STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

CS 146: Intro to Web Programming and Project Development

Instructor: Iraklis Tsekourakis

Lieb 213

Email: itsekour@stevens.edu

Introduction to CSS

CSS Lists

- In HTML, there are two main types of lists:
  - unordered lists (<ul>) - the list items are marked with bullets
  - ordered lists (<ol>) - the list items are marked with numbers or letters
- 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
  - Add background colors to lists and list items

CSS Lists

- Use classes to make different lists, this way you can use ul.name or ol.name for different styles
- list-style-type:
  - For unordered use circle, disc, none, square
  - For ordered, options galore! armenian, cjk-ideographic, decimal, decimal-leading-zero, georgian, hebrew, hiragana, hiragana-iroha, katakana, katakana-iroha, lower-alpha, lower-greek, lower-latin, lower-roman, upper-alpha, upper-greek, upper-latin, upper-roman
- list-style-image: url('filename.gif');
- list-style-position: inside or outside

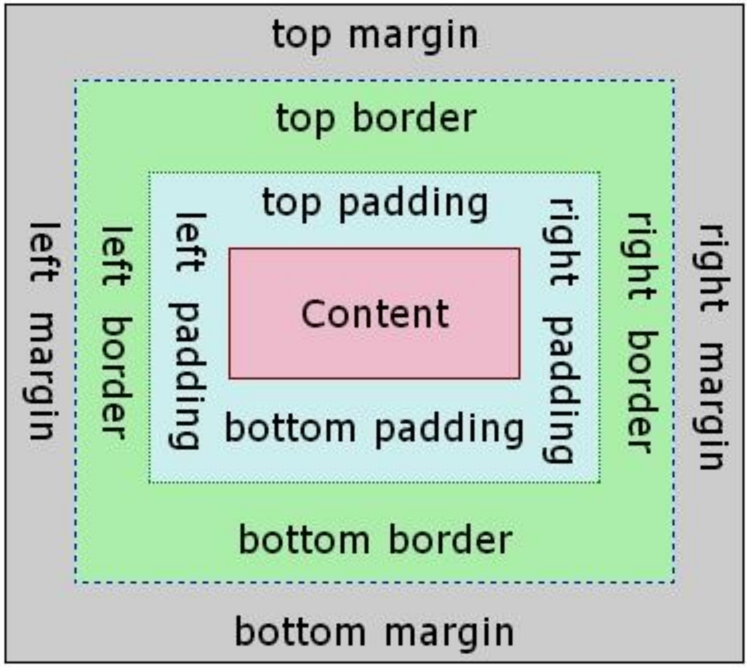
IE and Opera Strike back!

- When using images for lists, they display a bit off in “certain” browsers
- You can “fix it” like this

```
ul {
    list-style-type: none;
    padding: 0px;
    margin: 0px;
}
li {
    background-image: url(‘file.gif’);
    background-repeat: no-repeat;
    background-position: 0px 5px;
    padding-left: 14px;
}
```

CSS The Box Model

- All HTML elements can be considered as boxes; in CSS, the term "box model" is used when talking about design and layout
- The CSS box model is essentially a box that wraps around every HTML element; it consists of: margins, borders, padding, and the actual content



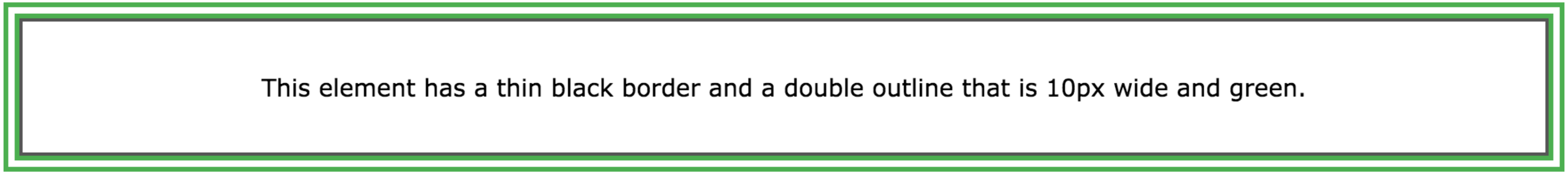
CSS Box Model

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

320px (width)
+ 20px (left + right padding)
+ 10px (left + right border)
+ 0px (left + right margin)
= 350px

CSS Outline

- The CSS outline properties specify the style, color, and width of an outline
- An outline is a line that is drawn around elements (outside the borders) to make the element "stand out"
- However, the outline property is different from the border property - The outline is NOT a part of an element's dimensions; the element's total width and height is not affected by the width of the outline



This element has a thin black border and a double outline that is 10px wide and green.

