Basic text markup and style

Chapter 1

Outline

- A. Structured text
  - HTML markup: Inline vs. block elements
- B. Text styles and CSS Box Model
  - Embedded style. Type selector
  - Text styles. Paragraph styles.
  - Font style inheritance and elastic design
  - CSS Box Model
- C. Basic CSS selectors
  - Id and class selector
  - Descendant selector
- D. Lists: markup and style
  - CSS selector specificity

A. Structured text

- Structure of an HTML doc
- Basic elements (block and inline)
- Whitespace handling and special characters

Basic structure

- An HTML doc has the root element <html>, which contains two children:
  - <head>, which describes the HTML doc
  - <body>, which provides the content of the HTML doc, usually shown in the browser window

```
<!DOCTYPE html>
<html>
  <head> … </head>
  <body>
    …
    …
  </body>
</html>
```

Functions of <head>

- provide data about the page
  - <title> : appears in title bar of browser window
  - <meta> : metadata, information about the html page. Defines the character encoding
- link to external resources
  - <style> : embedded style sheet
  - <script> : add behavior with JavaScript
- embed style and script
  - <link> : link to external resources, e.g. CSS style sheet
  - <script> : add behavior with JavaScript

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
    <meta … />
    <link … />
    <script …> </script>
  </head>
  <body> … </body>
</html>
```

Block and inline elements in <body>

Most elements that can appear inside <body> belong to two kinds:

- Block elements are displayed in rectangular blocks stacked from top to bottom
  - May contain text, inline elements and block elements
- Inline elements are displayed inline with the text flow
  - contain text and other inline elements.
  - Inline elements *cannot* contain block elements

```
<body>
  <h1>Structure of HTML</h1>
  <p>An HTML doc has a <em>tree structure</em>,
   where a parent element contains
    some child elements.</p>
</body>
```

Common block elements

- Common block elements:
  - <p> : paragraph
  - <h1>, <h2> … <h6> : headings on different levels
  - <pre> : preformatted content
  - <blockquote>
  - Lists: <ul>, <ol>, <li>, <dl>, <dt>, <dd>
  - <div>, <section>, <aside> …
  - Tables: <table>
  - Forms: <form>

Text blocks

- <body> contains some block elements.
- Block elements are displayed in rectangles, stacked from top to bottom.

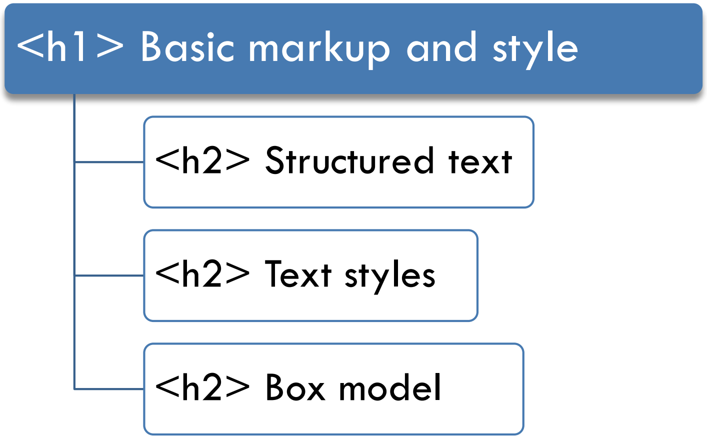
Example: p101.html

```
<body>
   <h1>Basic markup and style</h1>
   <p>This chapter studies the basics of HTML and CSS</p>
   <h2>Structured text</h2>
   <p>Structure of HTML document. Common inline
       and block elements for marking up text.</p>
   <h2>Text styles</h2>
   <p>Basic text styling in CSS. Style rules.</p>
</body>
```

Headings

- Headings are titles of chapters, sections and sub-sections.
  - <h1> is a heading on level 1 (e.g. chapter), <h2> is a heading on level 2 (e.g. section), and <h3> is a heading on level 3 (e.g. sub-section).
  - Similarly, <h4>, <h5> and <h6>.

```
<body>
  <h1>Basic markup and style</h1>
  <p>…</p>
  <h2>Structured text</h2>
  <p>…</p><p>…</p>
  <h2>Text styles</h2>
  <p>…</p><p>…</p>
  <h2>Box model</h2>
  <p>…</p><p>…</p>
</body>
```



Paragraphs

- □ <p> contains some text formatted as a paragraph.
- □ New line characters ⏎(e.g. input by pressing ENTER) within a <p> element do not break the text into paragraphs.
- □ White space characters (e.g. space • and new line ⏎) between block elements do not affect formatting.

