CSS Handout

CS 3172 Summer 2003

1 History

CSS1 Dec 1996

- about 50 properties,
- mostly for screen use

CSS2 May 1998

- about 70 more properties
- includes all of CSS1
- includes properties for non-screen presentation

2 Cascading vs. Inheritance

In *inheritance* rules are applied and overwritten by parent elements. In *cascading* all rules that apply to an element are used but some can be overwritten. Here's the order for CSS: first rules from the browser software are applied, the the user's style sheet, the the author's rules. If there is a tie then the last rule applied wins. But rules can be labeled `!important` to make them more win a tie. The user's `!important`rules overrules all others.

3 Selectors and Rules

Examples of Some Selectors

By	in CSS	in XHTML
ID alone	`#abc`	`id="abc"`
element type	`table`	`<table>...</table>`
class	`.note`	`class="note"`

element type & class	`table.note`	`<table class="note">`
		⋮
		`</table>`

attribute	`[title]`
	`table[border]`
	`[align="left"]`
	`[rel ~= "home"]`
		`<dl title="some text">`
		`<table border="1">`
		`<div align="left">`
		`<a rel="homepage"...</a>`

4 Colours

Three main ways to specify colours:

- by name (e.g. `green`)
- by rgb value (red-green-blue)
  - 0%, 0%, 0% is black; 100%, 100%, 100% is white; x%, x%, x% is a shade of grey
  - specified by percentage (e.g. `100%,35.5%,10%`)
  - specified by value 0 – 255 (e.g. `255,91,20`)
  - specified as hexadecimal triplet (e.g. `#FF5B14`)
- by 'System Name' (e.g. `WindowText`)

Browser-Safe Colours[1]

- 216 non-dithering colours (look flat)
- *not* exactly the same on all monitors
- photos, etc. use more colours and are dithered
- can be used for some consistency amongst browsers

5 The Box Model

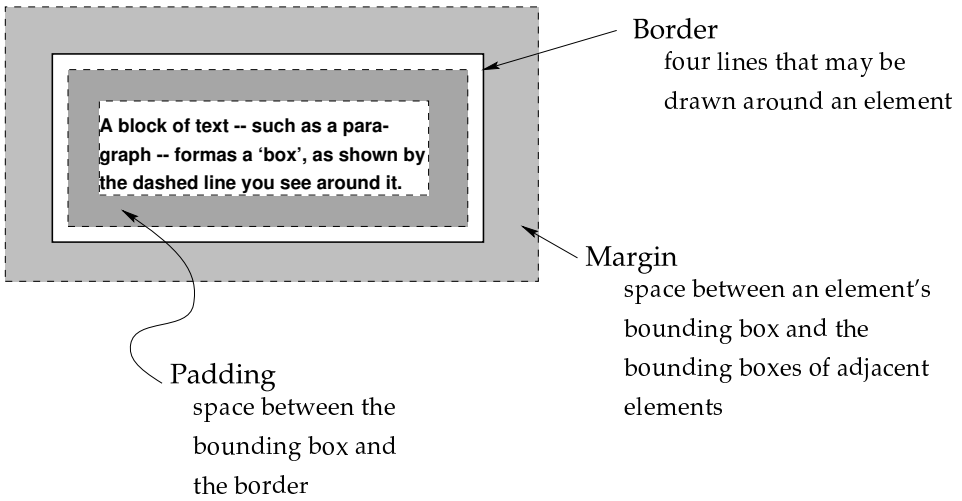

Figure adapted from Lie & Bos [2] (especially Figure 9.2 and text on p. 180)

Every element is drawn in a bounding box with three parts:

margin space between the enclosing box and the border

border decoration around content (can set width & style for all four sides)

padding space between border and element

Collapsing Margins

- if two elements touch then their top and bottom margins overlap
- padding or borders can keep the margins from touching

6 Floating Elements

There are many properties for floating elements. Here are the most important ones.

float

- values: `left` or `right` or `none`
- example `left` move the element as far to the left as possible until the margin, padding, or border of a *block-level element* is touched.

clear

- values: `none` or `left` or `right` or `both`
- where floating elements are not allowed to be
- example*

```
/** Example of float and clear **/
/* Make all images float left: */
img { float: left }
/* H2 headings must not be next to images: */
h2  { clear: both }
```

position

- values: `static` or `relative` or `absolute` or `fixed`
- `static`: normal elements, placed relative to their parents
- `relative`: placed relative to enclosing block (usually the parent)
- `fixed`
  - for block elements only
  - relative to viewport, e.g. browser window, or page of paper
- `absolute`
  - for block elements only
  - like `fixed` but not tied to the viewport

visibility values: `visible` or `hidden`

x-index value: integer (smallest values at the back)

*From Lie and Bos [2], page 203.

7 Tables

There are two models for table borders: collapsing and separate.
Set like this: `table { border-collapse: separate }`

Collapsing model

- only 1 border is shown between cells
- where there are two to choose from, the largest is used
- borders can be set on rows, columns, cells, tables, and groups

Separate model

- borders can only be set on cells and tables

8 See Also

- Eric Meyer on browser rendering models and how they affect CSS (in the readings part of the resource section)
- CSS examples (in the materials section)
- CSS lecture notes

References

[1] Jim Krause. *color index*. HOW Design Books, 2002.

[2] Håkon Wium Lie and Bert Bos. *Cascading Style Sheets: Designing for the Web*. Addison-Wesley an imprint of Pearson Education, second edition, 1999.