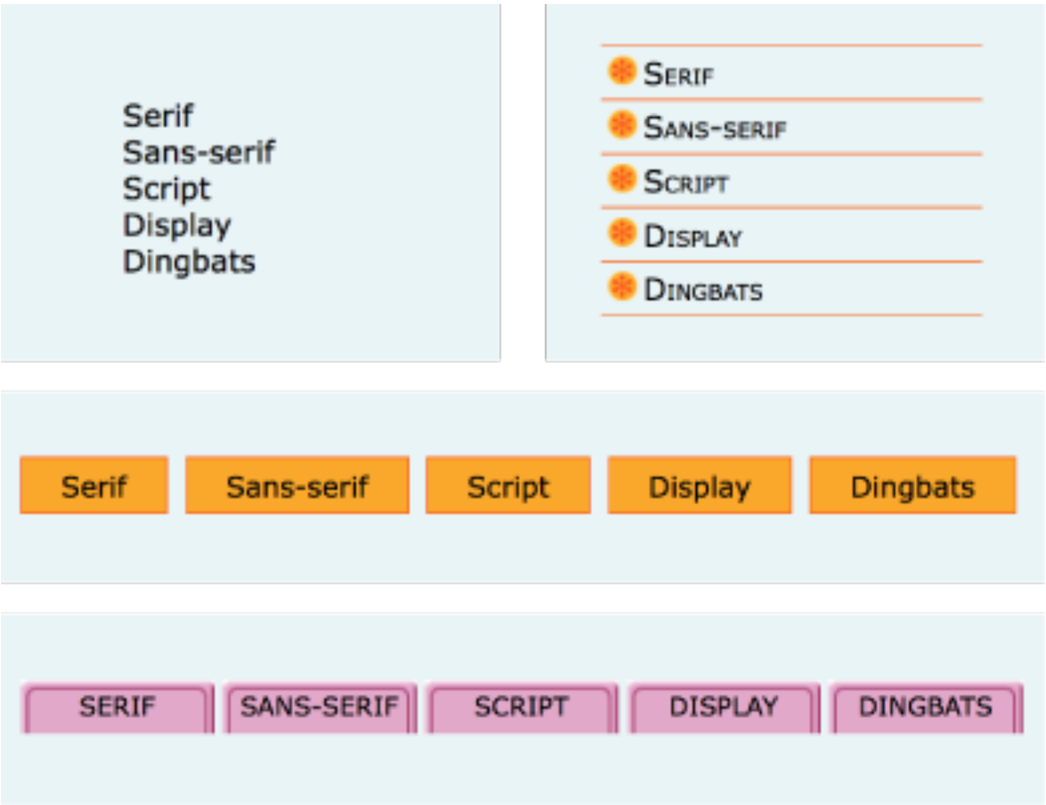
Unordered Lists (cont’d)

```
<ul>
  <li>Serif</li>
  <li>Sans-serif</li>
  <li>Script</li>
  <li>Display</li>
  <li>Dingbats</li>
</ul>
```



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

You can change the appearance of the list dramatically with style sheet rules.