```
<body>
•••<p>This chapter studies the⏎
basics of HTML and CSS</p>⏎
•••<p>Structure of HTML document. Common inline⏎
       and block elements for marking up text.</p>⏎
⏎•••
•••<p>Basic text styling in CSS. Style rules.</p>⏎
</body>
```

Preformatted Text <pre>

- <pre> is a special block element that preserves spaces, tabs, and carriage return characters in content.
  - Similar to <p>, it requires inline content
  - Use it when the formatting has meaning. E.g. program source code
  - The first line break after <pre> is ignored.

```
<pre>
int sum=0;
for (int i=1; i&lt;=10; i++)
   sum+=i;
</pre>
```

Whitespace and line break

- In most cases, HTML treats spaces, tabs and carriage returns as the same whitespace character. In addition, multiple whitespace characters collapse to one.
- To enforce a line break within a paragraph, use the empty element <br/> (only use this if the line break changes the meaning of the passage, e.g. as in a poem)
- To avoid whitespace collapsing, use the non-breaking space &nbsp;

```
<p>You          cannot
   add a line break
like this.  <br/>Use the br element to do so. </p>
```

You cannot add a line break like this.
Use the br element to do so.

Common inline elements

- Common inline elements:
  - <em>, <strong> : text emphasis
  - <abbr> : abbreviation
  - <q> : inline quotes
  - <br /> : line break
  - <code> : inline computer code
  - <span>
  - <a> : hyperlink
  - <img /> : images
  - <object>, <iframe>

Emphasis

- <em> represents stress emphasis
  - Usually, <em> is formatted as *italic*. You may customize the presentation by CSS.
  - <i> has different meaning in HTML5

```
<p>Compare these two sentences.</p>
<p>Call the doctor <em>now</em>!</p>
<p>Call the <em>doctor</em> now!</p>
```

Strong importance

- <strong> represents strong importance
  - Usually, <strong> is formatted as **bold**. You may customize the presentation by CSS.
  - <b> has different meaning in HTML5

```
<p><strong>Warning</strong>
Guard dogs</p>
```

Warning
Guard dogs

Inline quote

- Quote using inline elements <q>
  - May use CSS to specify suitable quotation marks

```
<p>Einstein said <q>Everything should be made as
simple as possible, but not simpler</q>. Therefore, I
think the midterm test should also be simple.</p>
```

Block quote

- The block element <blockquote> is used to quote a passage. The content should be block elements.

```
<blockquote cite="http://en.wikipedia.org/wiki/Css">
<p>Cascading Style Sheets (CSS) is a style sheet
language used to describe the presentation semantics (that
is, the look and formatting) of a document</p>
</blockquote>
```

Check the formatting in p102.html

Abbreviation

- <abbr> marks up abbreviations with their long form.
  - The long form (in title attribute) is usually shown as tooltip when the cursor is over the abbreviation.

```
<p>You can use <abbr title="structured query
language">SQL</abbr> to query data from a
relational <abbr title="Database Management
System">DBMS</abbr>.</p>
```

Computer code <code>

- Use <code> to mark up inline computer code

```
<p>To read student name and ID of all students,
  use <code>SELECT name, studid from
student</code>.</p>
```

Semantic HTML

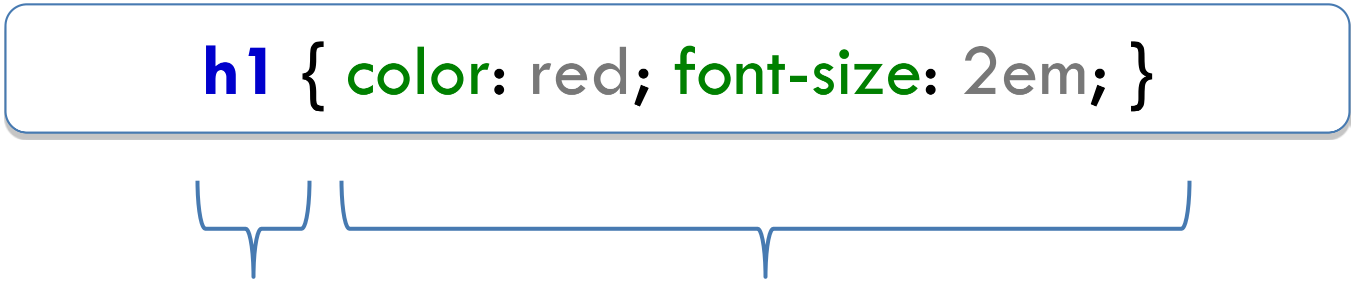
- You should use HTML according to the meaning of data.
  - Don’t use HTML to adjust the presentation of data. Use CSS to format the data
  - E.g. <em> is usually formatted as italic. But you should only use it to emphasize some content.
  - <table> should only be used to mark up tabular data. Don’t use <table> for page layout

B. Text styles

- Syntax of CSS style rule
  - Type selector
- Text style properties
  - Font color, font face, font size
  - italic, bold, underline
- Inheritance of text style properties
  - Elastic design
- Paragraph style
  - Line spacing, indentation, alignment, …
- CSS Box Model
  - Content, padding, border, margin
  - Margin collapsing

CSS style rule

- A CSS style rule consists of two parts
  - A selector to choose some elements
  - A list of property settings to configure the presentation of the selected elements.



