3.2: More CSS

- 3.1: Basic CSS
- 3.2: More CSS
The HTML id attribute (3.2.2)

Our mission is to provide the most spectacular spatulas and splurge on our specials until our customers "explode" with splendor!

- allows you to give a unique ID to any element on a page
- each ID must be unique; can only be used once in the page

Linking to sections of a web page

Visit textpad.com to get the TextPad editor.

- a link target can include an ID at the end, preceded by a #
- browser will load that page and scroll to element with given ID
CSS ID selectors

Spatula City! Spatula City!

Our mission is to provide the most spectacular spatulas and splurge on our specials until our customers “esplode” with splendor!

- applies style only to the paragraph that has the ID of `mission`
- element can be specified explicitly: `p#mission`

The HTML class attribute (3.2.3)

Spatula City! Spatula City!

See our spectacular spatula specials!

Today only: satisfaction guaranteed.

- classes are a way to group some elements and give a style to only that group (“I don't want ALL paragraphs to be yellow, just these three...”)
- unlike an id, a class can be reused as much as you like on the page
CSS class selectors

CSS

```css
.special {
    background-color: yellow;
    font-weight: bold;
}
p.shout {
    color: red;
    font-family: cursive;
}
```

Output

```
Spatula City! Spatula City!

See our spectacular spatula specials!

Today only: satisfaction guaranteed.
```

- applies rule to any element with class `special` or a `p` with class `shout`

Multiple classes

HTML

```
<h2 class="shout">Spatula City! Spatula City!</h2>
<p class="special">See our spectacular spatula specials!</p>
<p class="special shout">Satisfaction guaranteed.</p>
<p class="shout">We'll beat any advertised price!</p>
```

Output

```
Spatula City! Spatula City!

See our spectacular spatula specials!

Satisfaction guaranteed.

We'll beat any advertised price!
```

- an element can be a member of multiple classes (separated by spaces)
Buy early, buy often!

<table>
<thead>
<tr>
<th>class</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:active</td>
<td>an activated or selected element</td>
</tr>
<tr>
<td>:focus</td>
<td>an element that has the keyboard focus</td>
</tr>
<tr>
<td>:hover</td>
<td>an element that has the mouse over it</td>
</tr>
<tr>
<td>:link</td>
<td>a link that has not been visited</td>
</tr>
<tr>
<td>:visited</td>
<td>a link that has already been visited</td>
</tr>
<tr>
<td>:first-letter</td>
<td>the first letter of text inside an element</td>
</tr>
<tr>
<td>:first-line</td>
<td>the first line of text inside an element</td>
</tr>
<tr>
<td>:first-child</td>
<td>an element that is the first one to appear inside another</td>
</tr>
</tbody>
</table>

4.1: Styling Page Sections

- 4.1: Styling Page Sections
- 4.2: Introduction to Layout
- 4.3: Floating Elements
- 4.4: Sizing and Positioning
Motivation for page sections

- want to be able to style individual elements, groups of elements, sections of text or of the page
- (later) want to create complex page layouts

Sections of a page: `<div>` (4.1.1)

*a section or division of your HTML page (block)*

```html
<div class="shout">
  <h2>Spatula City! Spatula City!</h2>
  <p class="special">See our spectacular spatula specials!</p>
  <p>We'll beat any advertised price!</p>
</div>
```

Spatula City! Spatula City!

See our spectacular spatula specials!

We’ll beat any advertised price!

- a tag used to indicate a logical section or area of a page
- has no appearance by default, but you can apply styles to it
Inline sections: <span> (4.1.2)

an inline element used purely as a range for applying styles

<h2>Spatula City! Spatula City!</h2>
<p>See our <span class="special">spectacular</span> spatula specials!</p>
<p>We'll beat <span class="shout">any advertised price</span>!</p>

- has no onscreen appearance, but you can apply a style or ID to it, which will be applied to the text inside the span

CSS context selectors (4.1.3)

```
selector1 selector2 {
  properties
}
```

- applies the given properties to selector2 only if it is inside a selector1 on the page

```
selector1 > selector2 {
  properties
}
```

- applies the given properties to selector2 only if it is <em>directly</em> inside a selector1 on the page (selector1 tag is immediately inside selector2 with no tags in between)
Context selector example

```html
<p>Shop at <strong>Hardwick's Hardware</strong>...</p>
<ul>
  <li>The <strong>best</strong> prices in town!</li>
  <li>Act while supplies last!</li>
</ul>
```

```css
li strong { text-decoration: underline; }
```

Shop at Hardwick's Hardware...

- The <strong>best</strong> prices in town!
- Act while supplies last!

More complex example

```html
<div id="ad">
  <p>Shop at <strong>Hardwick's Hardware</strong>...</p>
  <ul>
    <li class="important">The <strong>best</strong> prices in town!</li>
    <li>Act <strong>while supplies last!</strong></li>
  </ul>
</div>
```

```css
#ad li.important strong { text-decoration: underline; }
```

Shop at Hardwick's Hardware...