Serif
Sans-serif
Script
Display
Dingbats

SERIF
SANS-SERIF
SCRIPT
DISPLAY
DINGBATS

Serif | Sans-serif | Script | Display | Dingbats

SERIF | SANS-SERIF | SCRIPT | DISPLAY | DINGBATS

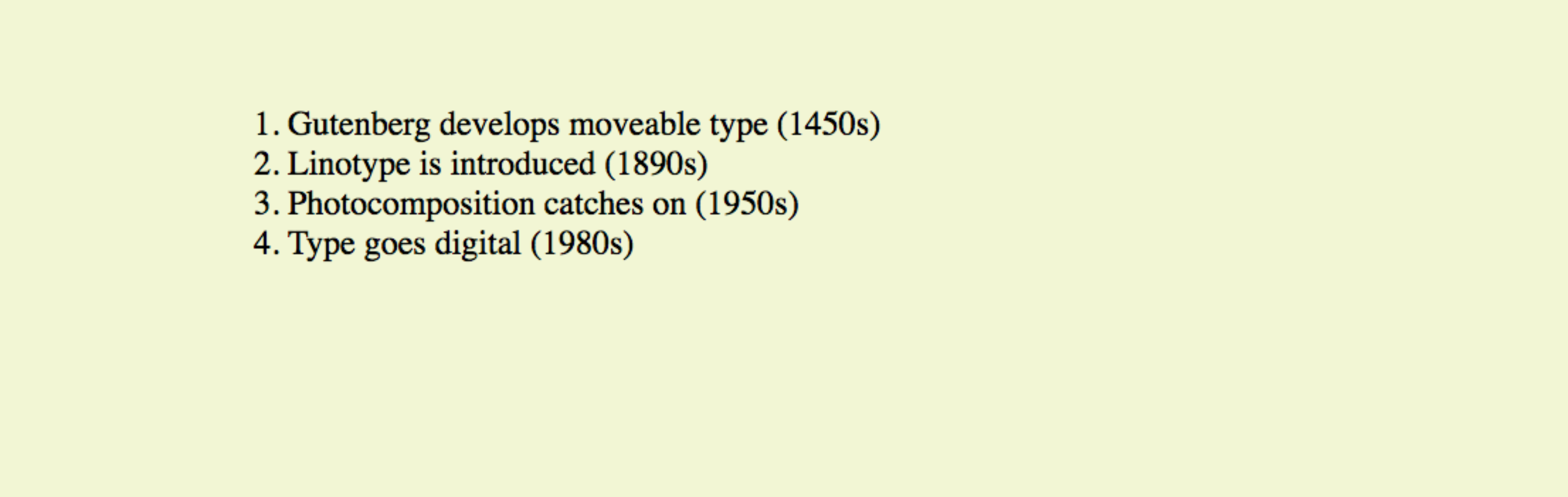
Ordered Lists

In ordered lists items occur in a particular order, such as step-by-step instructions or driving directions.

`<ol> </ol>`	Defines the whole list
`<li> </li>`	Defines each list item

Ordered Lists (cont'd.)

```
<ol>
  <li>Gutenberg 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. Gutenberg develops moveable type (1450s)
2. Linotype is introduced (1890s)
3. Photocomposition catches on (1950s)
4. Type goes digital (1980s)

Description Lists

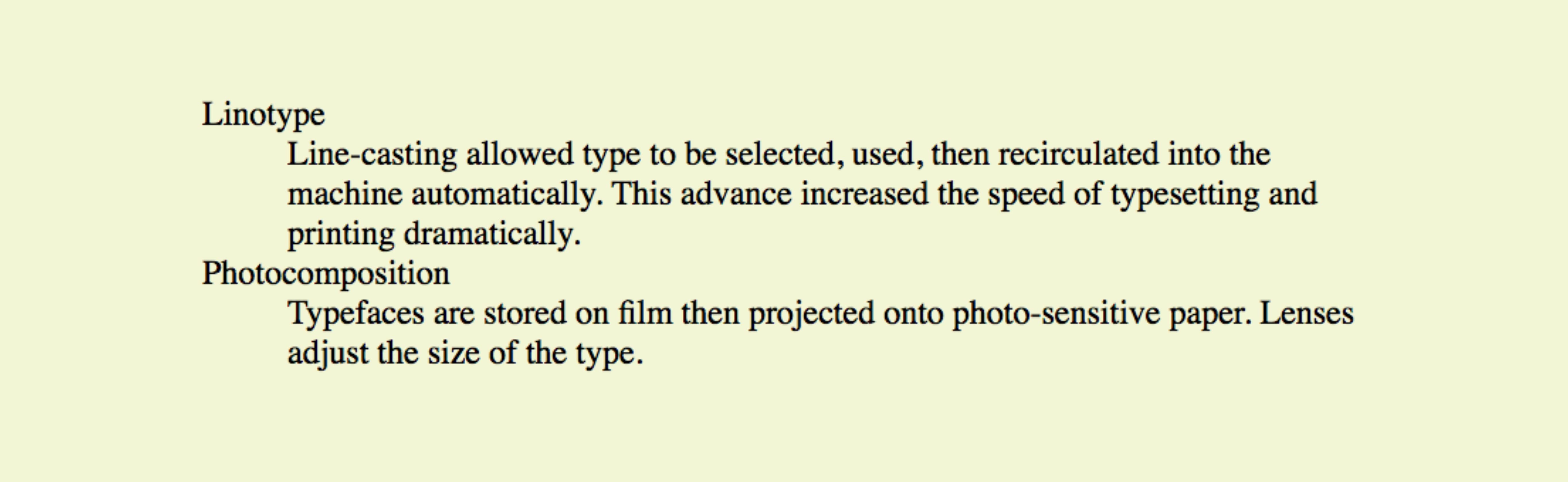
Description lists are used for any type of **name/value pairs**, such as terms/definitions, questions/answers, etc.

`<dl> </dl>`	Defines the whole list
`<dt> </dt>`	Defines a name, such as a term
`<dd> </dd>`	Defines a value, such as a definition

Description Lists (cont’d)