```
h1 { color: red; font-size: 2em; }
```

A **selector** specifies which elements the rule applies to.

Inside a pair of braces is a list of **property** settings. The settings are separated by semicolon.

Type selectors

```
em { color: red; }
```

- An type selector matches elements of a given type
  - E.g. set font color of em elements to red

Embedded style

- A convenient way to apply style rules is to embed them in the **style** element inside the <head>
  - Style applies to the current doc only
  - In production, it is recommended to use external CSS style sheet. (see chap 2)

```
<head>…
  <style>
      /* style rules */
      h1 { color: red; font-size: 2em; }
  </style>
</head>
<body>
  … <h1>Big heading in red</h1>  …
</body>
```

color

```
em { color: red; }
```

- The color property sets text color.
- Ways to specify a color value:
  - Use standard color name, or
    - aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, yellow, **transparent**.
  - red, green and blue components.
    - color: #ff0000 *or* color: #f00 *(hexadecimal)*
    - color: rgb(255,0,0) *(decimal, 0-255)*
  - CSS3 also supports alpha transparency
    - color: rgba(255,0,0,0.5) *(alpha, 0-1)*
  - Ref: http://www.w3schools.com/cssref/css_colors.asp

font-family

- The font-family property suggests a list of font
- Difficult to get the same font appearance in different platforms (Mac, Windows, mobile phones….)
  - Fonts of the same name may look slightly different. Ref http://www.typechart.com/
  - Operating systems have different fonts installed. Ref http://cssfontstack.com/
- Use font stack to obtain predictable font appearance
  - The browser will use the first font available in the list
  - End font stack with generic font families (serif, sans-serif, monospace, etc). They map to existing fonts of the same family in the browser machine.

```
body { font-family: "Times New Roman", serif; }
h1 { font-family: Arial, Helvetica, sans-serif; }
```

Further readings on web fonts

- Search 'web typography' for general info, e.g. http://www.smashingmagazine.com/2009/01/27/css-typographic-tools-and-techniques/
- CSS3 @font-face supports downloadable web font
  - Google font: http://www.google.com/fonts
  - Open source Pan-CJK font: http://blog.typekit.com/2014/07/15/introducing-source-han-sans/
  - More Chinese web font: http://en.justfont.com/
- A font stack is a list of similar fonts with fallback for different platforms
  - http://www.awayback.com/revised-font-stack/ : Fonts in popular sites and their availability in different platforms
  - : common fonts in Win and Mac
  - http://awesome-fontstacks.com/ : font bundles and webfonts
  - http://www.typechart.com/ : compare similar font in Win and Mac

Font-related properties

Property	Common values	Other values
font-family	serif, sans-serif	monospace, other font names
font-size	2em, 80%, 12pt	small, medium, large, larger, smaller
font-weight	**bold**	normal
font-style	*italic*	normal
font-variant	small-caps	normal
text-decoration	underline	overline / line-through / none
text-transform	uppercase	capitalize / lowercase / none
letter-spacing	0	
word-spacing	0	

http://csstypeset.com/

font-size

- The font-size property sets the size of font
  - In relative unit, relative to the current font size: e.g. font-size: 1.5em , font-size: 150%
  - In absolute unit: e.g. font-size: 12pt , font-size: 16px
- The default font-size (for the root element <html>) is usually 16px
  - On desktop environment, usually 12pt = 16px
- Modern browsers scale font size and other length in proportion when users zoom-in / out.
  - .. absolute unit is not really absolute …
- Further info: http://kyleschaeffer.com/best-practices/css-font-size-em-vs-px-vs-pt-vs/

font property

- You can combine several font-related properties using the shorthand font property.
  - The order of properties is important

```
p {
  font: italic bold 2em "Times New Roman", serif;
}
```

```
p {
  font-style: italic;
  font-weight: bold;
  font-size: 2em;
  font-family: "Times New Roman", serif;
}
```

Styling paragraphs

- line-height
  - E.g. line-height: 2 to use double line spacing
- text-align
  - left, right, center, justify
  - E.g. h1 { text-align: center; } places the heading in the center
- text-indent
  - How much to indent the first line of a paragraph
- margin
  - E.g. p { margin: 2em 0em; } sets a 2em margin between paragraphs and no margin on left and right sides.
  - More about this in the section about CSS Box Model …

Inheritance of properties

- Most font related properties are inherited
  - Child elements inherit settings from parent
  - Child elements can also override the settings

