

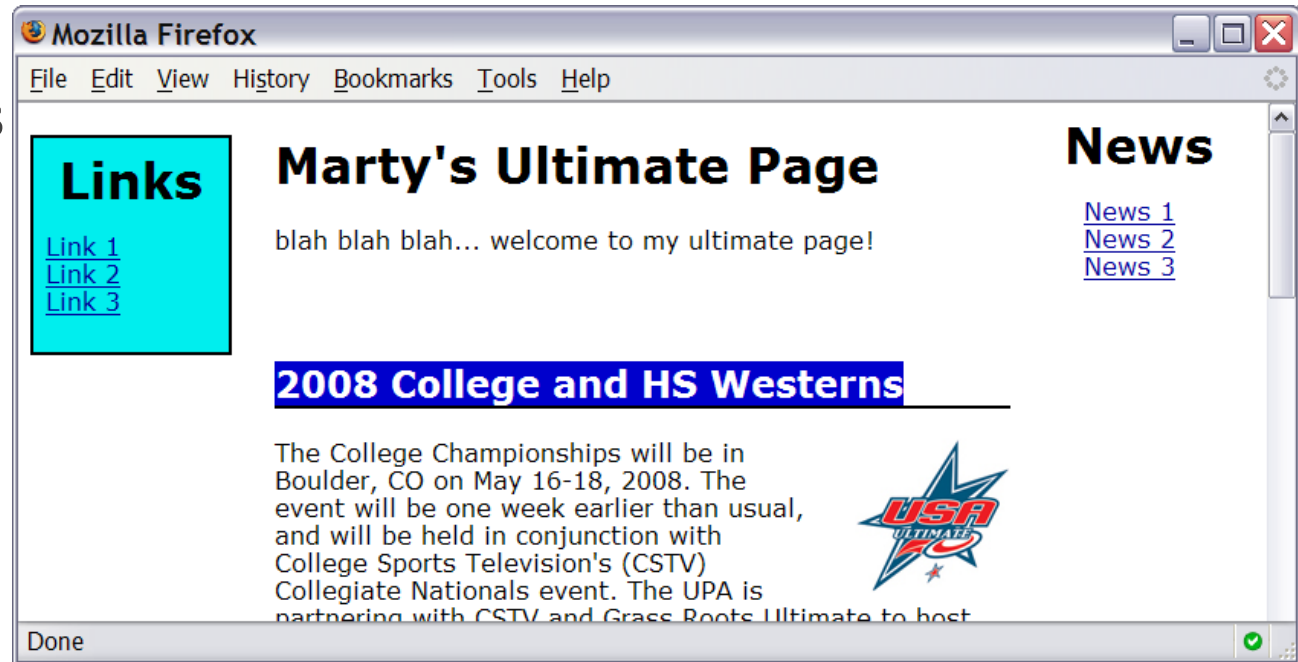
# CSE 154

---

## LECTURE 3: PAGE SECTIONS AND THE CSS BOX MODEL

# 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



# 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 “esplode” 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, preceded 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;  
}
```

CSS

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 (“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

```
.special { /* any element with class="special" */
  font-weight: bold;
}
p.shout { /* only p elements with class="shout" */
  color: red;
  font-family: cursive;
}
```

CSS

**Spatula City! Spatula City!**

**See our spectacular spatula specials!**

**Today only: satisfaction guaranteed.**

output

- applies 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>
```

**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)



# CSS for following examples

---

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

CSS

- for the next several slides, assume that the above CSS rules are defined

# 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 sections: <span>

---

*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`

# 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!</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...**

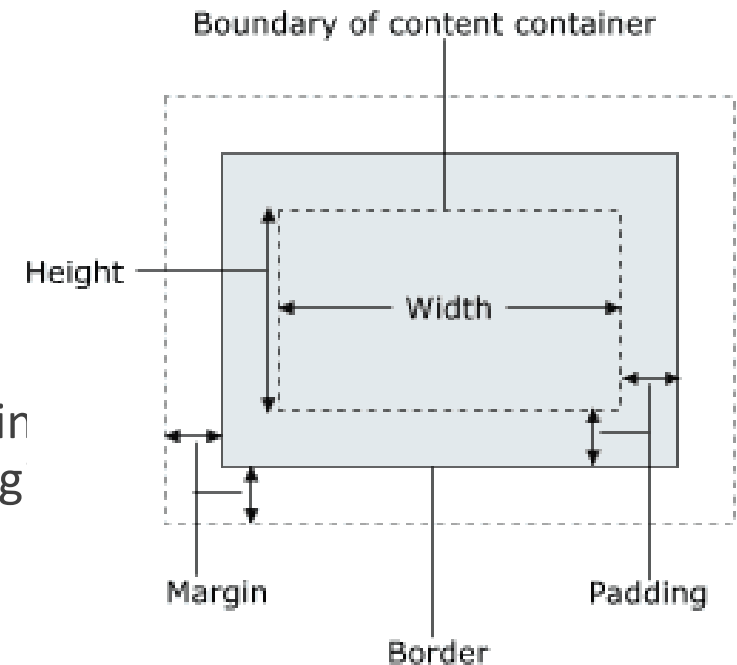
•The **best** prices!

