



中山大學
SUN YAT-SEN UNIVERSITY

Lecture 4

Page Sections and the CSS Box Model

SE-805 Web 2.0 Programming

(<http://my.ss.sysu.edu.cn/wiki/display/W2PSC/Home> , supported by Google;
using some slides of & inspired by Marty Stepp's CSE 190 M courseware)

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>explode</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

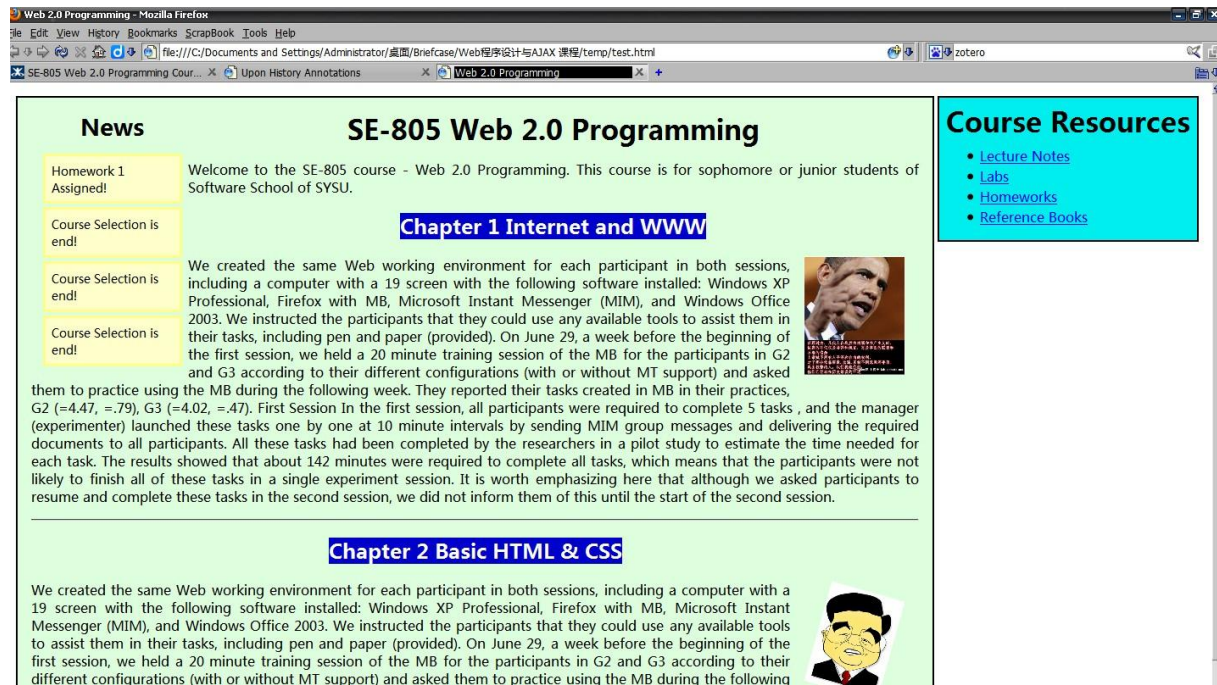
class	description
:active	an activated or selected element
:focus	an element that has the keyboard focus
:hover	an element that has the mouse over it
:link	a link that has not been visited
:visited	a link that has already been visited
:first-letter	the first letter of text inside an element
:first-line	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