```
 body { font-family: sans-serif; font-size: 16px; }
 h1 { font-size: 20px; }

<body>
   <h1>Big heading at 20px</h1>
   <p>normal text at 16px.
        But all text uses sans-serif font..</p>
</body>
```

Elastic design

- □ A common practice is to set all font-size in em, relative to the base font size of the body.
- □ You may then adjust the font-size of all text using a percentage font-size for <body>

```
<html>
  <head><title
    <style>
      body { font-size: 80%; }
      h1 { font-size: 2em; }
      h2 { font-size: 1.5em; }
    </style>
  </head>
  <body>
      <h1>Heading 1</h1>
      <h2>Heading 2</h2>
      <p>some normal text</p>
  </body>
</html>
```

Adjust this to resize all text proportionally.

Exercise: assuming the default font size for <html> is 16px, calculate the font size of all elements.

Exercise

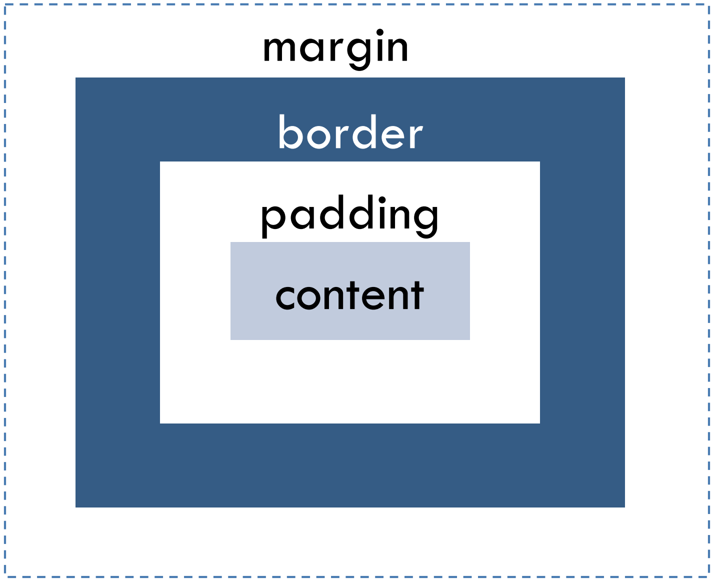
```
<html>
  <head><title>example</title>
    <style>
      body { font-size: 100%; }
      h1 { font-size: 2em; }
      h2 { font-size: 1.5em; }
      em { font-size: 2em; }
    </style>
  </head>
  <body>
      <h1>Heading 1 <em>new</em></h1>
      <h2>Heading 2</h2>
      <p>some normal text</p>
      <p><em>more</em> normal text</p>
  </body>
</html>
```

The default font-size for the body is 16px. The font-size of “Header 1” is 16x2 px. and the font-size of “Header 2” is 16x1.5 px.

What are the font-size of the two <em> elements?

CSS Box Model

Every element is displayed in a box.



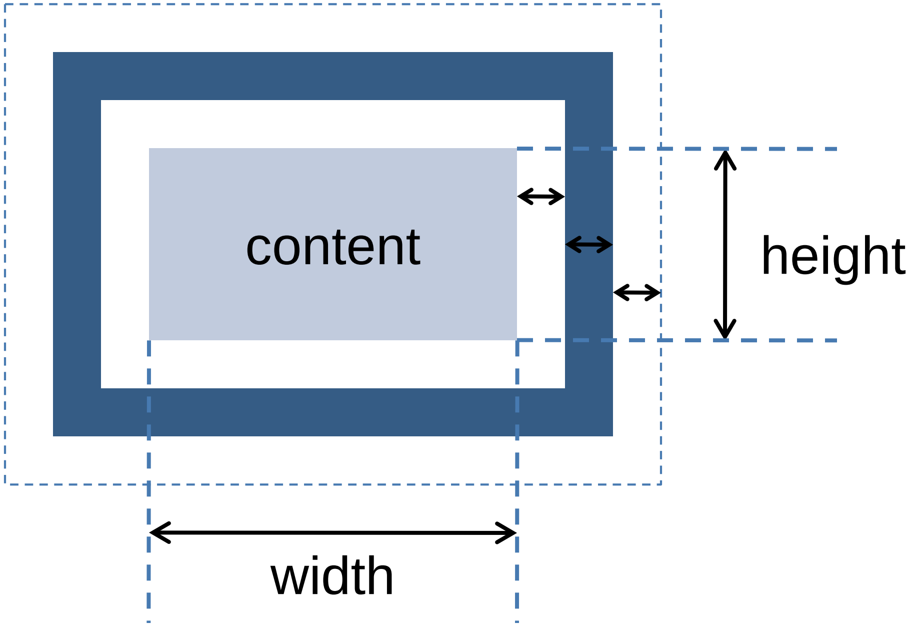
- Element content is shown in the content area.
- Padding is space around the content. It is painted with background color / image.
- Border is a rectangular frame surrounding the padding.
- Margin is transparent space that separates adjacent boxes.

Also check the interactive CSS box model demo at
http://www.markusvogl.com/web1/interactive_box_model/css_box_demo.html

Box dimension

- width and height specify the dimension of the content area
- Other properties configure the size of the padding edge, border edge, and the margin edge
- You can use either relative or absolute length unit