- The <strong>best</strong> prices in town!
- Act <strong>while supplies last!</strong>
4.2: Introduction to Layout

- 4.1: Styling Page Sections
- 4.2: Introduction to Layout
- 4.3: Floating Elements
- 4.4: Sizing and Positioning

The CSS Box Model (4.2.1)

- for layout purposes, every element is composed of:
  - the actual element's content
  - a border around the element
  - padding between the content and the border
    (inside)
  - a margin between the border and other content
    (outside)
- width = content width + L/R padding + L/R border + L/R margin
- height = content height + T/B padding + T/B border + T/B margin
  - IE6 doesn't do this right
Document flow - block elements

- h1
- h2
- p
- h2
- p
- p

Document flow - inline elements

- p
  - text
  - em
  - text
  - text
  - a
  - text
Document flow - a larger example

CSS properties for borders

```css
h2 { border: 5px solid red; }
```

This is a heading.

<table>
<thead>
<tr>
<th>property</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>border</td>
<td>thickness/style/size of border on all 4 sides</td>
</tr>
</tbody>
</table>

- **thickness** (specified in px, pt, em, or thin, medium, thick)
- **style** (none, hidden, dotted, dashed, double, groove, inset, outset, ridge, solid)
- **color** (specified as seen previously for text and background colors)
## More border properties

<table>
<thead>
<tr>
<th>property</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>border-color, border-width, border-style</td>
<td>specific properties of border on all 4 sides</td>
</tr>
<tr>
<td>border-bottom, border-left, border-right, border-top</td>
<td>all properties of border on a particular side</td>
</tr>
<tr>
<td>border-bottom-color, border-bottom-style, border-bottom-width, border-left-color, border-left-style, border-left-width, border-right-color, border-right-style, border-right-width, border-top-color, border-top-style, border-top-width</td>
<td>properties of border on a particular side</td>
</tr>
</tbody>
</table>

Complete list of border properties

### Border example 2

```css
h2 {
    border-left: thick dotted #CC0088;
    border-bottom-color: rgb(0, 128, 128);
    border-bottom-style: double;
}
```

This is a heading.

- each side's border properties can be set individually
- if you omit some properties, they receive default values (e.g. `border-bottom-width` above)
CSS properties for padding

<table>
<thead>
<tr>
<th>property</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>padding</td>
<td>padding on all 4 sides</td>
</tr>
<tr>
<td>padding-bottom</td>
<td>padding on bottom side only</td>
</tr>
<tr>
<td>padding-left</td>
<td>padding on left side only</td>
</tr>
<tr>
<td>padding-right</td>
<td>padding on right side only</td>
</tr>
<tr>
<td>padding-top</td>
<td>padding on top side only</td>
</tr>
</tbody>
</table>

Complete list of padding properties

Padding example 1

```css
p { padding: 20px; border: 3px solid black; }

h2 { padding: 0px; background-color: yellow; }
```

This is the first paragraph

This is the second paragraph

This is a heading
Padding example 2

```css
p {
  padding-left: 200px; padding-top: 30px;
  background-color: fuchsia;
}
```

This is the first paragraph

This is the second paragraph

- each side's padding can be set individually
- notice that padding shares the background color of the element

CSS properties for margins

<table>
<thead>
<tr>
<th>property</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>margin</td>
<td>margin on all 4 sides</td>
</tr>
<tr>
<td>margin-bottom</td>
<td>margin on bottom side only</td>
</tr>
<tr>
<td>margin-left</td>
<td>margin on left side only</td>
</tr>
<tr>
<td>margin-right</td>
<td>margin on right side only</td>
</tr>
<tr>
<td>margin-top</td>
<td>margin on top side only</td>
</tr>
</tbody>
</table>

Complete list of margin properties
Margin example 1

```css
p {
  margin: 50px;
  background-color: fuchsia;
}
```

This is the first paragraph

This is the second paragraph

- notice that margins are always transparent
  (they don't contain the element's background color, etc.)

Margin example 2

```css
p {
  margin-left: 8em;
  background-color: fuchsia;
}
```

This is the first paragraph

This is the second paragraph

- each side's margin can be set individually
CSS properties for **dimensions** (4.3, 4.4.1)

<table>
<thead>
<tr>
<th>property</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>width</code>, <code>height</code></td>
<td>how wide or tall to make this element (block elements only)</td>
</tr>
<tr>
<td><code>max-width</code>, <code>max-height</code>, <code>min-width</code>, <code>min-height</code></td>
<td>max/min size of this element in given dimension</td>
</tr>
</tbody>
</table>

**An h2 heading**

<table>
<thead>
<tr>
<th>property</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>width</code></td>
<td>50%</td>
</tr>
<tr>
<td><code>background-color</code></td>
<td>aqua</td>
</tr>
</tbody>
</table>

This paragraph uses the first style above.

<table>
<thead>
<tr>
<th>property</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>margin-left</code></td>
<td>auto</td>
</tr>
<tr>
<td><code>margin-right</code></td>
<td>auto</td>
</tr>
<tr>
<td><code>width</code></td>
<td>750px</td>
</tr>
</tbody>
</table>

**Centering a block element: auto margins**

- works best if `width` is set (otherwise, may occupy entire width of page)
- to center inline elements within a block element, use `text-align: center;`