Motivation for page sections

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

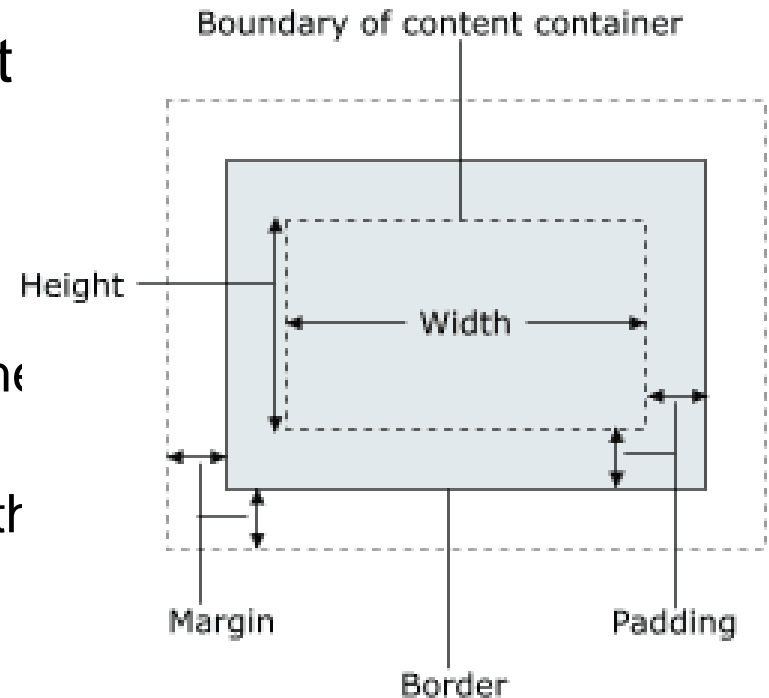
Outline

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

The CSS Box Model

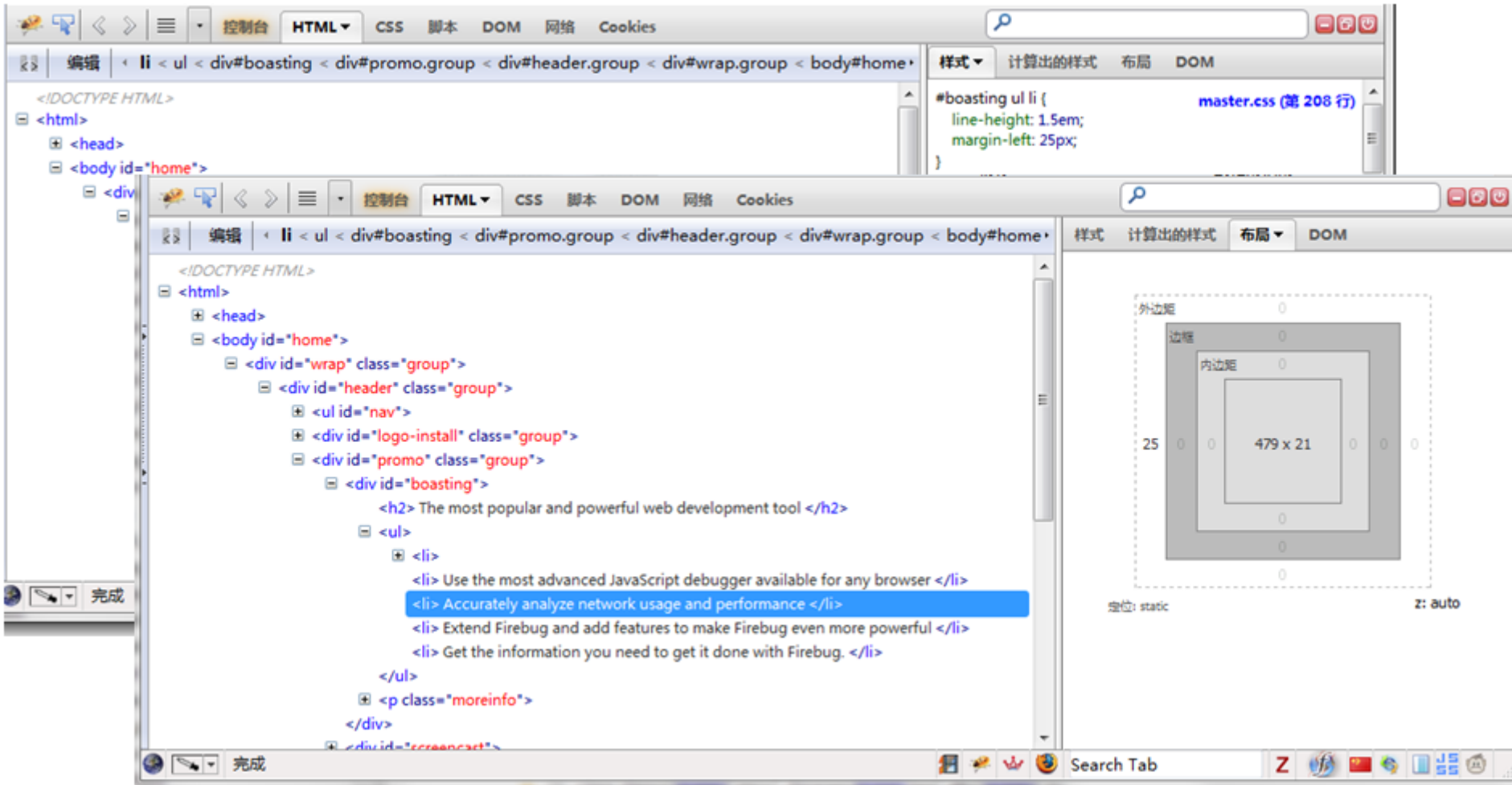
- for layout purpose, every element composed of:
 - the actual element's **content**
 - a **border** around the element
 - padding** between the content and the border (*inside*)
 - a **margin** between the border and other content (*outside*)
- Width amount = width + L/R padding + L/R border + L/R margin