```
<dl>
  <dt>Linotype</dt>
  <dd>Line-casting allowed type to be selected, used, then recirculated
into the machine automatically. This advance increased the speed of
typesetting and printing dramatically.</dd>

  <dt>Photocomposition</dt>
  <dd>Typefaces are stored on film then projected onto photo-sensitive
paper. Lenses adjust the size of the type.</dd>
</dl>
```



Linotype

Line-casting allowed type to be selected, used, then recirculated into the machine automatically. This advance increased the speed of typesetting and printing dramatically.

Photocomposition

Typefaces are stored on film then projected onto photo-sensitive paper. Lenses adjust the size of the type.

Page Organizing Elements

HTML5 introduced elements that give meaning to the typical sections of a web page:

- `main`
- `header`
- `footer`
- `section`
- `article`
- `aside`
- `nav`

Main Content

```
<main> </main>
```

- Identifies the primary content of a page or application
- Helps users with screen readers get to the main content of the page
- Requires JavaScript workaround in Internet Explorer

```
<body>
<header>…</header>
<main>
  <h1>Humanist Sans Serif</h1>
  …content continues…
</main>
</body>
```

Headers and Footers

```
<header> </header>
<footer> </footer>
```

`header` identifies the introductory material that comes at the beginning of a page, section, or article (logo, title, navigation, etc.).

`footer` indicates the type of information that comes at the end of a page, section, or article (author, copyright, etc.)

Headers and Footers (cont’d)

```
<article>
  <header>
    <h1>More about WOFF</h1>
    <p>by Jennifer Robbins, <timedatetime="2017-11-11">
      November 11, 2017</time></p>
  </header>
  <!-- ARTICLE CONTENT HERE -->

  <footer>
    <p><small>Copyright &copy;2017 Jennifer Robbins.</
small></p>
    <nav>
    <ul>
      <li><a href="/">Previous</a></li>
      <li><a href="/">Next</a></li>
    </ul>
    </nav>
  </footer>
</article>
```

Sections

`<section> </section>`

`section` identifies thematic section of a page or an article. It can be used to divide up a whole page or a single article:

```
<section>
  <h2>Typography Books</h2>
  <ul>
     <li>…</li>
  </ul>
</section>

<section>
  <h2>Online Tutorials</h2>
  <p>These are the best tutorials on the Web.</p>
  <ul>
    <li>…</li>
  </ul>
</section>
```

Articles

```
<article> </article>
```

`article` is used for self-contained works that could stand alone or be used in a different context (such as syndication).

Useful for magazine/newspaper articles, blog posts, comments, etc.

```
<article>
  <h1>Get to Know Helvetica</h1>
  <section>
    <h2>History of Helvetica</h2>
    <p>…</p>
  </section>

  <section>
    <h2>Helvetica Today</h2>
    <p>…</p>
  </section>
</article>
```

Aside (Sidebar)

`<aside> </aside>`

aside identifies content that is separate from but tangentially related to the surrounding content (think of it as a sidebar).

