



中山大學

SUN YAT-SEN UNIVERSITY

Lecture 4

Page Sections and the CSS Box Model

SE-805 Web 2.0 Programming (supported by Google)

<http://my.ss.sysu.edu.cn/courses/web2.0/>

School of Software, Sun Yat-sen University

Outline

- **More CSS**
- Styling Page Sections
- Introduction to Layout

The HTML `id` Attribute

```
<p>Spatula City! Spatula City!</p> <p id="mission">Our mission is  
to provide the most spectacular spatulas and splurge on our specials  
until our customers <q>esplode</q> with splendor!</p>
```

HTML

Spatula City! Spatula City!

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

output

- 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

```
<p>Visit <a href=
"http://www.textpad.com/download/index.html#downloads">
textpad.com</a> to get the TextPad editor.</p> <p><a
href="#mission">View our Mission Statement</a></p>
```

HTML

Visit [textpad.com](http://www.textpad.com) to get the TextPad editor.
[View our Mission Statement](#)

output

- A link target can include an ID at the end, precede by a #
- Browser will load that page and scroll to element with given ID

CSS ID Selectors

```
#mission {  
  font-style: italic;  
  font-family: "Garamond", "Century Gothic", serif;  
}
```

HTML

Spatula City! Spatula City!

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

output

- Applies style only to the paragraph that has the ID of mission
- Element can be specified explicitly `p#mission { ... }`

The HTML **class** Attribute

```
<p class="shout">Spatula City! Spatula City!</p>  
<p class="special">See our spectacular spatula specials!</p>  
<p class="special">Today only: satisfaction guaranteed.</p>
```

HTML

Spatula City! Spatula City!
See our spectacular spatula specials!
Today only: satisfaction guaranteed.

output

- Classes are a way to group some elements and give a style to only that group
- Unlike an **id**, a **class** can be reused as much as you like on the page

CSS class Selectors

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

CSS

Spatula City! Spatula City!

See our spectacular spatula specials!

Today only: satisfaction guaranteed.

output

- Applies corresponding rule to any element with class **special** or a **p** with class **shout**

Multiple Classes

```
<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>
```

HTML

Spatula City! Spatula City!

See our spectacular spatula specials!

Satisfaction guaranteed.

We'll beat any advertised price!

output

- An element can be a member of multiple classes (separated by spaces)

CSS pseudo-classes

```
a:link      { color: #FF0000; }      /* unvisited link */
a:visited   { color: #00FF00; }      /* visited link */
a:hover     { color: #FF00FF; }      /* mouse over link */
```

CSS

Buy early, buy often!