CSS Outline

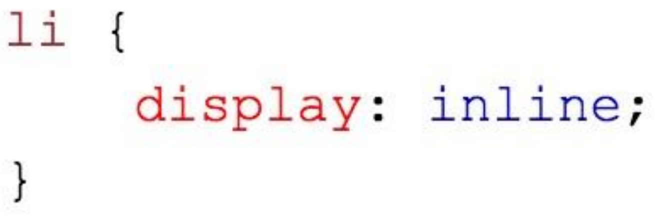
- The outline-style property can have one of the following values:
  - dotted - Defines a dotted outline
  - dashed - Defines a dashed outline
  - solid - Defines a solid outline
  - double - Defines a double outline
  - groove - Defines a 3D grooved outline. The effect depends on the outline-color value
  - ridge - Defines a 3D ridged outline. The effect depends on the outline-color value
  - inset - Defines a 3D inset outline. The effect depends on the outline-color value
  - outset - Defines a 3D outset outline. The effect depends on the outline-color value
  - none - Defines no outline
  - hidden - Defines a hidden outline

CSS: The Display Property

- The display property is the most important CSS property for controlling layout
- The display property specifies if/how an element is displayed
- Every HTML element has a default display value depending on what type of element it is; the default display value for most elements is block or inline
- *display: none;* is commonly used with JavaScript to hide and show elements without deleting and recreating them

Overriding the Default Display Value

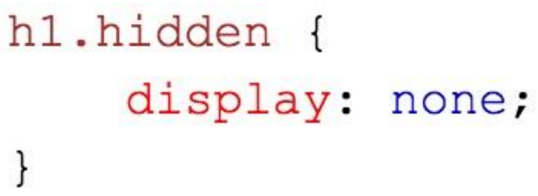
- As mentioned, every element has a default display value; however, you can override this
- Changing an inline element to a block element, or vice versa, can be useful for making the page look a specific way, and still follow the web standards
- A common example is making inline <li> elements for horizontal menus:



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

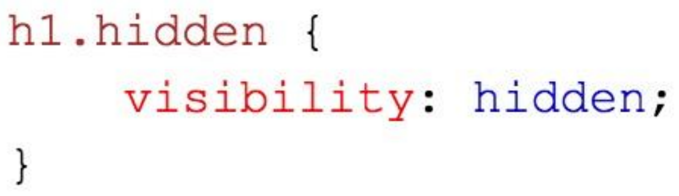
Hide an Element

- Hiding an element can be done by setting the display property to none; the element will be hidden, and the page will be displayed as if the element is not there:



```
h1.hidden {
    display: none;
}
```

- *visibility:hidden;* also hides an element; however, the element will still take up the same space as before. The element will be hidden, but still affect the layout:



```
h1.hidden {
    visibility: hidden;
}
```

CSS: The Position Property

- The position property specifies the type of positioning method used for an element
- There are four different position values:
  - static
  - relative
  - fixed
  - absolute
- Elements are then positioned using the top, bottom, left, and right properties; however, these properties will not work unless the position property is set first; they also work differently depending on the position value

CSS Position

- Static positioned elements are not affected by the top, bottom, left, and right properties; they follow the flow of the page
- For relative positioned elements, setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position
- An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled; the top, right, bottom, and left properties are used to position the element
- An element with *position: absolute;* is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed)

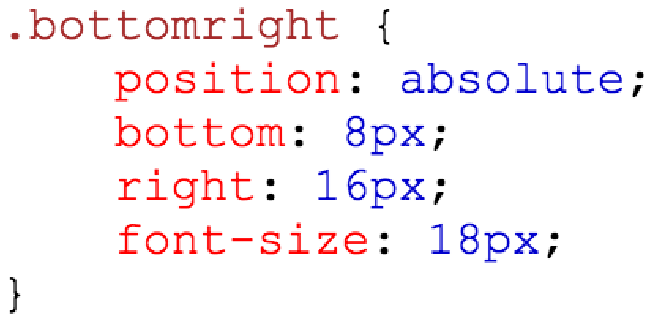
Overlapping Elements

- When elements are positioned, they can overlap other elements
- The z-index property specifies the stack order of an element (which element should be placed in front of, or behind, the others)
- An element can have a positive or negative stack order
- An element with greater stack order is always in front of an element with a lower stack order

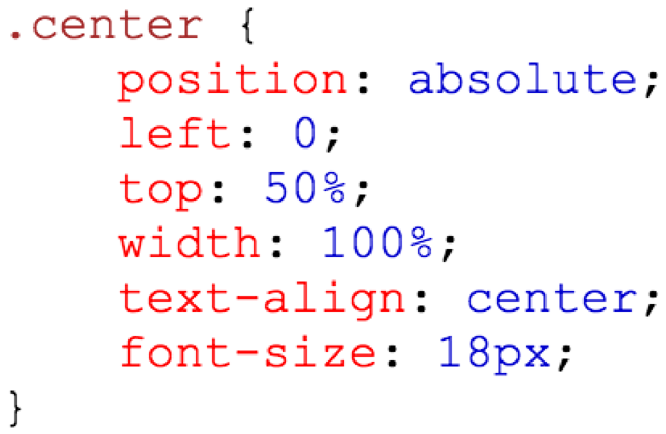
Positioning/Display/Box Model Exercise

Let’s see!

Solution



```
.bottomright {
    position: absolute;
    bottom: 8px;
    right: 16px;
    font-size: 18px;
}
```



```
.center {
    position: absolute;
    left: 0;
    top: 50%;
    width: 100%;
    text-align: center;
    font-size: 18px;
}
```