```
<h1>Web Typography</h1>
<p>Back in 1997, there were competing font formats and tools
for making them…</p>
<p>We now have a number of methods for using beautiful fonts
on web pages…</p>
<aside>
  <h2>Web Font Resources</h2>
  <ul>
  <li><a href="http://typekit.com/">Typekit</a></li>
  <li><a href="http://fonts.google.com">Google Fonts</a></li>
  </ul>
</aside>
```

Navigation

`<nav> </nav>`

`nav` identifies the primary navigation for a site or lengthy section or article. It provides more semantic meaning than a simple unordered list.

```
<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>
```

Inline Elements

- Called **text-level semantic elements** in the spec.
- Describe the types of elements that appear in the flow of text.

a
em
strong
q
abbr
cite
dfn
code
var
samp
kbd
sub/sup
mark
time
data
ins/del
b
i
s
u
small
bdi/bdo
data
span

Inline Elements

Emphasis

```
<em> </em>
```

Text that should be emphasized. Usually displayed in italics.

```
<p><em>Arlo</em> is very smart.</p>
<p>Arlo is <em>very</em> smart.</p>
```

```
<strong></strong>
```

Text that is important, serious, or urgent. Usually displayed in bold.

```
<p>When returning the car, <strong>drop the keys in the
red box by the front desk</strong>.</p>
```

TIP: Use these elements semantically, not to achieve font styles. Think of how it would be read with a screen reader.

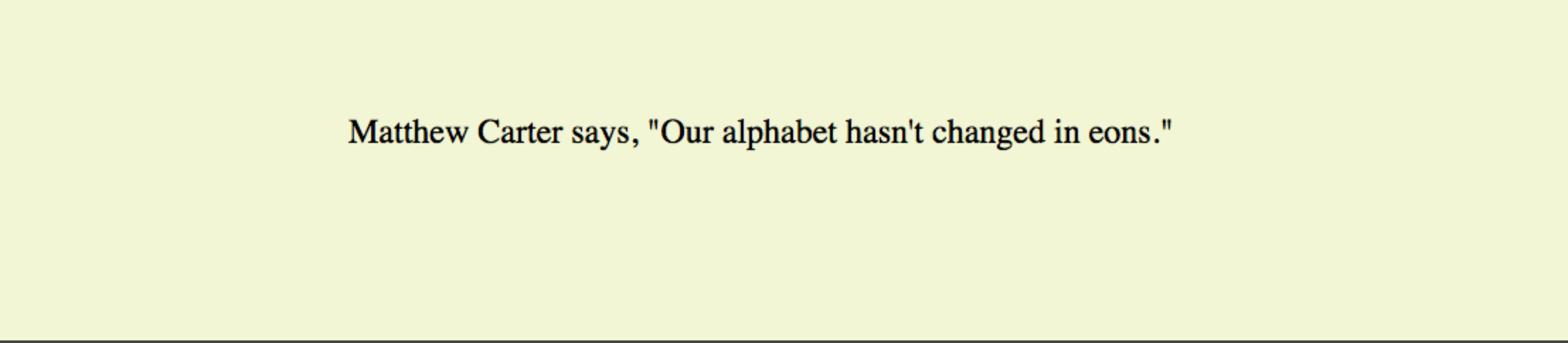
Inline Elements

Short Quotations

`<q> </q>`

For quoted phrases in the flow of text. Browsers add appropriate quotation marks automatically.

```
<p>Matthew Carter says, <q>Our alphabet hasn't changed in
eons.</q></p>
```



Matthew Carter says, "Our alphabet hasn't changed in eons."

Inline Elements

Abbreviations and Acronyms

```
<abbr> </abbr>
```

The **title** attribute provides the long version of a shortened term, which is helpful for search engines and assistive devices.