output

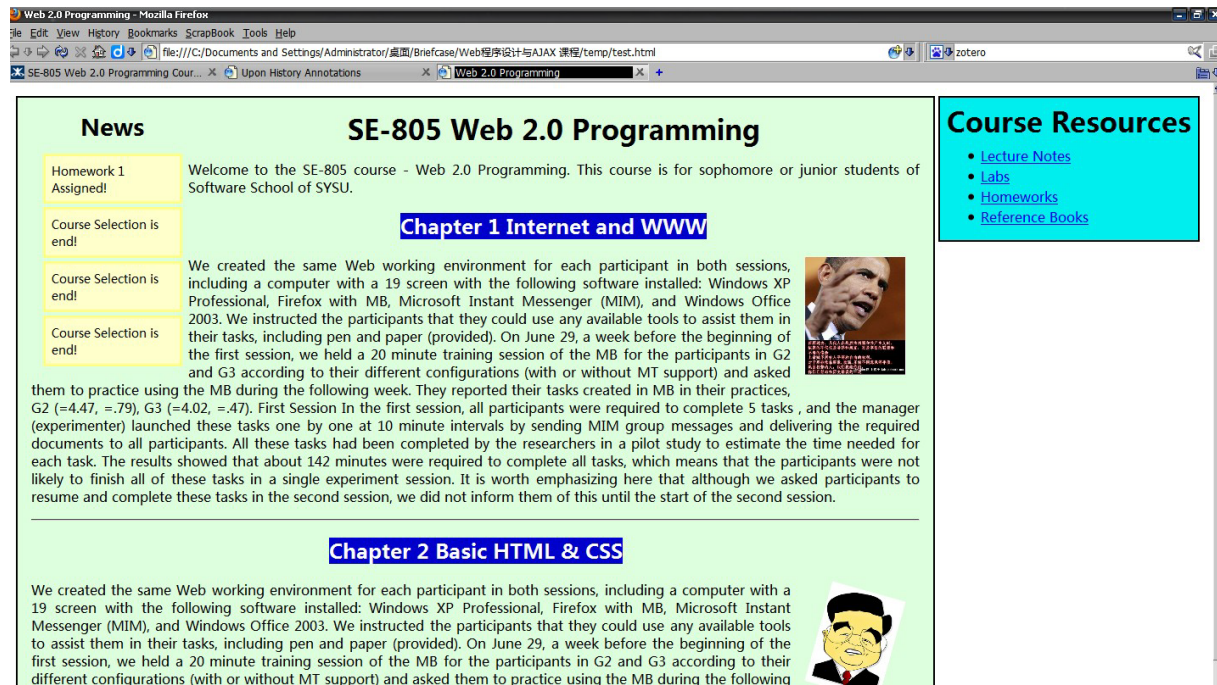
Class	Description
:active	an activated or selected element
:focus	an element that has the keyboard focus
:hover	an element that has the mouse over it
:link	a link that has not been visited
:visited	a link that has already been visited
:first-letter	the first letter of text inside an element
:first-line	the first line of text inside an element
:first-child	an element that is the first one to appear inside another

Outline

- More CSS
- **Styling Page Sections**
- Introduction to Layout

Page Segmentation

- style individual elements, groups of elements, and sections of text of the page differently
- create complex page layouts



Sections of a Page: `<div>`

a section or division of your HTML page (block)

```
<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>
```

HTML

Spatula City! Spatula City!

See our spectacular spatula specials!

We'll beat any advertised price!

output

- 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 Section: ``

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>
```

HTML

Spatula City! Spatula City!

See our **spectacular** spatula specials!

We'll beat **any advertised price!**

output

- Has no onscreen appearance, but you can apply a style or ID to it, which will be applied to the text inside the `span`
- So, when should we use `<div>`, ``, and when `<p>`, `<h1>`, etc.?

CSS Context Selectors

```
selector1 selector2 {  
  properties  
}
```

CSS

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

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

CSS

- Applies the given properties to *selector2* only if it is **directly** inside a *selector1* on the page (*selector2* tag is immediately inside *selector1* with no tags in between)

Context Selector Example

```
<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>
```

HTML

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

CSS

Shop at **Hardwick's Hardware...**

- The **best** prices in town!
- Act while supplies last!

output

More Complex Example

```
<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>
```

HTML

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

CSS

Shop at **Hardwick's Hardware...**

- The **best** prices in town!
- Act **while supplies last!**

output

CSS Cascade

- The browser's style sheet is the weakest.
- The user's style sheet takes precedence over the browser's style sheet.
- The author's style sheet is the strongest and takes precedence over the user's and the browser's style sheets.
- The (X)HTML style attribute is more important than styles defined in any style sheet.
- Within a style sheet, when conflict occurs, the most specific rule wins.

Specificity of Selector

- The specificity of a CSS selector is a four digits number likes **abcd**
- Count 1 if the styles are applied from the (X)HTML style attribute, and 0 otherwise; this becomes variable **a**.
- Count the number of ID attributes in the selector; the sum is variable **b**.
- Count the number of attributes, pseudo-classes, and class names in a selector; the sum is variable **c**.
- Count the number of element names in the selector; this is variable **d**.
- Ignore pseudo-elements.
- When two conflict rules has same specificity, the one occurs later is style sheet file wins
- At last, a rule with **!important** overrides precedence!

