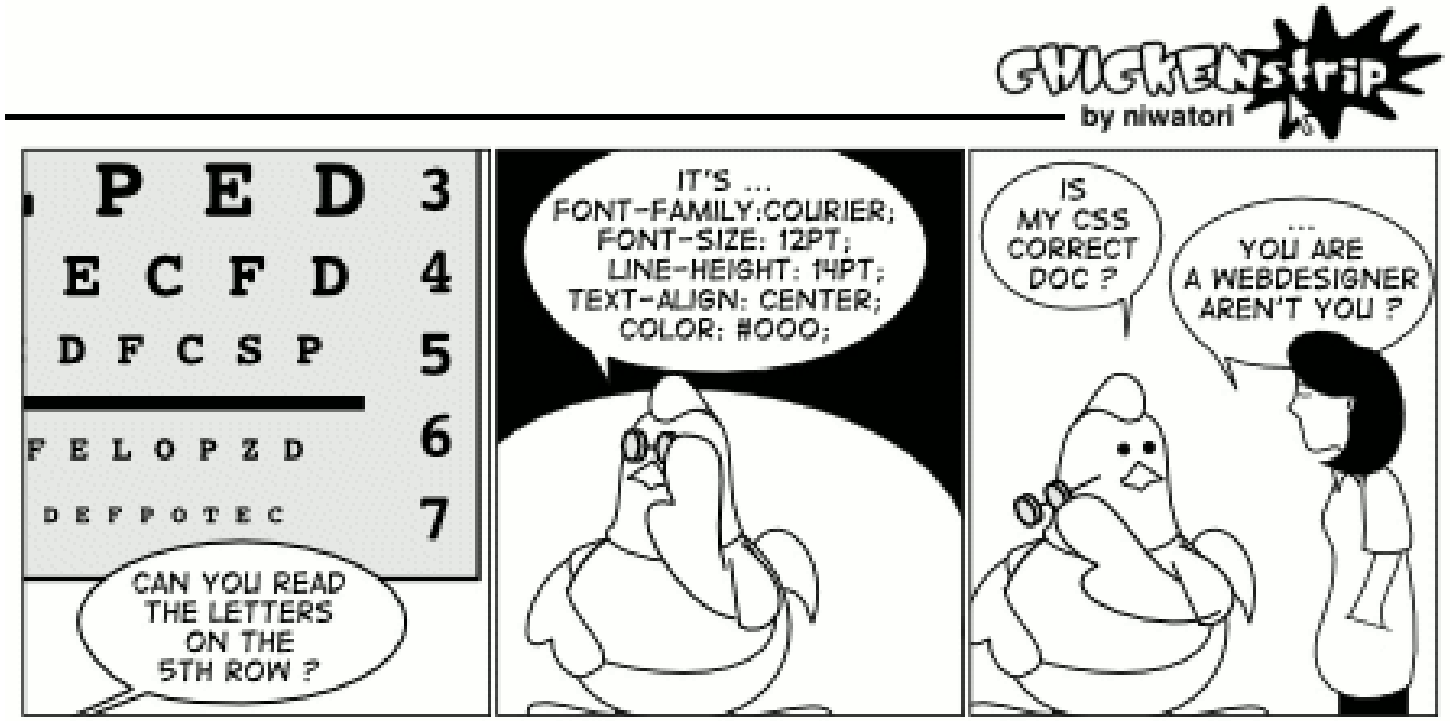
CSE 154

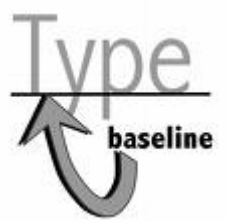


LECTURE 4: PAGE SECTIONS AND THE CSS BOX MODEL

The vertical-align property

property	description
vertical-align	specifies where an inline element should be aligned vertically, with respect to other content on the same line within its block element's box

- can be `top`, `middle`, `bottom`, `baseline` (default), `sub`, `super`, `text-top`, `text-bottom`, or a length value or `%`
  - `baseline` means aligned with bottom of non-hanging letters



Vertical Align

```
img { vertical-align: bottom }
```

beginning text 

end text

```
img { vertical-align: middle }
```

beginning text 

end text

```
img { vertical-align: top }
```

beginning text 

end text

The HTML id attribute

```HTML
<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>
```

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 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

```css
#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 "esplode" 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

```html
<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

```css
.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

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

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

Sections of a page:

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

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

```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>
```

HTML

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

CSS

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

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

ouput

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!</li>
    <li>Act <strong>while supplies last!</strong></li>
  </ul>
</div>
```

HTML

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

CSS

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

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

output

Inline sections:

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!