```
<abbr title="Points">pts.</abbr>

<abbr title="American Type Founders">ATF</abbr>
```

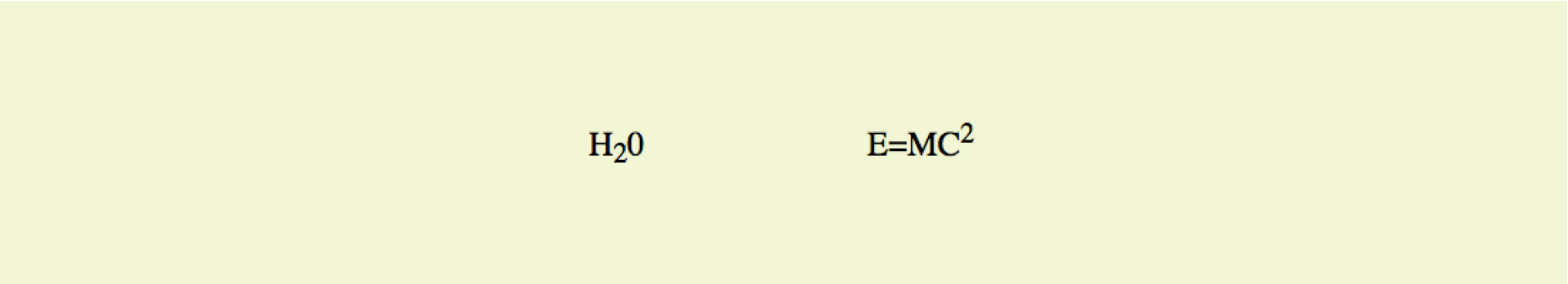
Superscript and Subscript

```
<sup> </sup>
<sub> </sub>
```

Causes the selected text to display in a smaller size and slightly above (**sup**) or below (**sub**) the baseline.

```
<p>H<sub>2</sub>0</p>

<p>E=MC<sup>2</sup></p>
```



$H_2 0$

$E=MC^2$

Inline Elements

Citations

```
<cite> </cite>
```

Identifies a reference to another document.

```
<p>Passages of this article were inspired by <cite>The
Complete Manual of Typography</cite> by James Felici.</p>
```

Inline Elements

Defining Terms

```
<dfn> </dfn>
```

Identifies the first and defining instance of a word in a document. There is no default rendering, so you need to format them using style sheets.

```
<p><dfn>Script typefaces</dfn> are based on handwriting.</p>
```

Inline Elements

Code-Related Elements

`<code> </code>`	Code in the flow of text
`<var> </var>`	Variables
`<samp> </samp>`	Program sample
`<kbd> </kbd>`	User-entered keyboard strokes

New Semantic Definitions for Old Presentational Inline Elements

`<b> </b>` Phrases that need to stand out without added emphasis or importance (*bold*)

`<i> </i>` Phrases in a different voice or mood than the surrounding text (i*talic*)

`<s> </s>` Text that is incorrect (*strike-through*)

`<u> </u>` Underlined text, when underlining has semantic purpose

`<small> </small>` Addendum or side note (*smaller text size*)

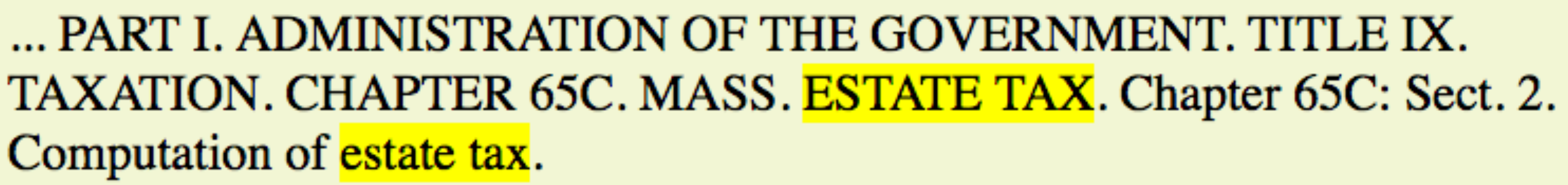
Inline Elements

Highlighted Text

```
<mark> </mark>
```

For phrases that may be particularly relevant to the reader (for example, when displaying search results):