```
body {  
    font-size: 24px;  
}  
p {  
    background: lightblue !important;  
}  
p {  
    background: none;  
}
```

The !important rule takes precedence.

Specificity of Selector

Selector	Selector Type	Specificity
*	Universal Selector	0000 (a = 0, b = 0, c = 0, d = 0)
li	Element Name	0001 (a = 0, b = 0, c = 0, d = 1)
ul li	Element Name	0002 (a = 0, b = 0, c = 0, d = 2)
div h1 + p	Element Name	0003 (a = 0, b = 0, c = 0, d = 3)
input[type='text']	Element Name + Attribute	0011 (a = 0, b = 0, c = 1, d = 1)
.someclass	Class Name	0010 (a = 0, b = 0, c = 1, d = 0)
div.someclass	Element Name + Class Name	0011 (a = 0, b = 0, c = 1, d = 1)
div.someclass.someother	Element Name + Class Name + Class Name	0021 (a = 0, b = 0, c = 2, d = 1)
#someid	ID Name	0100 (a = 0, b = 1, c = 0, d = 0)
div#someid	Element Name + ID Name	0101 (a = 0, b = 1, c = 0, d = 1)
style (attribute)	style (attribute)	1000 (a = 1, b = 0, c = 0, d = 0)

CSS Inheritance

- Many properties in a CSS rule are inheritable by children elements of the rule specified, but some are not.
- Types of properties are inheritable: text, color, and font
- Types of properties are not: border, margin, padding
- All direct or inherited rules are same when considering their specificities
- If you can't remember whether a property is inheritable, better to figure it out by examining, other than via Googling or W3-Schooling

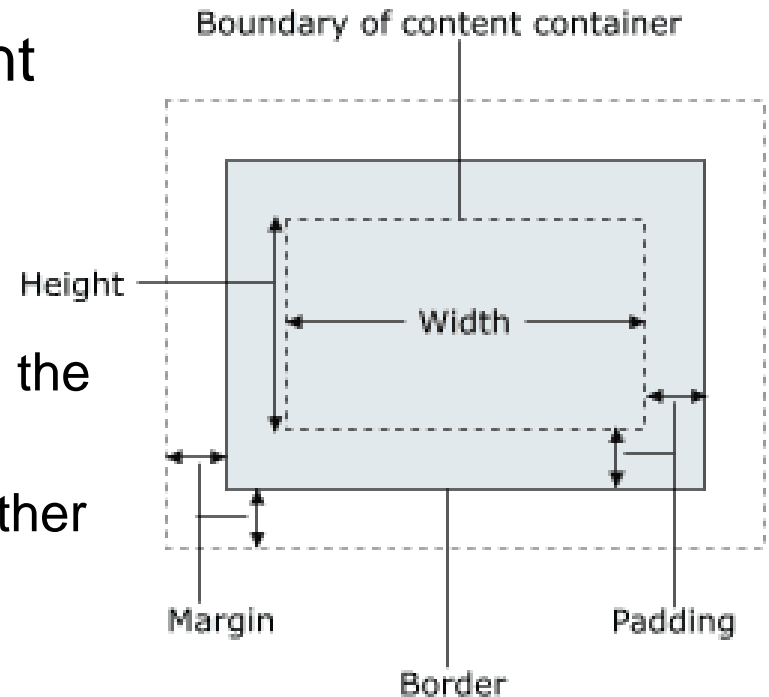
Outline

- More CSS
- Styling Page Sections
- **Introduction to Layout**

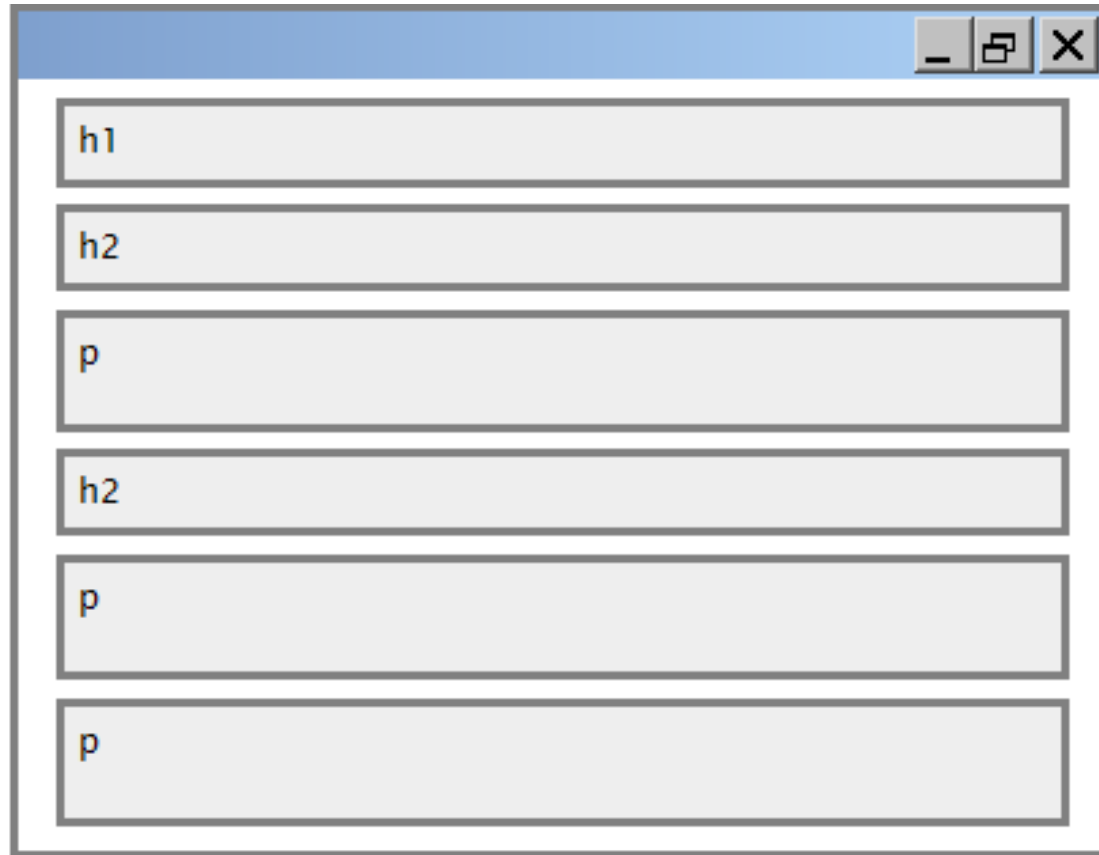
The CSS Box Model

- For layout purpose, every element is composed of:
 - The actual element's **content**
 - A **border** around the element
 - A **padding** between the content and the border (*inside*)
