IS 242 Web Application Development 1

Lecture 6: Introduction to CSS (Part 2)

Outlines of today’s lecture

- Continue our last lecture
- More on CSS properties
- CSS Inheritance

Font Family

- The **font family** of a text is set with the **font-family property**.
- The **font-family** property should hold several font names as a **"fallback"** system. If the browser does not support the first font, it tries the next font.
- Start with the font you want, and end with a **generic family**, to let the browser pick a similar font in the generic family, if no other fonts are available.
- Note: If the name of a font family is more than one word, it must be in quotation marks, like: "Times New Roman”.
- More than one font family is specified in a comma-separated list:

```
p {
    font-family: "Times New Roman", Times, serif;
}
```

Font Style

- The font-style property is mostly used to specify italic text.

- This property has three values:
  - **normal** - The text is shown normally
  - **italic** - The text is shown in italics
  - **oblique** - The text is "leaning" (oblique is very similar to italic, but less supported)

```
p.normal {
     font-style: normal;
}

p.italic {
     font-style: italic;
}

p.oblique {
     font-style: oblique;
}
```

Font Size

- The **font-size** property sets the size of the text.

- Being able to manage the text size is important in web design. However, you should not use font size adjustments to make paragraphs look like headings, or headings look like paragraphs.

- Always use the proper HTML tags, like <h1> - <h6> for headings and <p> for paragraphs.

Examples:

```
h1 {
      font-size: 40px;
}
h1 {
      font-size: 2.5em; /* 40px/16=2.5em */
}
```

CSS Lists

- The **CSS list properties** allow you to:
  - Set different list item markers for ordered lists
  - Set different list item markers for unordered lists
  - Set an image as the list item marker

- The type of list item marker is specified with the **list-style-type** property:

  Examples

  ```
  ul {
      list-style-type: circle;
  }
  ol {
      list-style-type: lower-alpha;
  }
  ```

- To specify an image as the list item marker, use the **list-style-image** property:

  ```
  ul {
     list-style-image: url('sqpurple.gif');
  }
  ```

CSS Tables

- The look of an HTML table can be greatly improved with CSS
- To specify table borders in CSS, use the **border** property.
- The example below specifies a black border for <table>, <th>, and <td> elements:

```
table, th, td {
    border: 1px solid black;
}
```

- The **border-collapse** property sets whether the table borders are collapsed into a single border or separated:

```
table {
    border-collapse: collapse;
}
table, th, td {
    border: 1px solid black;
}
```

CSS Tables (Cont.)

- Width and height of a table is defined by the **width** and **height** properties.
- The example below sets the width of the table to 100%, and the height of the <th> elements to 50px:

```
table {
     width: 100%;
}

th {
     height: 50px;
}
```

- The **text-align** property sets the horizontal alignment, like left, right, or center.
- By default, the text in <th> elements are center-aligned and the text in <td> elements are left-aligned.
- The following example left-aligns the text in <th> elements:

```
th {
     text-align: left;
}
```

- The **vertical-align** property sets the vertical alignment, like top, bottom, or middle.
- By default, the vertical alignment of text in a table is middle (for both <th> and <td> elements).
- The following example sets the vertical text alignment to bottom for <td> elements:

```
td {
     height: 50px;
     vertical-align: bottom;
}
```