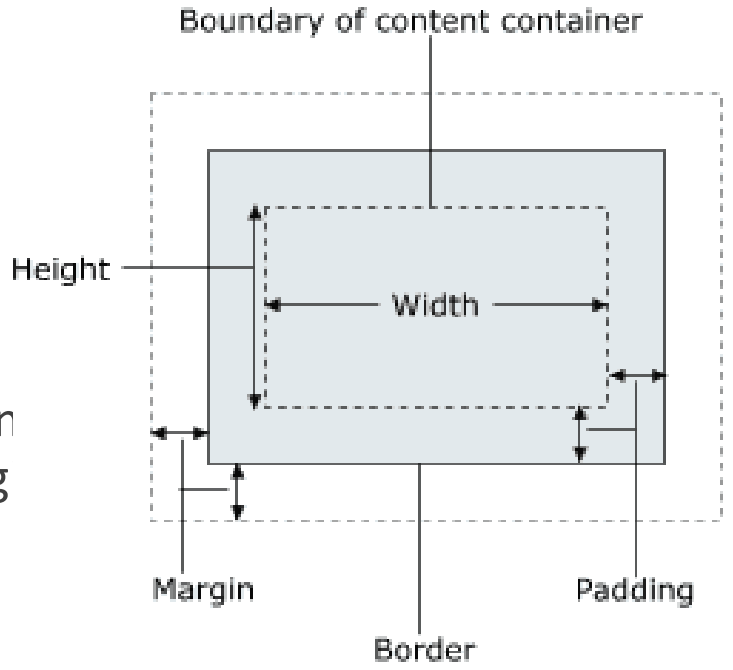
We'll beat any advertised price!

output

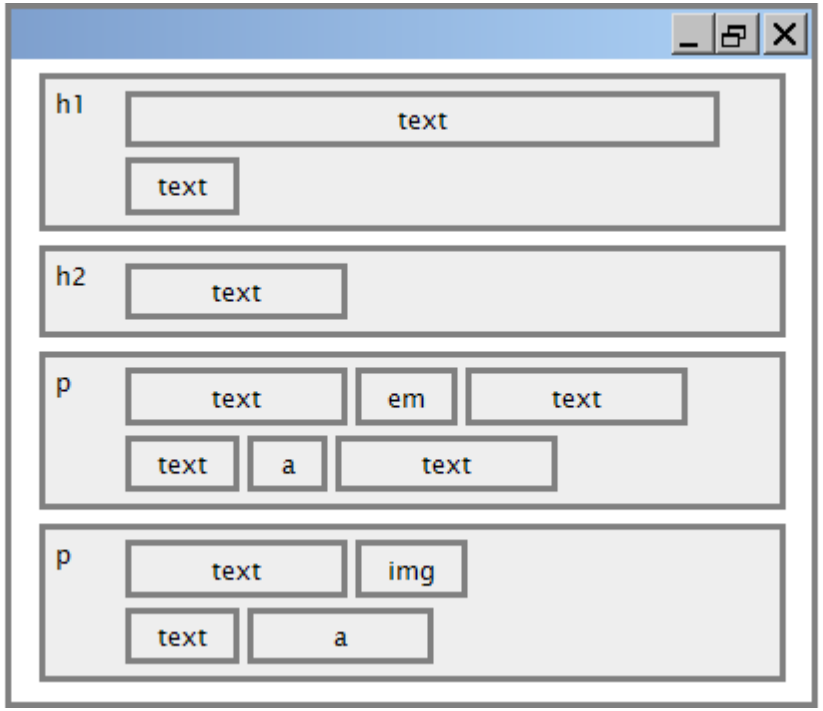
- has no onscreen appearance, but you can apply a style or ID to it, which will be applied to the text inside the `span`