 - A **margin** between the border and other content (*outside*)
- $\text{width} = \text{content width} + \text{L/R padding} + \text{L/R margin}$

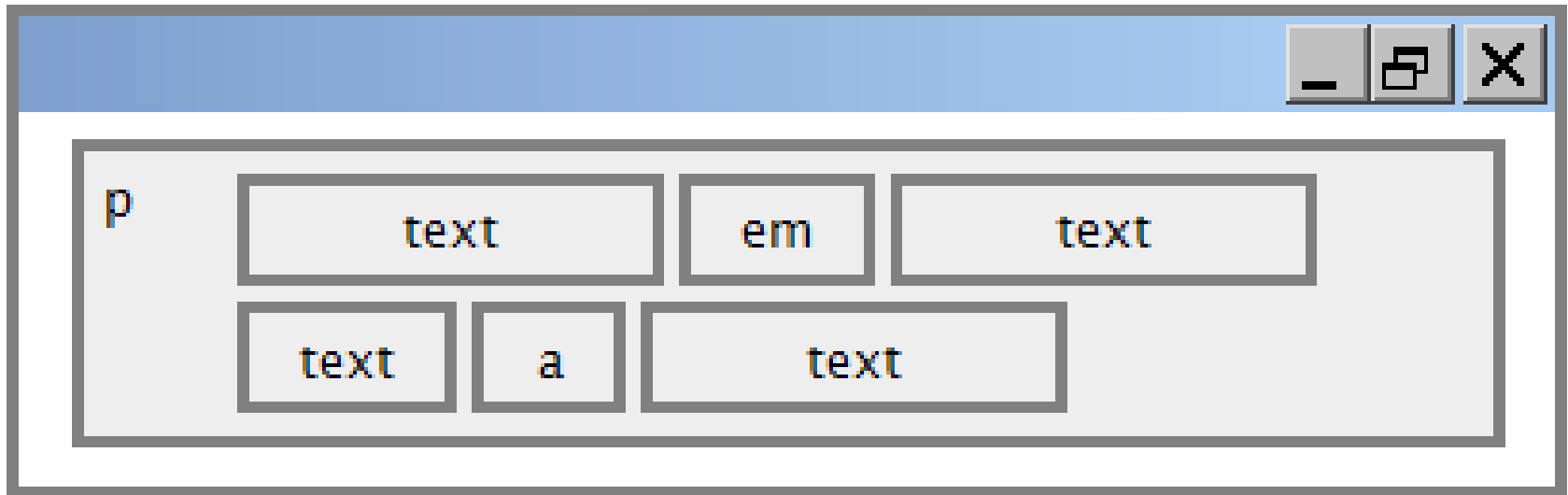
$\text{height} = \text{content height} + \text{T/B padding} + \text{T/B margin}$



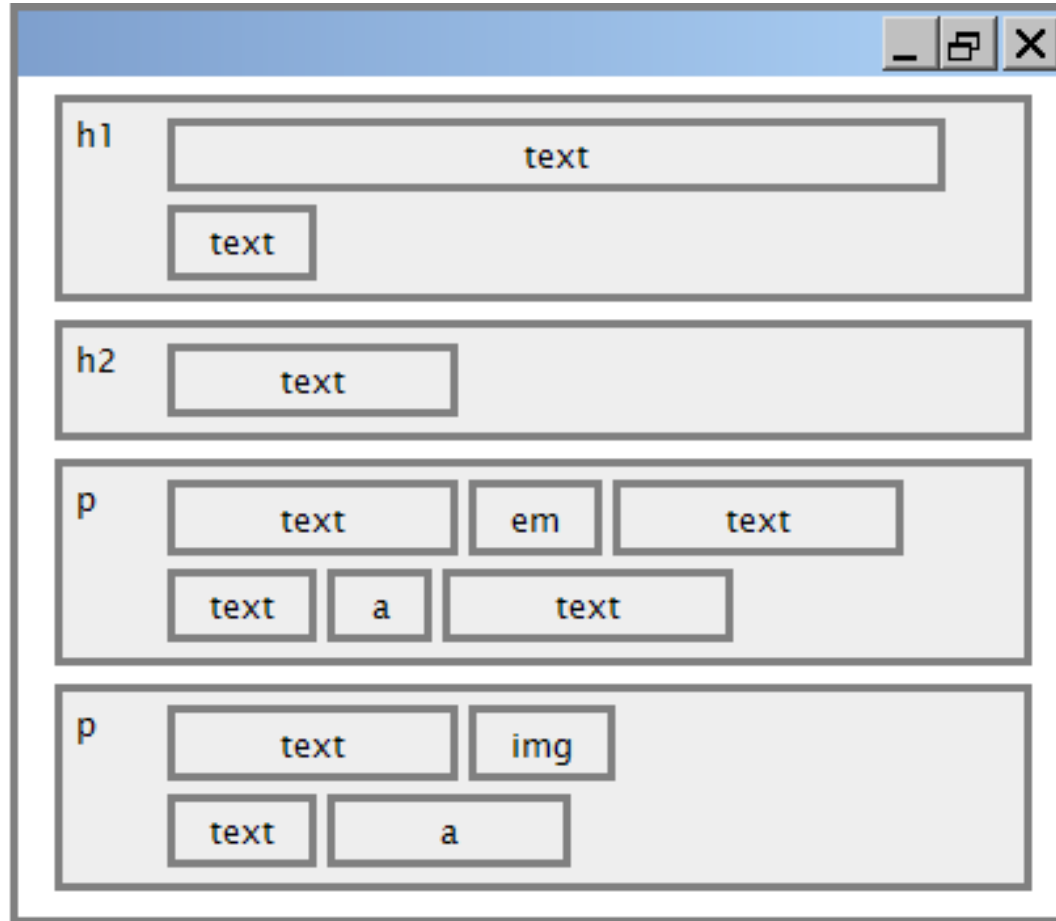
Document Flow – Block Elements



Document Flow – Inline Elements



Document Flow – a Larger Example



CSS Properties for Borders

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

CSS

This is a heading.

output

property	description
<u>border</u>	thickness/style/size of border on all 4 sides

- **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

Property	Description
<u>border-color</u> , <u>border-width</u> , <u>border-style</u>	specific properties of border on all 4 sides
<u>border-bottom</u> , <u>border-left</u> , <u>border-right</u> , <u>border-top</u>	all properties of border on a particular side
<u>border-bottom-color</u> , <u>border-bottom-style</u> , <u>border-bottom-width</u> , <u>border-left-color</u> , <u>border-left-style</u> , <u>border-left-width</u> , <u>border-right-color</u> , <u>border-right-style</u> , <u>border-right-width</u> , <u>border-top-color</u> , <u>border-top-style</u> , <u>border-top-width</u>	properties of border on a particular side

[Complete list of border properties](#)

Border Example 2

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

CSS

This is a heading.

output

- 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

Property	Description
<u>padding</u>	padding on all 4 sides
<u>padding-bottom</u>	padding on bottom side only
<u>padding-left</u>	padding on left side only
<u>padding-right</u>	padding on right side only
<u>padding-top</u>	padding on top side only
<u>Complete list of padding properties</u>	

Padding Example 1