height amount = content height + T/B padding + T/B border + T/B margin



Debug Boxes of Web Page Elements

- Firefox



Debug Boxes of Web Page Elements

● Chrome

The screenshot displays the Chrome DevTools interface. The **Elements** panel on the left shows the DOM tree. The **div#frs_nav** element is selected, which is a navigation bar. The **Styles** panel on the right shows the CSS rules for the selected element, including 'word-break: break-all;'. The **Metrics** panel shows a box model diagram with dimensions: margin (10), border (10), padding (10), and content (977.272705078125 x 34.54545211791992 x 2.7272727489471436).

Debug Boxes of Web Page Elements

IE

文件(F) 查找(N) 禁用(S) 查看(V) 图像(I) 缓存(C) 工具(T) 验证(A) 浏览器模式(B): IE9 文档模式: IE9 标准(M)

HTML CSS 控制台 脚本 探查器 网络 搜索 HTML...

样式 跟踪样式 布局 属性

```

<img width="276" height="110" title="Google" id="logo" style="padding-top: 28px;" onload="windo
<br/>
<form name="f" action="/search">
  <table class="jhp" cellSpacing="0" cellPadding="0">
    <tbody>
      <tr vAlign="top">
        <td align="center" nowrap="">
          <input name="client" type="hidden" value="aff-ime"/>
          <input name="hl" type="hidden" value="zh-CN"/>
          <input name="source" type="hidden" value="hp"/>
          <input name="biw" type="hidden" value="784"/>
          <input name="bih" type="hidden" value="448"/>
          <div class="ds" style="height: 32px; margin-top: 4px; margin-right: 0px; margin-bo
          <input name="q" title="Google" class="lst" id="lst-ib" style="color: #000; pad
          <br style="line-height: 0;"/>
          <input name="btnG" class="lsb" onclick="this.checked=1" type="submit" value="Googl
          <input name="btnI" style="margin-left: 10px;" onclick="this.checked=1" type="submi
        <input name="oq" type="hidden"/>
        <input name="aq" type="hidden"/>
        <input name="aqi" type="hidden"/>
        <input name="aql" type="hidden"/>
        <input name="gs_sm" type="hidden"/>
        <input name="gs_upl" type="hidden"/>
      <div style="font-size: 83%; min-height: 3.5em;">
        <br/>
        <div id="als">
          <font id="addlang" size="-1">
            <br/>
            <br/>
          </font>
        </div>
      </div>
    </tbody>
  </table>
</form>