```
p {
   width: 200px;
   height: 100px;
   padding: 20px;
   border-width: 20px;
   margin: 20px;
}
```



Relative length unit

- em – height of the letter 'M' in the current font size, scale in proportion to the current font size
- % –
  - for width, relative to the width of the container box
  - for height, relative to the height of the container box, *if that has a fixed value*
  - for padding and margin, relative to the *width* of the container box
  - N/A for border

```
p {
   width: 80%;
   height: 10em;
   padding: 1em;
   border-width: 0.2em;
   margin: 2em;
}
```

Absolute length unit

- px (pixel) – mainly for screen media
- pt (1 point = 1/72 inch), cm, mm, in
  - Absolute for print media. May still scale up/down for screen media

unit	name	example
px	pixel	border: 1px
pt	point	font-size: 12pt
cm	centimeter	width: 10cm
mm	millimeter	width: 100mm
in	inch	width: 3.94in

More box dimension properties

□ There are shorthand properties to configure the four sides together. We can also configure individual side.

	padding	border	margin
four sides	padding	border-width	margin
one side	padding-top	border-top-width	margin-top
	padding-right	border-right-width	margin-right
	padding-bottom	border-bottom-width	margin-bottom
	padding-left	border-left-width	margin-left

Shorthand properties

- The shorthand properties padding, border-width and margin set the width of the four sides in a single rule.
  - They can take 1 – 4 values. Check the meaning below.

Example	Values apply to
padding: 1em	All sides
padding: 10px 2em	[top & bottom] [left & right]
padding: 1px 5px 2px	[top] [left & right] [bottom]
padding: 10px 10px 1em 1em	[top] [right] [bottom] [left]

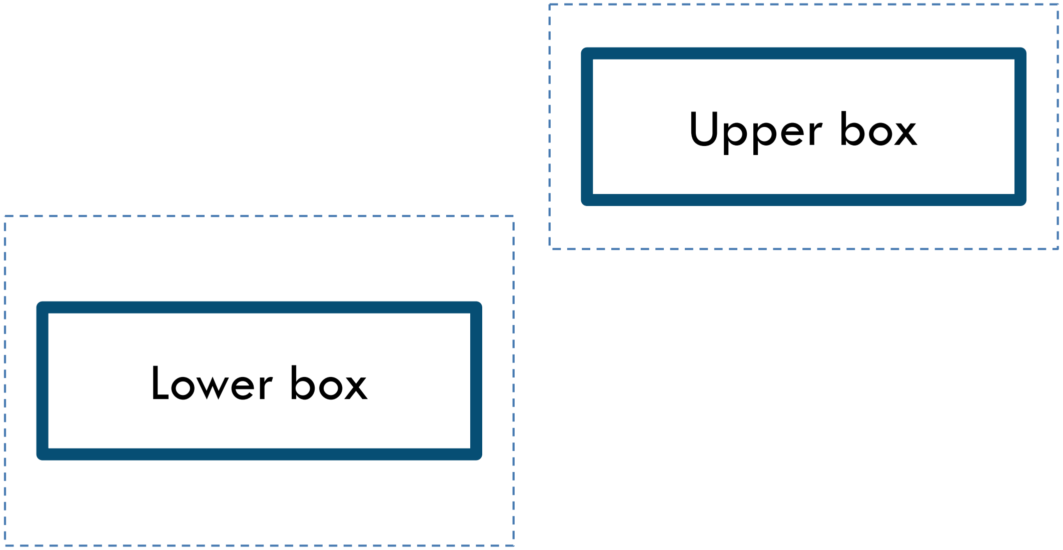
Also check the interactive demo which uses the shorthand notation at http://guyroutledge.github.io/box-model/

Other border properties

- In addition to border-width, we can select the style and color of borders.
  - border-style: solid / dotted / dashed / none, etc
  - border-color
  - They can take 1 to 4 values, with similar meaning as border-width.
  - You can also set border property on a particular side, e.g. border-top-style, border-left-color
- A further shortcut: border: 2px solid black
- No border: border: none *or* border: 0
- Example: http://www.yourhtmlsource.com/stylesheets/cssborders.html

Margin collapsing

- When the margin of a upper box touches the margin of a lower box, the two margins merge.

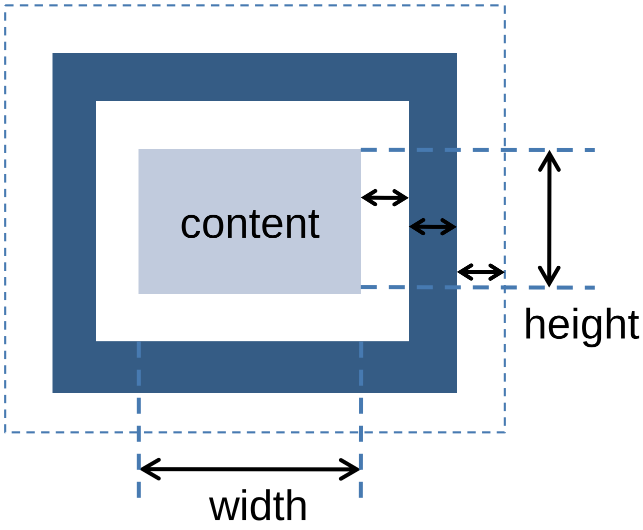


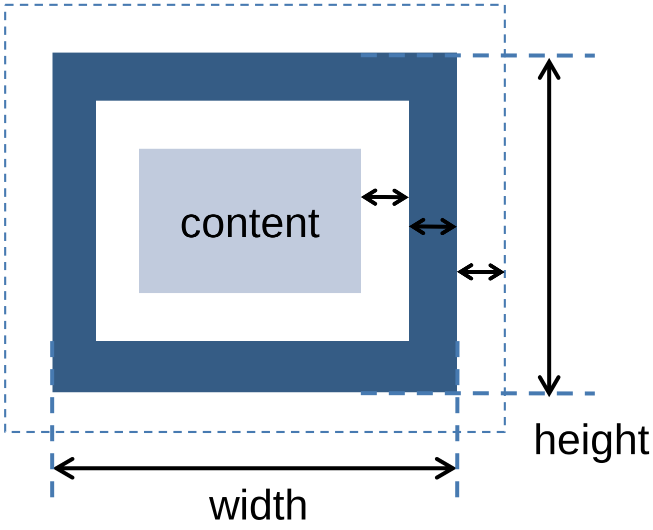
More box properties

- background-color sets the background color
  - Default is transparent
- border-radius makes round-corner box
  - http://www.cssmatic.com/border-radius
- box-shadow supports different kinds of shadow
  - http://westciv.com/tools/boxshadows/index.html
  - http://markusstange.wordpress.com/2009/02/15/fun-with-box-shadows/

Alternative box model

- □ In some situation, it is more convenient to specify the total width / height that includes padding and border
- □ CSS3 defines an alternative box model
  - box-sizing: content-box – width/height include the content area only
  - box-sizing: border-box – width/height also include padding and border





C. Basic CSS selectors

- Selection using extra attribute in HTML
  - id selector and id attribute in HTML
  - class selector and class attribute in HTML
- Based on parent/child relationship in HTML document tree
  - Descendant selectors

Attributes: class & id

- We often need to select one or more elements in a page from CSS and JavaScript ..
  - E.g. change the background color of the first paragraph, or highlight all code in a page with HTML, CSS and JavaScript.
- .. but the type selector selects ALL elements of a given type
  - E.g. p { color: red; } changes the text color of all paragraphs.
- Solution: attribute id and class, and corresponding CSS selectors