```
p { padding: 20px; border: 3px solid black; }  
h2 { padding: 0px; background-color: yellow; }
```

CSS

This is the first paragraph

This is the second paragraph

This is a heading

Padding Example 2

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

CSS

This is the first paragraph

This is the second paragraph

Output

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

CSS Properties for Margins

property	description
<u>margin</u>	margin on all 4 sides
<u>margin-bottom</u>	margin on bottom side only
<u>margin-left</u>	margin on left side only
<u>margin-right</u>	margin on right side only
<u>margin-top</u>	margin on top side only
<u>Complete list of margin properties</u>	

Margin Example 1

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

CSS

This is the first paragraph

This is the second paragraph

output

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

Margin Example 2

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

CSS

This is the first paragraph

This is the second paragraph

Output

- Each side's margin can be set individually

CSS Properties for Dimensions

```
p { width: 350px; background-color: yellow; }
h2 { width: 50%; background-color: aqua; }
```

CSS

This paragraph uses the first style above.

An h2 heading

output

Property	Description
<u>width</u> , <u>height</u>	how wide or tall to make this element (block elements only)
<u>max-width</u> , <u>max-height</u> , <u>min-width</u> , <u>min-height</u>	max/min size of this element in given dimension

Centering a Block Element: **auto** Margins