```

Offset 52

Margin 11px

Border 1px

Padding 0px

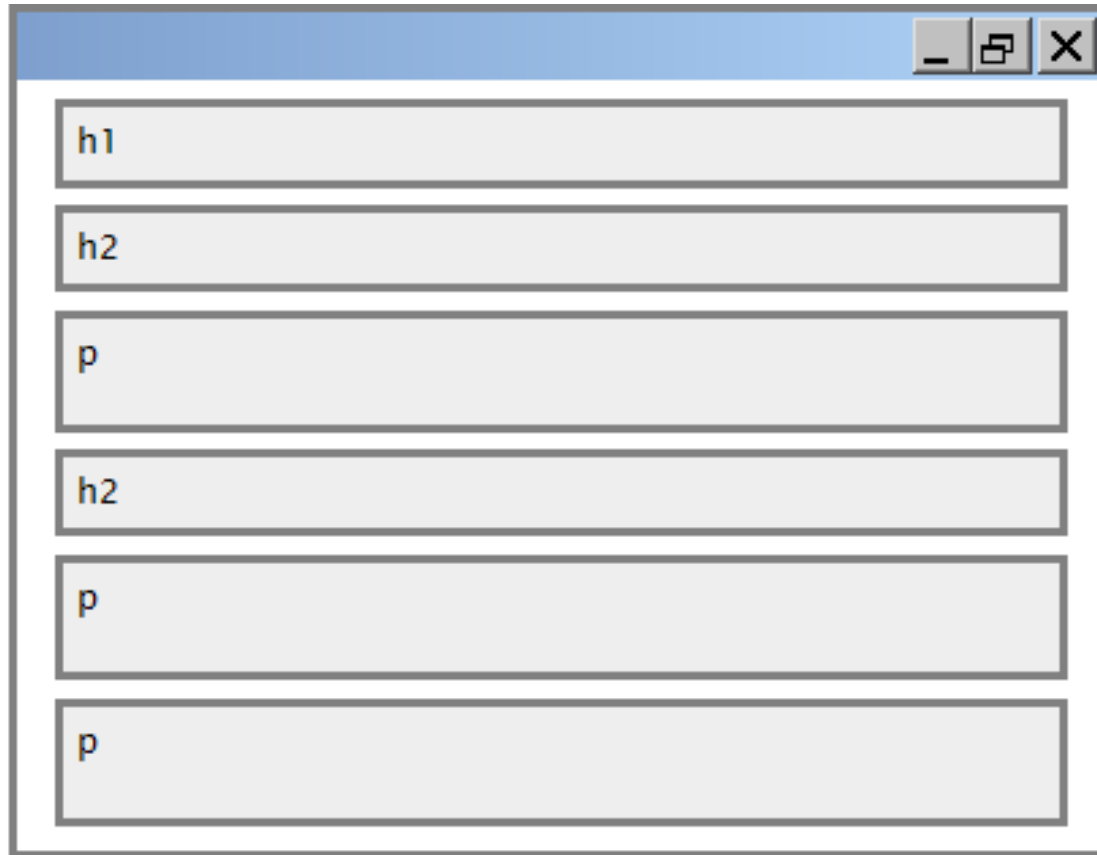
63.38px x 27px

172 6px 1px 8px 8px 1px 6px

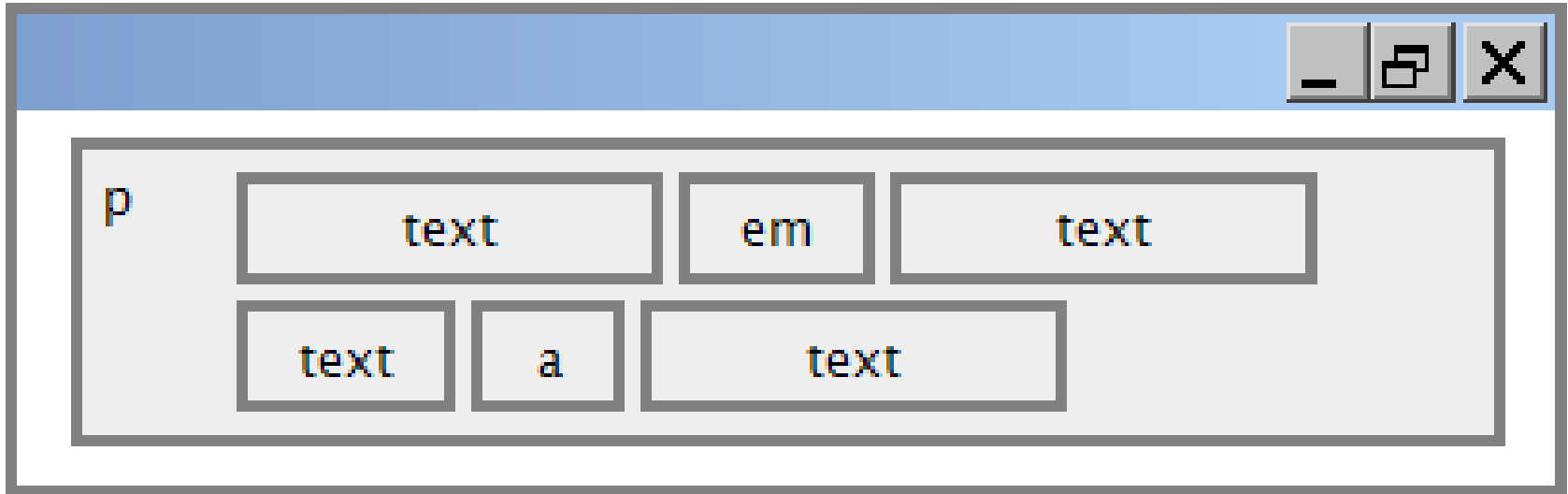
112 0px 1px 11px

坐标: (0, 0) Z-index = auto

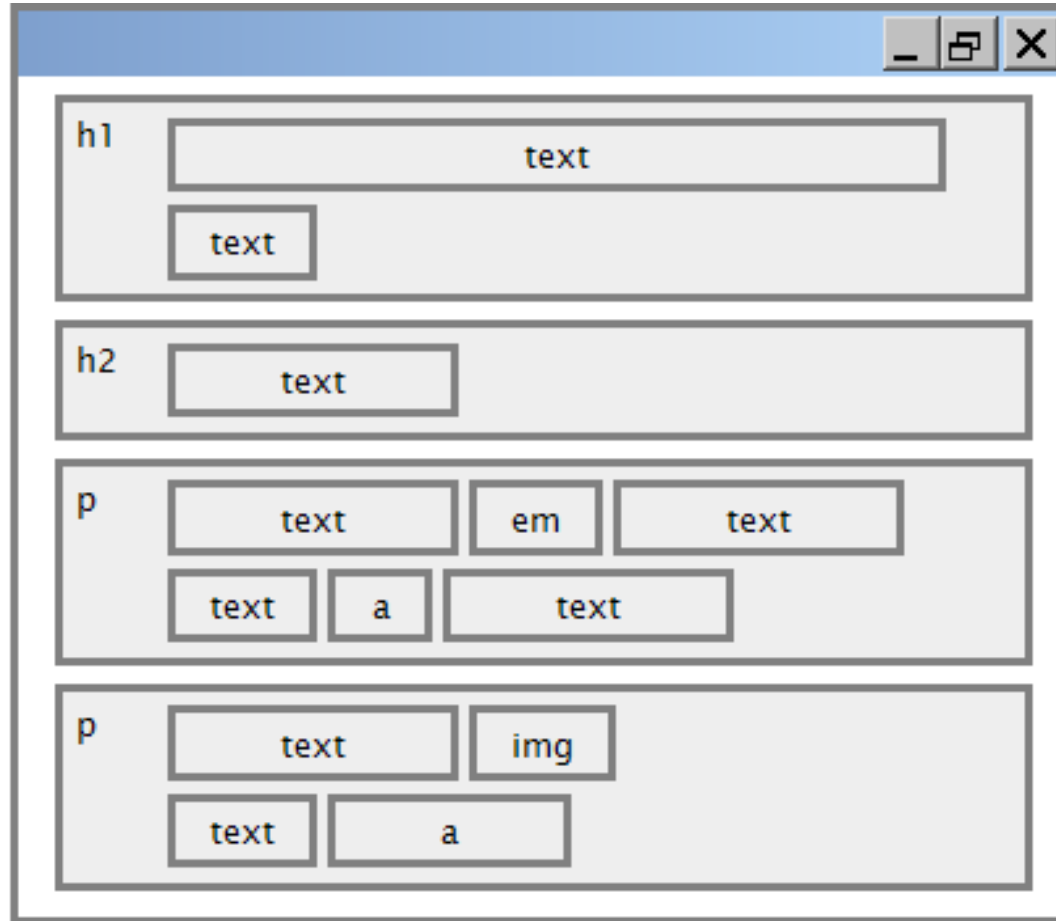
Document flow – block elements



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

CSS

- 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**

background-image

- Multiple background images

源代码 3-14 background-image 属性（多图片）示例

```
<td style='background-image:url(images/bird.png), url(images/泰山.jpg);'>多个图片背景 </td>
```



图 3-11 源代码 3-14 运行效果（左：运行效果；中：bird.png；右：泰山.jpg）

background-size

源代码 3-14 background-image 属性（多图片）示例

```
<td style='background-image:url(images/bird.png), url(images/泰山.jpg);'>多个图片背景 </td>
```



