

# CSC435: Web Programming

## Lecture 6: CSS: Box Model

Bei Xiao

American University

Feb 3, 2014

# Page Layout

- Look at [www.xkcd.com](http://www.xkcd.com)
- CSS can manage page layout

# The Box Model of HTML Element

- Every HTML element can be thought of as a box, with a width and height.
- To get a desired page layout, you must carefully specify the width and height of each element.
- Margins, padding, and borders are also important.
- You must understand the box model to correctly specify your dimensions.

# The Box Model

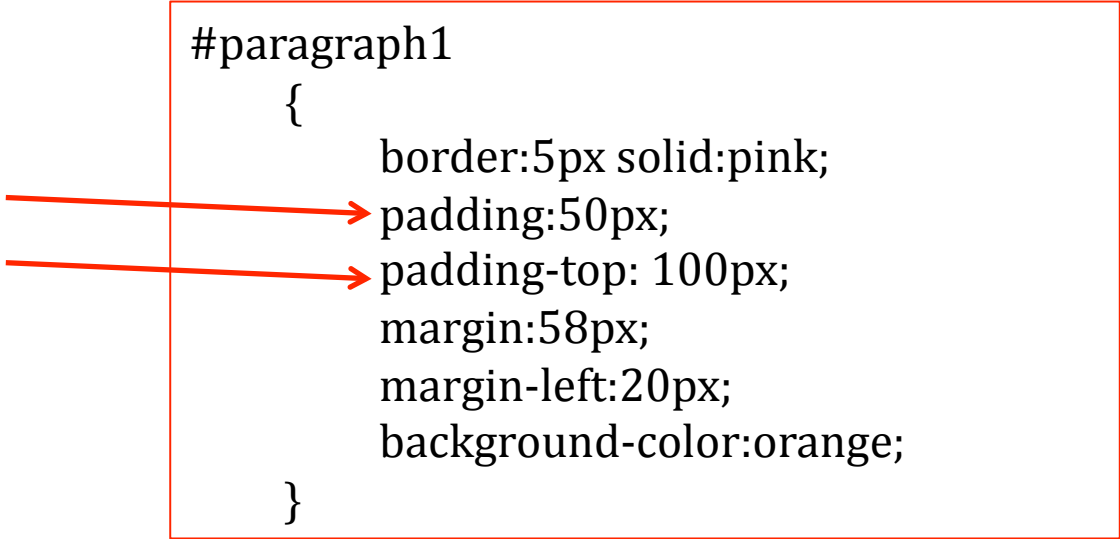


# Ordering of the attributes matters

Padding can overwrite padding-top, if it is specified after padding-top

The  
ordering  
can be  
crucial

```
#paragraph1
{
    border:5px solid:pink;
    padding:50px;
    padding-top: 100px;
    margin:58px;
    margin-left:20px;
    background-color:orange;
}
```

A diagram illustrating the importance of attribute ordering in CSS. It shows a CSS rule for the selector #paragraph1. The rule contains several properties: border:5px solid:pink; padding:50px; padding-top: 100px; margin:58px; margin-left:20px; and background-color:orange;. Two red arrows originate from the text 'The ordering can be crucial' on the left. One arrow points to the 'padding:50px;' line, and the other points to the 'padding-top: 100px;' line. The entire code block is enclosed in a thin red rectangular border.

# The Box Model



# Width and Height

- The total width of the element in the example below is 250px:

width:220px;

padding:10px;

border:5px;

margin:0px;

220 px (width) + 20px (left+right padding) +10px (left +right border) +0px (left +right margin)= 250px

Total element width = width + left padding + right padding + left border + right border + left margin + right margin

Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

# Computing Box Dimensions

- “width” and “height” of an element refers to the content. If you have margins, padding, etc. those also add to the dimensions of the box.
- This is important because it is the entire box, and not merely the content, that takes the space of the page.