```
<p id="summary">
  <code class="attr">id</code> and
  <code class="attr">class</code> are
  used to reference elements.
</p>
```

Id attribute and id selector

- The attribute id provides a unique identifier to an HTML element

```
<p id="first">..</p>
<p>..</p>
<p>..</p>
<p id="last">..</p>
```

- The id selector selects a single element with the id value for formatting in CSS

```
/* CSS style */
#first { border: 1px solid red; }
#last { background-color: green; }
```

More about id selectors

```
#first { border: 1px solid red; }
```

- An id selector selects a single element in an HTML doc by its 'id'
- For clarity, you can also include the element type.
  - p#first { border: 1px solid red; }
- You SHOULD NOT assign the same id value to more than one elements
  - Although most browsers tolerate this error in CSS, you lose the ability to select the elements uniquely in JavaScript.

Class attribute and class selector

- The attribute class adds a label to some HTML elements
  - More than one elements may belong to the same class

```
<p >..</p>
<p class="revised">..</p>
<p>..</p>
<p class="revised">..</p>
```

- The class selector selects all elements that belong to the class for formatting in CSS

```
/* CSS style */
.revised { border-left: 3px solid red; }
```

More about class selectors

```
p.revised { border-left: 3px solid red; }
```

- A class selector with element type matches elements in the specified type that also belong to the class.
  - p.revised { border-left: 3px solid red; }
- A class selector without element type matches an element (of any type) in the class.
  - .revised { color: red; }

Example

```
 .hilite { color: red; }
 p.hilite { border: 1px solid red; }

<p>blah blah blah …</p>
<p>Remember:
   <q class="hilite">Keep it short and simple</q>
   blah blah blah …
</p>
<p class="hilite">An important paragraph to highlight …</p>
<p>blah blah blah …</p>
```

An element in several classes

- An element can also belong to more than one class
  - List the classes in one class attribute, separate them by space.