图 3-11 源代码 3-14 运行效果（左：运行效果；中：bird.png；右：泰山.jpg）

background-position

源代码 3-16 background-position 属性示例

```
<td style='background-position: right top;'>右顶部<br/> (right top) </td>  
<td style='background-position: bottom;'>底部 (bottom) </td>  
<td style='background-position: 25px 5px;'>距左上角<br/> (25, 5) 像素</td>  
<td style='background-position: 50% 50%;'>50% 50%<br/>等价于center</td>
```



图 3-13 源代码 3-16 运行效果

background-origin

源代码 3-17 background-origin 属性示例

```

0  <style>
1      td{..... border: solid 10px rgba(255, 0, 0, 0.2); padding: 10px;}
2  </style>
3
4  <td>默认origin </td>
5  <td style='background-origin: padding-box'>padding-box </td>
6  <td style='background-origin: border-box'>border-box </td>
7  <td style='background-origin: content-box'>content-box </td>

```



图 3-15 源代码 3-17 运行效果

background-clip

■ 源代码 3-18 background-clip 属性示例

```
<style> ↵
  td{ ↵
    padding: 10px; border: solid 10px rgba(255, 0, 0, 0.2); ↵
    background-repeat:no-repeat;..... ↵
  } ↵
</style> ↵
<td>默认clip </td> ↵
<td style='background-clip: padding-box'>padding-box </td> ↵
<td style='background-clip: border-box'>border-box </td> ↵
<td style='background-clip: content-box'>content-box </td> ↵
```



图 3-16 源代码 3-18 运行效果

background-repeat

■ 源代码 3-19 background-repeat 属性示例

<td>从左上角开始
同时横向、纵向重复</td>

<td style='background-repeat:repeat-y;'>从左上角开始
纵向重复</td>

<td style='background-position: bottom; background-repeat:repeat-x;'>
从底部中间开始
横向重复</td>

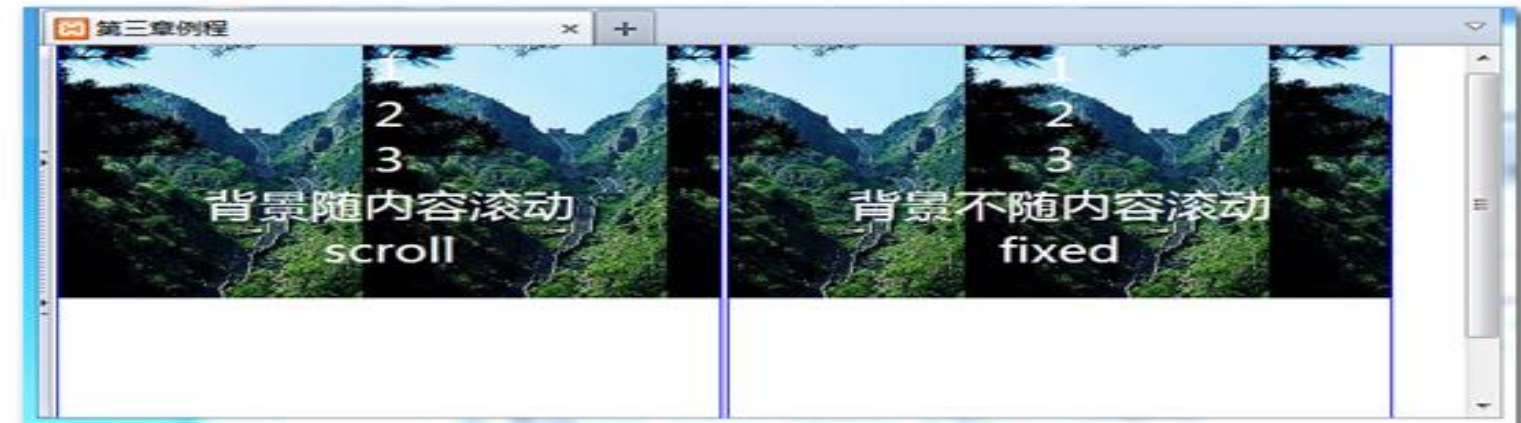


图 3-17 源代码 3-19 运行效果

background-attachment

源代码 3-20 background-attachment 属性示例

```
<style>
td { background-repeat: repeat-x;
.....
}
</style>
<td>背景随内容滚动<br/>scroll</td>
<td style='background-attachment: fixed;'>背景不随内容滚动<br/>fixed</td>
```