```
p {  
  margin-left: auto;  
  margin-right: auto;  
  width: 750px;  
}
```

CSS

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

output

- 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**

Summary

- More CSS
 - HTML attributes: id, class
 - Multiple classes
 - Pseudo-classes
- Styling Page Sections
 - Page section motivation
 - div, span
 - CSS context selector
 - CSS Cascading & Inheritance
- Introduction to Layout
 - Box Model, document flow
 - Properties for borders, paddings, margins
 - Properties for dimensions

Exercises

- Complete the example of this class
 - the initial files can be downloaded from [lecture example](#)
 - the result Web page looks like this:



Further Readings

- W3C CSS2 Specification: <http://www.w3.org/TR/REC-CSS2/>
- W3Schools CSS2 Reference: http://www.w3schools.com/css/css_reference.asp
- W3Schools CSS Tutorial: <http://www.w3schools.com/css/default.asp>
- Chapter 3, 4, 7, 8, and 11 of Beginning CSS Cascading Style Sheets for Web Design, second edition
- <http://www.barelyfitz.com/screencast/html-training/css/positioning/>
- <http://www.quirksmode.org/css/display.html>
- http://en.wikipedia.org/wiki/User-centered_design
- <http://www.stcsig.org/usability/newsletter/9807-webguide.html>

Thank you!

