

# Introduction To Web Programming I

(CSC 211)

Lecture Note 6

Box Model (*cont...*)

31<sup>st</sup> January, 2020

# Borders and Background

- ▶ Now that you know how to size and space elements using the box model, you are ready to learn how to apply other formatting to boxes. That includes adding borders and setting background colors and images.
- ▶ **Properties for setting borders**
- ▶ **border** - A border **width**, border **style**, and border **color**. The values are applied to all sides of the border.
- ▶ **border-side** - Border width, style and color values for the specified side of a border.
- ▶ **border-width** - One to four relative or absolute values (excluding a percent) or keywords that specify the *widths* for each side of a border. Possible keywords are *thin*, *medium*, and *thick*.
- ▶ **border-style** - One to four keywords that specify styles for each side of a border. Possible values are *dotted*, *dashed*, *solid*, *double*, *groove*, *ridge*, *inset*, *outset*, and *none*. The default is none.

- ▶ **border-color** - One to four color values or keywords that specify the color for each side of a border. The default is the color for the element.
- ▶ **border-side-width** - A relative or absolute value (excluding a percent) or a keyword that specifies the width of the indicated side of a border.
- ▶ **border-side-style** - A keyword that specifies the style for the indicated side of a border.
- ▶ **border-side-color** - A color value or keyword that specifies the color of the indicated side of a border.
- ▶ **The syntax for the shorthand border and border-side properties**
  - ❖ **border:** [width] [style] [color];
  - ❖ **border-side:** [width] [style] [color];
- ▶ **How to set border properties**
  - ❖ **border:** thin solid green;
  - ❖ **border:** 2px dashed #808080;
  - ❖ **border:** 1px inset; /\* uses the element's color property \*/

► **How to set side borders**

- ❖ `border-top: 2px solid black;`
- ❖ `border-right: 4px double blue;`

► **How to set the widths of borders**

- ❖ `border-width: 1px; /* all four sides */`
- ❖ `border-width: 1px 2px;`
- ❖ `border-width: 1px 2px 2px;`
- ❖ `border-width: 1px 2px 3px;`

► **How to set the style of borders**

- ❖ `border-style: dashed;`
- ❖ `border-style: solid none;`

► **How to set the color of borders**

- ❖ `border-color: #808080;`
- ❖ `border-color: black gray;`

- ▶ **How to set the width, style and color of border sides**
  - ❖ `border-bottom-width: 4px;`
  - ❖ `border-right-style: dashed;`
  - ❖ `border-left-color: gray;`
- ▶ When you code the `border-radius` property, you can assign one rounding radius to all four corners or a different radius to each corner.
- ▶ When you code the `box-shadow` property, positive values offset the shadow to the *right* or *down*, and **negative** values offset the shadow to the *left* or *up*.
- ▶ The third value in the `box-shadow` property determines how much the shadow is *blurred*, and the fourth value determines how *far* the blur is spread.
- ▶ The fifth value in the `box-shadow` property specifies the *color* of the shadow. If this is omitted, it is the same color as the border.
- ▶ These properties are supported by all modern browsers. If they are not supported by a browser, they are ignored.

# How to add rounded corners and shadows to borders

## ► The syntax for the border-radius and box-shadow properties

- ❖ `border-radius: radius;`
- ❖ `border-radius: topLeft topRight lowerRight lowerLeft;`
- ❖ `box-shadow: horizontalOffset verticalOffset blurRadius spread color;`

## ► The HTML for a section

```
<section>
```

```
    <a href="index.html">This Season's Speakers</a>
```

```
</section>
```

► The CSS for the section

```
section {  
    border: 5px double red;  
    width: 500px;  
    padding: 10px;  
    font-weight: bold;  
    text-align: center;  
    border-radius: 20px;  
    box-shadow: 8px 6px 3px black;  
}
```

# How to set background colors and images

- ▶ **background** - Background *color*, *image*, *repeat*, *attachment*, and position values.
- ▶ **background-color** - A color value or keyword that specifies the color of an element's background. You can also specify the transparent keyword if you want elements behind the element to be visible. This is the default.
- ▶ **background-image** - A relative or absolute URL that points to the image. You can also specify the keyword none if you do not want to display an image. This is the default.
- ▶ **background-repeat** - A keyword that specifies if and how an image is repeated. Possible values are *repeat*, *repeat-x*, *repeat-y*, and *non-repeat*. The default is *repeat*, which causes the image to be repeated both horizontally and vertically to fill the background.



- ▶ **background-attachment** - A keyword that specifies whether an image scrolls with the document or remains in a fixed position. Possible values are `scroll` and `fixed`. The default is `scroll`.
- ▶ **background-position** - One or two relative or absolute values or keywords that specify the initial horizontal and vertical positions of an image. Keywords are *left*, *center*, and *right*; *top*, *center*, and *bottom*. If a vertical position is not specified, *center* is the default. If not position is specified, the default is to place the image at the top-left corner of the element.
- ▶ **The syntax for the shorthand background property**
  - ❖ `background: [color][image][repeat][attachment][position];`
- ▶ **How to use the shorthand property**
  - ❖ `background: blue;`
  - ❖ `background: blue url("../images/texture.gif");`
  - ❖ `background: #808080 url("../images/header.png") repeat-y scroll center top;`

## ► How to control image repetition, position, and scrolling

- ❖ `background-repeat: repeat;`      `/* repeats both directions */`
- ❖ `background-repeat: repeat-x;`      `/* repeats horizontally */`
- ❖ `background-repeat: repeat-y;`      `/* repeats vertically */`
- ❖ `background-repeat: no-repeat;`      `/* doesn't repeats */`
  