```
<p> ... PART I. ADMINISTRATION OF THE GOVERNMENT. TITLE IX.
TAXATION. CHAPTER 65C. MASS. <mark>ESTATE TAX</mark>. Chapter
65C: Sect. 2. Computation of <mark>estate tax</mark>.</p>
```



... PART I. ADMINISTRATION OF THE GOVERNMENT. TITLE IX. TAXATION. CHAPTER 65C. MASS. ESTATE TAX. Chapter 65C: Sect. 2. Computation of estate tax.

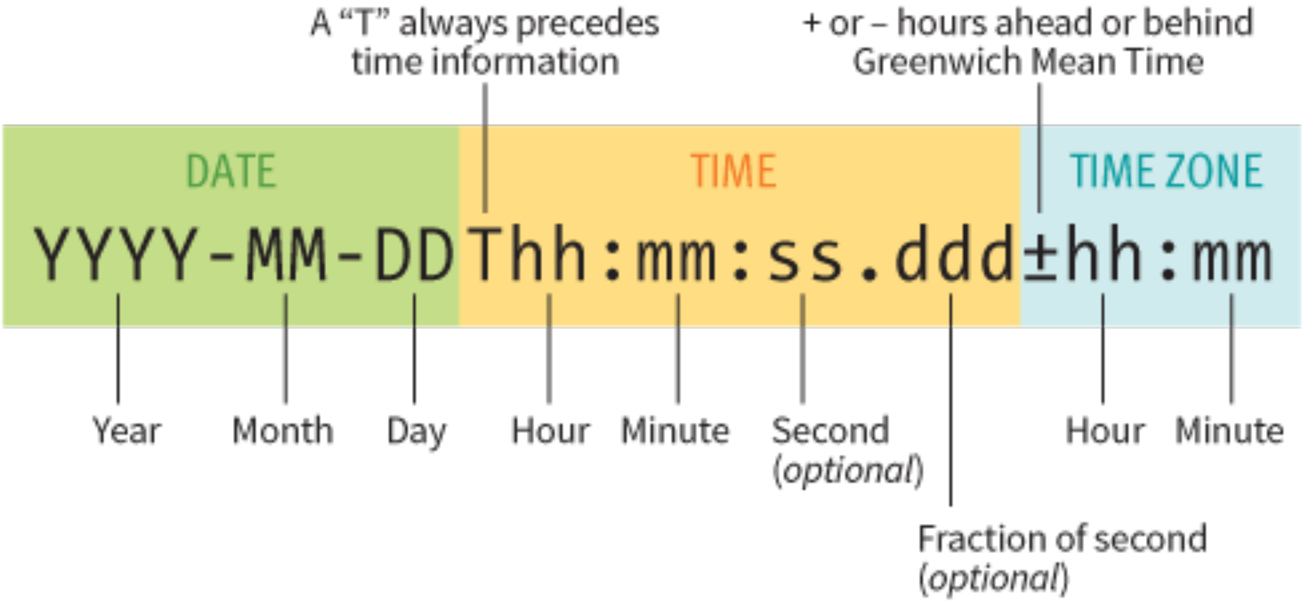
Inline Elements

Dates and Times

`<time> </time>`

Provides machine-readable equivalents for dates and times. The **`datetime`** attribute specifies the date/time information in a standardized time format:

```
<time datetime="1970-09-05T01:11:00">Sept.5, 1970,
1:11a.m.</time>
```


A "T" always precedes time information
+ or – hours ahead or behind Greenwich Mean Time
DATE
TIME
TIME ZONE
YYYY-MM-DDThh:mm:ss.ddd±hh:mm
Year
Month
Day
Hour
Minute
Second (optional)
Fraction of second (optional)
Hour
Minute

Inline Elements

Machine-Readable Information

```
<data> </data>
```

Helps computers make sense of content.

The **value** attribute provides the machine-readable information.

```
<data value="12">Twelve</data>

<data value="978-1-449-39319-9">CSS: The Definitive
Guide</data>
```

Inline Elements

Inserted and Deleted Content

```
<ins> </ins>
<del> </del>
```

Markup for edits indicating parts of a document that have been inserted or deleted:

```
Chief Executive Officer: <del title="retired">Peter
Pan</del><ins>Pippi Longstocking</ins>
```

Generic Elements

`<div> </div>`

Indicates division of content (generally block-level)

`<span> </span>`

Indicates a word or phrase

- Generic elements are given semantic meaning with the `id` and `class` attributes.
- They are useful for creating “hooks” for scripts and style rules.

Div Example

Use the **`div`** element to create a logical grouping of content or elements on the page.

It indicates that they belong together in some sort of conceptual unit or should be treated as a unit by CSS or JavaScript.