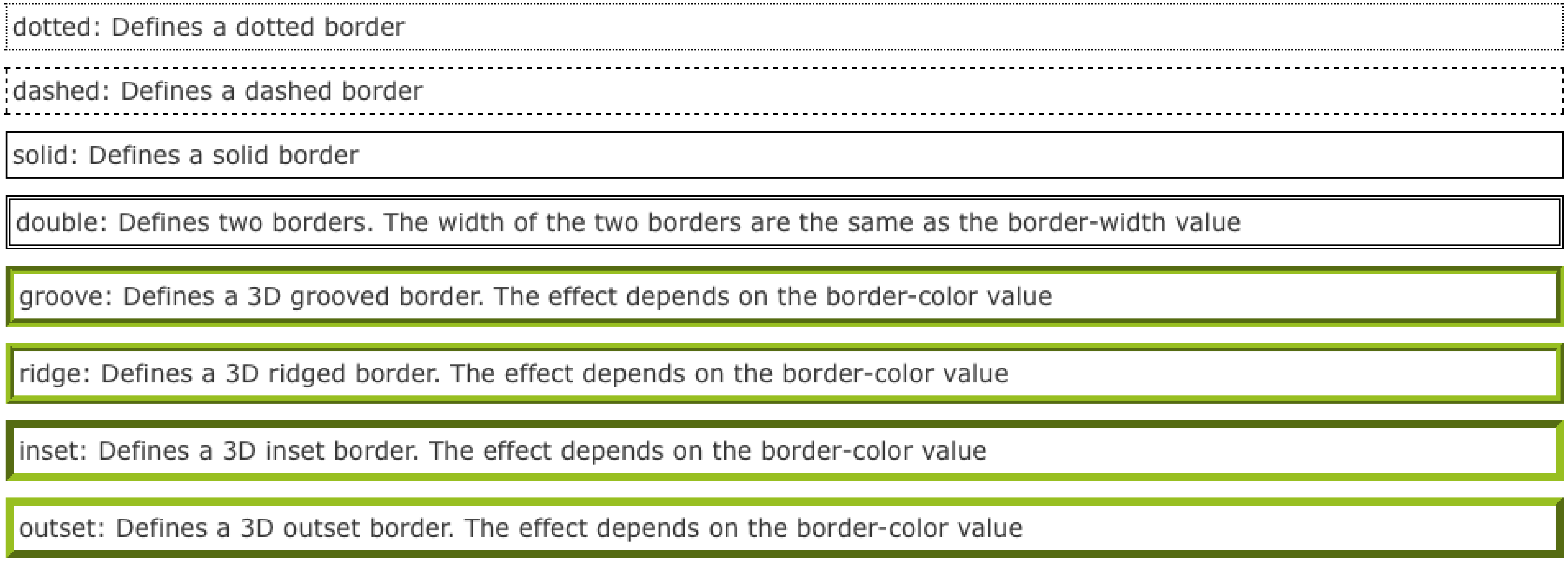
CSS Borders

- The CSS border properties allow you to specify the style, size, and color of an element's border.

Border Style

- The border-style property specifies what kind of border to display.
- Note: **None of the border properties will have ANY effect unless the border-style property is set!**

- *border-style values:*



dotted: Defines a dotted border

dashed: Defines a dashed border

solid: Defines a solid border

double: Defines two borders. The width of the two borders are the same as the border-width value

groove: Defines a 3D grooved border. The effect depends on the border-color value

ridge: Defines a 3D ridged border. The effect depends on the border-color value

inset: Defines a 3D inset border. The effect depends on the border-color value

outset: Defines a 3D outset border. The effect depends on the border-color value

CSS Borders (Cont.)

- Border Width
  - The **border-width** property is used to set the width of the border.
  - The width is set in pixels, or by using one of the three pre-defined values: thin, medium, or thick.
  - *Note: The "border-width" property does not work if it is used alone. Use the "border-style" property to set the borders first.*

```
p {
    border-style: solid;
    border-width: 5px;
}
```

- Border Color
  - The **border-color** property is used to set the color of the border.
  - You can also set the border color to "transparent".
  - If the border color is not set it is **inherited from the color property** of the element.
  - *Note: The "border-color" property does not work if it is used alone. Use the "border-style" property to set the borders first.*

```
p {
    border-style: solid;
    border-color: red;
}
```

CSS Margin

- The **margin** clears an area around an element (outside the border). The margin does not have a background color, and is completely transparent.

- The top, right, bottom, and left margin can be changed independently using separate properties. A **shorthand** margin property can also be used, to change all margins at once.

CSS Margin (Cont.)

Margin - Individual sides

In CSS, it is possible to specify different margins for different sides of an element:

```
p {
    margin-top: 100px;
    margin-bottom: 100px;
    margin-right: 150px;
    margin-left: 50px;
}
```

Margin - Shorthand property

To shorten the code, it is possible to specify all the margin properties in one property. This is called a shorthand property.

The shorthand property for all the margin properties is "margin":

```
p {
    margin: 100px 50px;
}
```

CSS Padding

- The CSS **padding** properties define the space between the element **border** and the element **content**.

- The padding clears an area around the content (inside the border) of an element. The padding is affected by the background color of the element.

- The top, right, bottom, and left padding can be changed independently using separate properties. A **shorthand** padding property can also be used, to change all paddings at once.

Width and Height of an Element

- When you set the **width** and **height** properties of an element with CSS, you just set the width and height of the content area.

- To calculate the full size of an element, you must also add padding, borders and margins.

- Let's style a <div> element to have a total width of 350px:

```
div {
    width: 320px;
    padding: 10px;
    border: 5px solid gray;
    margin: 0;
}
```

- The total width of an element should be calculated like this:

Total element width = width + left padding + right padding + left border + right border + left margin + right margin
The total height of an element should be calculated like this:
Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

CSS Display and Visibility

- Changing an inline element to a block element, or vice versa, can be useful for making the page look a specific way, and still follow web standards.
- The following example displays <li> elements as inline elements:

```
li {
     display: inline;
}
```

- Hiding an element can be done by setting the display property to "none" or the visibility property to "hidden". However, notice that these two methods produce different results:

- visibility:hidden hides an element, but it will still take up the same space as before.

- display:none hides an element, and it will not take up any space.

Inheritance in CSS

- If you specify the **font-family** or color properties on the <body> element, they will apply to most child elements.

- This is because the value of the font-family property is **inherited** by child elements. It saves you from having to apply these properties to as many elements (and results in simpler style sheets).

- You can compare this with the **border** property; it is **not inherited** by child elements.

- You can force a lot of properties to inherit values from their parent elements by using **inherit** for the value of the properties.

References

- www.w3schools.com
- Duckett, J. (2011). *HTML and CSS: Design and Build Websites*. John Wiley & Sons.
- Deitel & Deitel (2011). *Internet and World Wide Web How to Program, 5th Edition, Harvey & Paul Deitel & Associates.*