•Act **while supplies last!**

output

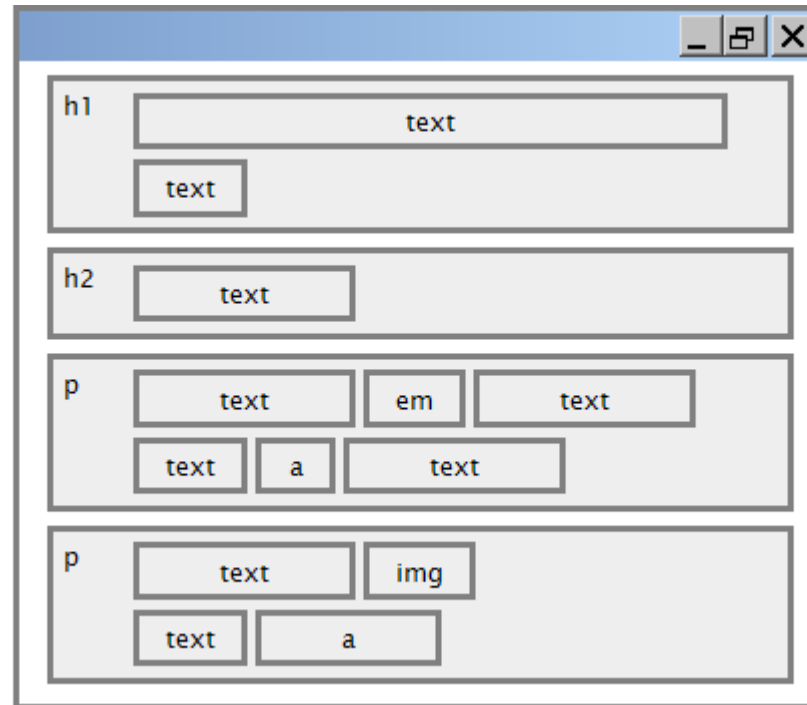
# The CSS Box Model

- 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 marg
  - IE6 doesn't do this right



# Document flow - block and inline elements

---





# CSS properties for borders

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

CSS

**This is a heading.**

output

property	description
<a href="#">border</a>	thickness/style/color 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
<a href="#">border-color</a> , <a href="#">border-width</a> , <a href="#">border-style</a>	specific properties of border on all 4 sides
<a href="#">border-bottom</a> , <a href="#">border-left</a> , <a href="#">border-right</a> , <a href="#">border-top</a>	all properties of border on a particular side
<a href="#">border-bottom-color</a> , <a href="#">border-bottom-style</a> , <a href="#">border-bottom-width</a> , <a href="#">border-left-color</a> , <a href="#">border-left-style</a> , <a href="#">border-left-width</a> , <a href="#">border-right-color</a> , <a href="#">border-right-style</a> , <a href="#">border-right-width</a> , <a href="#">border-top-color</a> , <a href="#">border-top-style</a> , <a href="#">border-top-width</a>	properties of border on a particular side
<a href="#">Complete list of border properties</a>	

# 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)

# Rounded corners with border-radius

```
p {  
  border: 3px solid blue;  
  border-radius: 12px;  
  padding: 0.5em;  
}
```

CSS

This is a paragraph.

This is another paragraph.  
It spans multiple lines.

output

- each side's border radius can be set individually, separated by spaces

# CSS properties for padding

---

<b>property</b>	<b>description</b>
<a href="#"><u>padding</u></a>	padding on all 4 sides
<a href="#"><u>padding-bottom</u></a>	padding on bottom side only
<a href="#"><u>padding-left</u></a>	padding on left side only
<a href="#"><u>padding-right</u></a>	padding on right side only
<a href="#"><u>padding-top</u></a>	padding on top side only
<a href="#"><u>Complete list of padding properties</u></a>	

# 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

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

# CSS properties for margins

---

<b>property</b>	<b>description</b>
<a href="#"><u>margin</u></a>	margin on all 4 sides
<a href="#"><u>margin-bottom</u></a>	margin on bottom side only
<a href="#"><u>margin-left</u></a>	margin on left side only
<a href="#"><u>margin-right</u></a>	margin on right side only
<a href="#"><u>margin-top</u></a>	margin on top side only
<a href="#"><u>Complete list of margin properties</u></a>	



# Margin example 1

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

CSS

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

```
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

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