```
<div class="listing">
  <img src="images/felici-cover.gif" alt="">

  <p><cite>The Complete Manual of Typography</cite>, James
Felici</p>

  <p>A combination of type history and examples of good and
bad type design.</p>
</div>
```

Span Example

Use the **span** element for text and other inline elements for which no existing inline element currently exists.

In this example, a **span** is used to add semantic meaning to telephone numbers:

```
<ul>
  <li>John: <span class="tel">999.8282</span></li>
  <li>Paul: <span class="tel">888.4889</span></li>
  <li>George: <span class="tel">888.1628</span></li>
  <li>Ringo: <span class="tel">999.3220</span></li>
</ul>
```

id and class Attributes

`id`
Assigns a unique identifier to the element.

`class`
Classifies elements into a conceptual group.

Use the **id** attribute to identify.
Use the **class** attribute to classify.

NOTE: **id** and **class** can be used with *all* HTML elements.

The id Attribute

The value of an **id** attribute must be used only once in a document.

Here it identifies a listing for a particular book by its ISBN:

```
<div id="ISBN0321127307">
  <img src="felici-cover.gif" alt="">
  <p><cite>The Complete Manual of Typography</cite>, James
Felici</p>
  <p>A combination of type history …</p>
</div>
```

Here it identifies a particular section of a document:

```
<section id="news">
  <!-- news items here -->
</section>
```

The class Attribute

A **`class`** value may be used by multiple elements to put them in conceptual groups for scripting or styling.

Here several book listings are classified as a “listing”:

```
<div id="ISBN0321127307" class="listing">
 …
</div>

<div id="ISBN0881792063" class="listing">
…
</div>
```

An element may belong to more than one class. Separate class values with character spaces:

```
<div id="ISBN0321127307" class="listing book nonfiction">
```

Brief ARIA Introduction

ARIA (Accessible Rich Internet Applications)
is a standardized set of attributes for making pages easier to navigate and use with assistive devices.

ARIA defines **roles**, **states**, and **properties** that developers can add to markup and scripts to provide richer information.

www.w3.org/TR/html-aria

ARIA Roles

Roles describe or clarify an element’s function in the document.

Examples: `alert`, `button`, `dialog`, `slider`, and `menubar`

```
<div id="status" role="alert">You are no longer
connected to the server.</div>
```

ARIA States and Properties

- ARIA defines a long list of **states** and **properties** that apply to interactive elements and dynamic content.

- Properties values are likely to be stable (example: `aria-labelledby`).

- States have values that are likely to change as the user interacts with the content (example: `aria-selected`).

Escaping Characters

Escaping a character means representing it by its named or numeric **character entity** in the source.

- Some characters must be escaped because they will be mistaken for code (example: the < character would be parsed as the start of an HTML tag).

- Some characters are invisible or just easier to escape than find on the keyboard.

Character Entity References

Character entities always begin with **&** and end with **;**.

Named entities
Use a predefined name for the character
(example: `&lt;` for the less-than symbol **<**)

Numeric entities
Use an assigned numeric value that corresponds to its position in a coded character set, such as UTF-8
(example: `&#060;` for the less-than symbol **<**).

A complete list of HTML named entities and their Unicode code-points is at www.w3.org/TR/html5/syntax.html#named-character-references.

Escaping HTML Syntax Characters

Always escape **<**, **>**, and **&** characters in content.
Escape **"** and **'** when they are in attribute values.

Character	**Description**	**Entity name**	**Decimal**	**Hexadecimal**
<	Less-than symbol	&lt;	&#060;	&#x3C;
>	Greater-than symbol	&gt;	&#062;	&#x3E;
"	Quote	&quot;	&#160;	&#x22;
'	Apostrophe	&apos;	&#039;	&#x27;
&	Ampersand	&amp;	&#038;	&#x26;

(Additional non-required character entities are listed in Chapter 5.)