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
border	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
border-color, border-width, border-style	specific properties of border on all 4 sides
border-bottom, border-left, border-right, border-top	all properties of border on a particular side
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	properties of border on a particular side
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;
}
```

CSS

This is a heading.

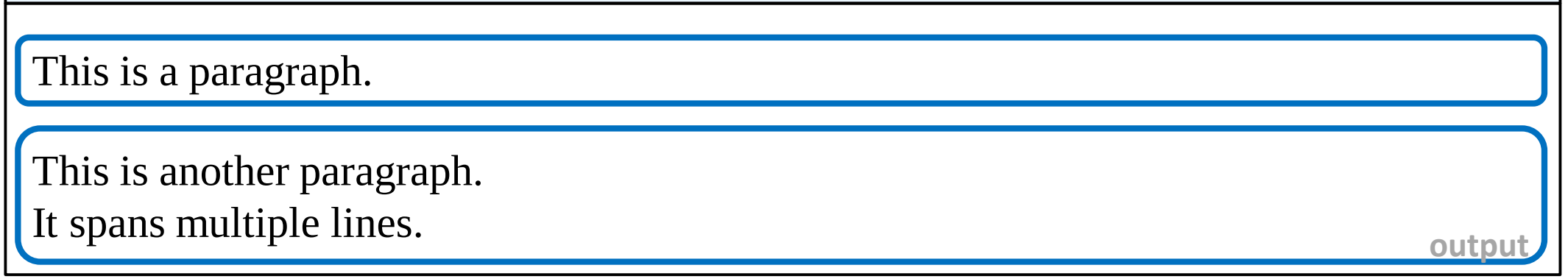
output

- 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

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

CSS



output

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

CSS properties for padding

property	description
padding	padding on all 4 sides
padding-bottom	padding on bottom side only
padding-left	padding on left side only
padding-right	padding on right side only
padding-top	padding on top side only
Complete list of padding properties	

CSS properties for margins

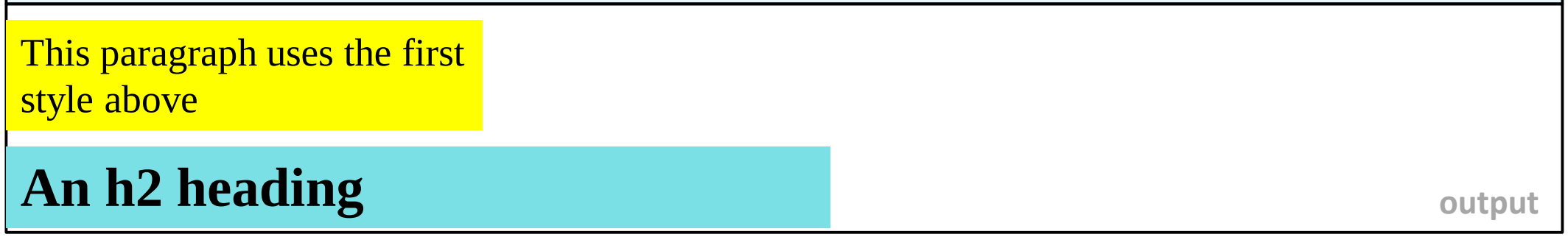
property	description
margin	margin on all 4 sides
margin-bottom	margin on bottom side only
margin-left	margin on left side only
margin-right	margin on right side only
margin-top	margin on top side only
Complete list of margin properties	

CSS properties for dimensions

```
p { width: 350px; background-color: yellow; }

h2 { width: 50%; background-color: aqua; }
```

CSS



output

property	description
width, height	how wide or tall to make this element (block elements only)
max-width, max-height, min-width, min-height	max/min size of this element in given dimension

Centering a block element: auto margins

```css
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)