```
<p class="revised" >paragraph one</p>
<p class="revised hilite" >paragraph two</p>
<p class="hilite" >paragraph three</p>
<p>paragraph four</p>
```

```
<!-- THIS IS INCORRECT -->
<p class="revised" class="hilite">paragraph two</p>
```

Select elements in multiple classes

- A class selector may specify several classes
  - Syntax: p.class1.class2
  - Selects <p> elements that belong to both class1 and class2

```
p.revised.hilite { color: red; }
p.hilite { font-style: italic; }
p.revised { background-color: yellow; }

<p class="revised" >paragraph one</p>
<p class="revised hilite" >paragraph two</p>
<p class= "hilite" >paragraph three</p>
<p>paragraph four</p>
```

Descendant selectors

```
h1 em { color: red; }
```

- A descendant selector consists of a list of simple selectors separated by spaces.
  - Simple selectors: type, id and class selector
- The selector of the form ‘A B’ matches an element B that is a descendant of an element A.
  - E.g. em inside level 1 heading

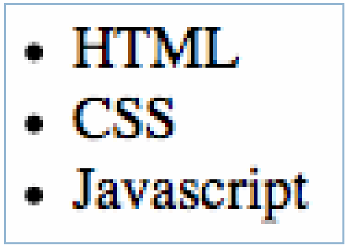
D. Lists

- *A list of related items*
- Three types:
  - Unordered list: <ul> <li>
  - Ordered list: <ol> <li>
  - Definition list: <dl> <dd> <dt>
- List styles
  - Text styles
  - Box model
  - Special CSS properties: list-style …

Unordered list

- Unordered list <ul> is a block element
- <ul> contains some <li> (list item)
- Each <li> may contain text, inline elements and block elements
- You can customize the bullets by CSS

```
<ul>
  <li>HTML</li>
  <li>CSS</li>
  <li>Javascript</li>
</ul>
```

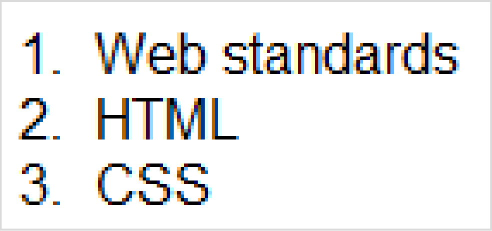


- HTML
- CSS
- Javascript

Ordered list

- Ordered list <ol> is a block element
- <ol> contains some <li>, similar to <ul>
- You can customize the numbering by CSS

```
<ol>
   <li>Web standards</li>
   <li>HTML</li>
   <li>CSS</li>
</ol>
```

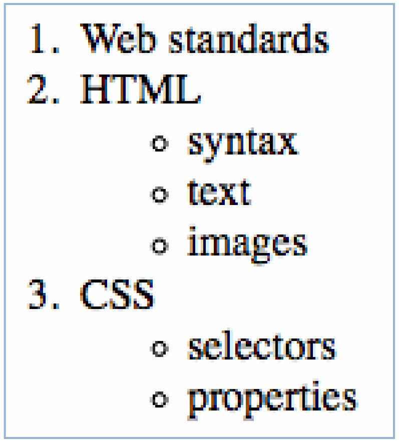


```
1.  Web standards
2.  HTML
3.  CSS
```

Nested lists

- <li> can contain block elements, even another list.
- Notice how each level adds some indentation

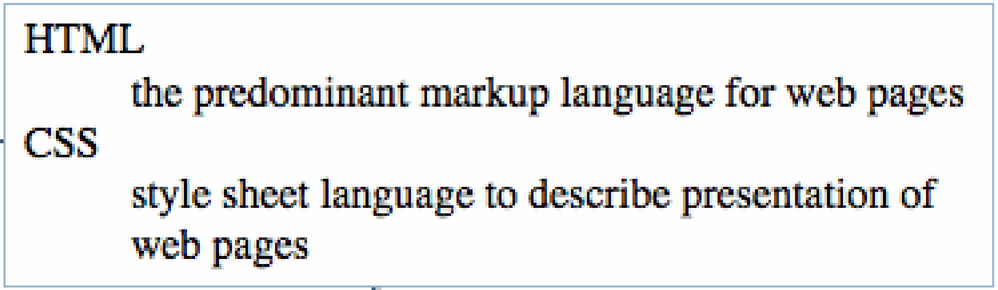
```
<ol>
   <li>Web standards</li>
   <li>HTML
        <ul><li>syntax</li>
              <li>text</li><li>images</i>
        </ul>
   </li>
   <li>CSS
       <ul><li>selectors</li><li>properties</li></ul>
   </li>
</ol>
```



1. Web standards
2. HTML
   - syntax
   - text
   - images
3. CSS
   - selectors
   - properties

Definition List

- <dl> is a block element, which contains several items. Each item is composed of
  - a term <dt>
  - an explanation <dd>



HTML

the predominant markup language for web pages

CSS

style sheet language to describe presentation of web pages