- ❖ `background-position: left top;`      `/* 0% from left, 0% from top */`
- ❖ `background-position: center top;`      `/* centered horizontally, 0% from top */`
- ❖ `background-position: 90% 90%;`      `/* 90% from left, 90% from top */`
  
- ❖ `background-attachment: scroll;`      `/* image moves as you scroll */`
- ❖ `background-attachment: fixed;`      `/* image does not moves as you scroll */`

## ► Accessibility guideline

- ❖ Do not use a background color or image that makes the text that is over it difficult to read

# How to use CSS3 to set background gradients

- ▶ The CSS3 for linear gradients lets you create gradients for background without using images.
- ▶ Today, all of the modern browsers provide some support for linear gradients. Except for Internet Explorer, though, they implement them with their own prefixes: `-webkit-` for WebKit browsers like Chrome and Safari, `-moz-` for Mozilla Firefox, and `-o-` for Opera.
- ▶ The first parameter of a linear gradient indicates the direction the gradient will go: `"to right"`, for left to right, `"to bottom"` for top to bottom, `"to left"` for right to left, `"to top"` for bottom to top, and a number of degrees if the gradient should be on an angle.
- ▶ The direction is followed by two or more parameters that consist of a color and percent. The first percent indicates where the first color should start, the last percent indicates where the last color should end, and the percents in between indicate the points at which one gradient stops and the next one starts.

- ▶ **The syntax for coding linear gradient in the background-image property**
  - ❖ `background-image: linear-gradient(direction, color %, color%, ...);`
- ▶ **The HTML for two divisions**
  - ❖ `<div id="div1"></div>`
  - ❖ `<div id="div2"></div>`
- ▶ **The CSS for the two divisions using chrome prefixes (-webkit-)**
  - `#div1 { background-image: -webkit-linear-gradient(to right, white 0%, red %100); }`
  - `#div2 { background-image: -webkit-linear-gradient(45deg, red 0%, white 50%, blue %100); }`
- ▶ **A background-image property that creates red, white, and blue stripes**
  - `background-image: -webkit-linear-gradient(45deg, red 0%, red 33%, white 33%, white 66%, blue 66%, blue 100%);`

# How to layout pages

## ► How to float elements in 2-column layouts

- ❖ When you float an element to the *right* or *left*, the content that follows flows around it.
- ❖ When you use the float property for an element, you also need to set its width.
- ❖ To stop the floating before an element, use the clear property.

## ► The properties for floating and clearing elements

- ❖ **float** - A keyword that determines how an element is floated. Possible values are *left*, *right*, and *none*. None is the default.
- ❖ **clear** - Determine whether an element is cleared from flowing into the space left by a floated element. Possible values are *left*, *right*, *both*, and *none* (the default).

## ► The HTML for a web page with a sidebar

```
<body>  
  <aside>  
    <p>The luncheon starts 15 minutes after the lecture ends</p>  
  </aside>  
  <main>  
    <p>Welcome to the Faculty of Computing. We have some fascinating web  
    development tutorials for you this semester!</p>  
  </main>  
  <footer>  
    <p>Please call today at (+234) 8023-4567-123 to get your tickets!</p>  
  </footer>  
</body>
```

- The CSS for the web page for floating the sidebar

```
body { width: 500px; }
```

```
main, aside, footer {
```

```
    margin: 0;
```

```
    padding: 0 20px; }
```

```
aside {
```

```
    margin: 0 20px 10px;
```

```
    width: 150px;
```

```
    float: right;
```

```
    border: 1px solid black; }
```

```
footer { clear: both; }
```

- In the example above, if the clear property for the footer is not set, its content will flow into the space beside the floated element.

# How to use floating in a 2-column, fixed-width layout

- ▶ The main element is floated to the left and the aside is floated to the right. Then, it does not matter whether the aside comes before or after the main element in the HTML.
- ▶ Another alternative is to float both the main element and the aside to the left, but then the main element has to be coded before the aside in the HTML.

```
<body>
```

```
  <header><h2>Header</h2></header>
```

```
  <main><h2>Main content</h2></main>
```

```
  <aside><h2>Aside</h2></aside>
```

```
  <footer><h2>Footer</h2></footer>
```

```
</body>
```



► The CSS for the page

```
* { margin: 0; padding: 0; }  
body {  
    width: 962px;  
    background-color: white;  
    margin: 15px auto;  
    border: 1px solid black; }  
h2 { padding: 1em; }  
header { border-bottom: 2px solid #ef9c00;  
}  
main {  
    height: 350px;  
    width: 600px;  
    float: left;  
    border-right: 2px solid #ef9c00; }
```

```
aside {  
    width: 360px;  
    float: left; }  
footer {  
    clear: both;  
    border-top: 2px solid #ef9c00; }
```

► The CSS for the page when both columns are fluid

```
body {  
    width: 90%;  
    background-color: white;  
    margin: 15px auto;  
    border: 1px solid black; }  
  
main {  
    width: 66%;  
    height: 350px;  
    border-right: 2px solid #ef9c00;  
    float: left; }  
  
aside {  
    width: 33%;  
    float: right;  
  
}
```

- The CSS for the page when the aside is fixed and the section is fluid

```
body {
```

```
    width: 90%;
```

```
    background-color: white;
```

```
    margin: 15px auto;
```

```
    border: 1px solid black; }
```

```
main {
```

```
    float: left; }
```

```
aside {
```

```
    height: 350px;
```

```
    border-left: 2px solid #ef9c00;
```

```
    float: right; }
```

- The benefit of using fluid columns sizes is that the size of the page is adjusted to the resolution of the browser.
- The disadvantage is that changing the size of the columns may affect the typography or the appearance of the page.