# Ways to specify size

- Pixels (px)
- Pt (1/72 inch) i.e. 12 pt font
- mm, cm, in
- em, ex (for font), 1em = 12px
- Percent<sup>0</sup>% (usually of the browser window dimension) E.g.:  
body{font-size:50%}, half the width of the browser window

# Exercise 1

- Finish the declarations that create the effects so that the paragraph box has the same width as the heading box.

# Exercise

- Finish the declarations that create the effects so that the paragraph box has the same width as the heading box.
- Answer:  $225(\text{width}) + 5(\text{left margin}) + 5(\text{left Border}) + 25(\text{left padding}) + 25(\text{right padding}) + 5(\text{left border}) + 10(\text{left margin}) = 300$

# Positioning

- Uses the “position” property, i.e. position: static
- Default is static, (normal flow).
- Fixed-a fixed position in the browser window.
- Relative- relative to its normal static position.
- Absolute-positioned relative to first parent with position other than static.

# Positioning

- Uses the “position” property, i.e. position: static
- Default is static, (normal flow).

# Normal Flow of the page

- Statically-positioned elements are positioned according to the “normal flow of the page”.
- Block element: always take up the full width available, with **newlines** before and after.  
*div, h1,p, ul, table, etc.*
- In-line element: only takes as much width as needed and do not force new lines  
*span, a, img, etc*

# Positioning

- Uses the “position” property, i.e. position: static
- Default is static, (normal flow).
- Fixed-a fixed position in the browser window.

# Positioning

- Uses the “position” property, i.e. position: static
- Default is static, (normal flow).
- Fixed-a fixed position in the browser window.
- Relative- relative to its normal static position.

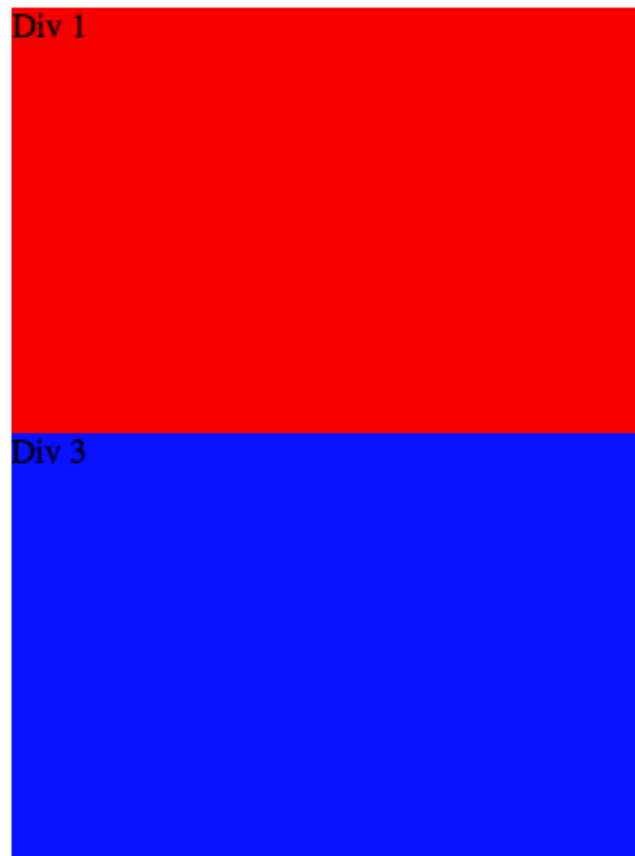


# Positioning

- Uses the “position” property, i.e. position: static
- Default is static, (normal flow).
- **Fixed**-a fixed position in the browser window.
- **Relative**- relative to its normal static position.
- **Absolute**-positioned relative to first parent with position other than static.

# Exercise

- Using positioning , make the Div 3 appear above Div2.



# Homework & Readings

Chapter 14, Robbins.

Code Academy, Web fundamentals:

[http://www.codecademy.com/courses/web-beginner-en-TlhFi/0/1?curriculum\\_id=50579fb998b470000202dc8b](http://www.codecademy.com/courses/web-beginner-en-TlhFi/0/1?curriculum_id=50579fb998b470000202dc8b)