```
<dl>
  <dt>HTML</dt>
  <dd>the predominant markup ...</dd>
  <dt>CSS</dt>
  <dd>style sheet language ...</dd>
</dl>
```

Styling a list

- Indentation is controlled by padding / margin of <li>
  - Exercise: check formatting of nested lists
- Bullet format follows font-size and color of <li>
- Several list-specific CSS properties:
  - list-style-type: none, disc, circle, square, decimal, upper-roman, lower-roman, upper-alpha, lower-alpha
  - list-style-image: url( ) uses images as bullets
  - list-style-position: inside places the bullet inside the box of a list item
  - Shorthand list-style: none url(arrow.gif) inside

Selecting items in a nested list

- How can we select only second level list items?
  - Type selector li { ... } selects both level 1 and level 2 list items.
  - We only want to select <li> that are inside a <ul>…

```
<ol>
  <li>Web standards</li>
  <li>HTML
      <ul><li>syntax</li>…</ul>
  </li>
  <li>CSS
       <ul><li>selectors</li>…</ul>
  </li>
</ol>
```

Example

```
<style>
  ul li { color: blue; }
  li { color: green; }
</style>
…
<ol>
   <li>Web standards</li>
   <li>HTML
        <ul><li>syntax</li>…</ul>
   </li>
   <li>CSS
         <ul><li>selectors</li>…</ul>
   </li>
</ol>
```

Both style rules match the two 2nd level <li>. However, the first rule is more *specific* and so it wins.

Conflicting style declarations

- Several style rules that match an element may specify conflicting values for a property

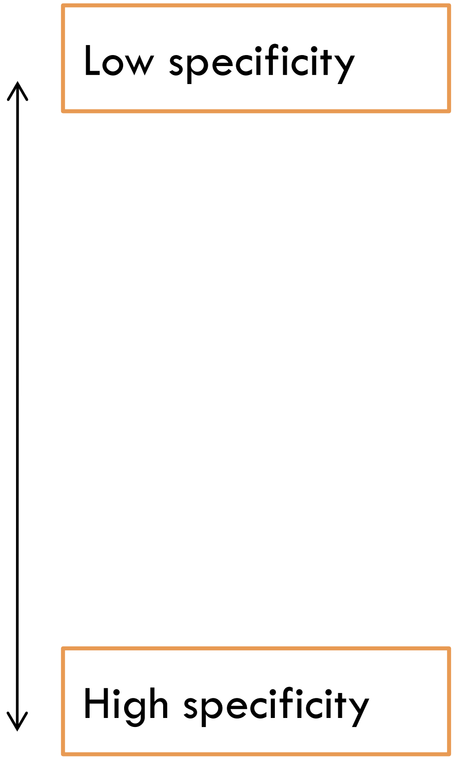
```
li { color: red; }
li li { color: green; }
li.hilte { color: blue; }

<ul>
   <li>one</li>
   <li>two:
      <ul><li>a</li><li>b</li>
         <li class="hilte">c</li></ul>
   </li>
   <li>three</li>
</ul>
```

Specificity

- Specificity of a CSS selector is a list of three numbers a-b-c
  - a is the number of ID selectors
  - b is the total number of class selectors
  - c is the total number of type selectors
  - (note: this is a simplification.. More detail in chap 5)
- When there are more than one applicable style declaration, use the ones with the highest specificity
  - Example: suppose selector 1 has specificity a1-b1-c1 and selector2 has a2-b2-c2
  - If a1>a2, selector 1 wins. If a2>a1, selector 2 wins.
  - If a1=a2, compare b1 and b2. If b1>b2, selector 1 wins. If b2>b1, selector b2 wins.
  - If a1=a2 and b1=b2, compare c1 and c2. The higher wins.

Example

Selectors	specificity
li	0-0-1
ol li	0-0-2
li li li	0-0-3
li.inner	0-1-1
ol.outer li	0-1-2
ol.outer li.important	0-2-2
li#last	1-0-1
ol#toc li	1-0-2
ol#toc li.important	1-1-2



Low specificity

High specificity

Example

- □ Determine the color of text in the nested list

```
li { color: red; }
li li { color: green; }
li.hilte { color: blue; }

<ul>
    <li>one</li>
    <li>two:
         <ul><li>a</li><li>b</li>
              <li class="hilte">c</li></ul>
    </li>
    <li>three</li>
</ul>
```