Background all in one

源代码 3-21 background 属性示例

```
<td style='background-color: yellow; background-position: bottom  
right; background-repeat: no-repeat; background-image:  
url(images/泰山.jpg);'>分别表述</td>  
<td style='background: no-repeat url(images/泰山.jpg) bottom right  
yellow'>综合表述</td>
```

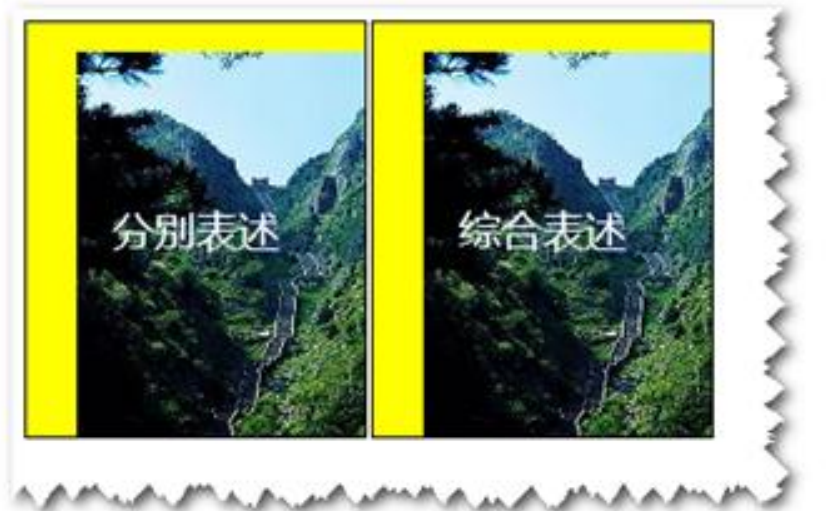


图 3-19 源代码 3-21 运行效果

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 demensions

Exercises

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

Web 2.0 Programming - Mozilla Firefox

file:///C:/Documents and Settings/Administrator/桌面/Briefcase/Web程序设计/AJAX 课程/temp/test.html

SE-805 Web 2.0 Programming Cour... Upon History Annotations Web 2.0 Programming zotero

News

SE-805 Web 2.0 Programming

Homework 1 Assigned! Welcome to the SE-805 course - Web 2.0 Programming. This course is for sophomore or junior students of Software School of SYSU.

Course Selection is end!

Course Selection is end!

Course Selection is end!

Chapter 1 Internet and WWW

We created the same Web working environment for each participant in both sessions, including a computer with a 19 screen with the following software installed: Windows XP Professional, Firefox with MB, Microsoft Instant Messenger (MIM), and Windows Office 2003. We instructed the participants that they could use any available tools to assist them in their tasks, including pen and paper (provided). On June 29, a week before the beginning of the first session, we held a 20 minute training session of the MB for the participants in G2 and G3 according to their different configurations (with or without MT support) and asked them to practice using the MB during the following week. They reported their tasks created in MB in their practices, G2 (=4.47, =.79), G3 (=4.02, =.47). First Session In the first session, all participants were required to complete 5 tasks, and the manager (experimenter) launched these tasks one by one at 10 minute intervals by sending MIM group messages and delivering the required documents to all participants. All these tasks had been completed by the researchers in a pilot study to estimate the time needed for each task. The results showed that about 142 minutes were required to complete all tasks, which means that the participants were not likely to finish all of these tasks in a single experiment session. It is worth emphasizing here that although we asked participants to resume and complete these tasks in the second session, we did not inform them of this until the start of the second session.

Chapter 2 Basic HTML & CSS

We created the same Web working environment for each participant in both sessions, including a computer with a 19 screen with the following software installed: Windows XP Professional, Firefox with MB, Microsoft Instant Messenger (MIM), and Windows Office 2003. We instructed the participants that they could use any available tools to assist them in their tasks, including pen and paper (provided). On June 29, a week before the beginning of the first session, we held a 20 minute training session of the MB for the participants in G2 and G3 according to their different configurations (with or without MT support) and asked them to practice using the MB during the following

Course Resources

- [Lecture Notes](#)
- [Labs](#)
- [Homeworks](#)
- [Reference Books](#)

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!

