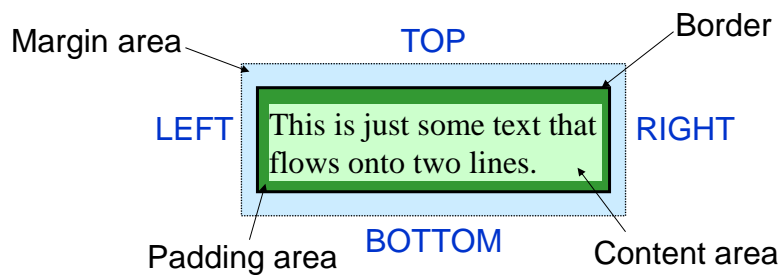


CSCU9B2: Web Tech Lecture 3

MORE HTML & CSS

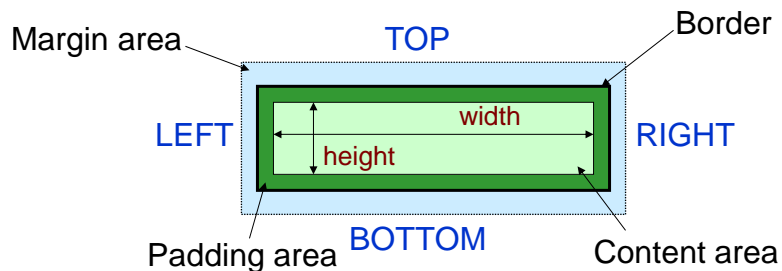
The Box Model

- Every HTML element (e.g. `h2`, `p` etc) lies within a virtual box:



Box Sizing

- By default, width and height specifications apply to the content area
- Property “box-sizing: border-box;” changes this to area including the border

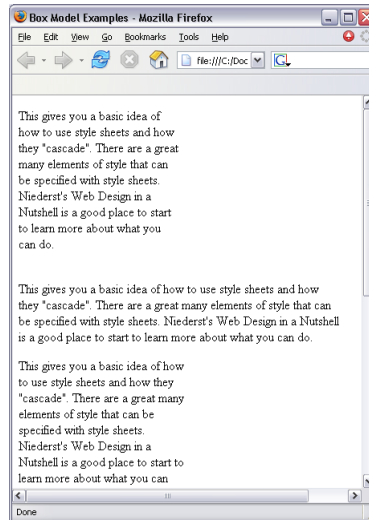


Width and Height

- Width and height can be specified for
 - Block-level elements e.g. `h1`, `p`
 - Inline replaced elements e.g. `img`
 - But not for inline text
- Absolute or percentage specifications
 - Percentage is relative to parent element
- Usually only specify width
 - Height required to fit contents calculated by browser
 - Width and height useful for images

Width and Height Examples

```
p.narrow { width: 200px;
           height: 200px; }
p.wide { width: 400px; }
p.halfpage { width: 50%; }
```



Borders

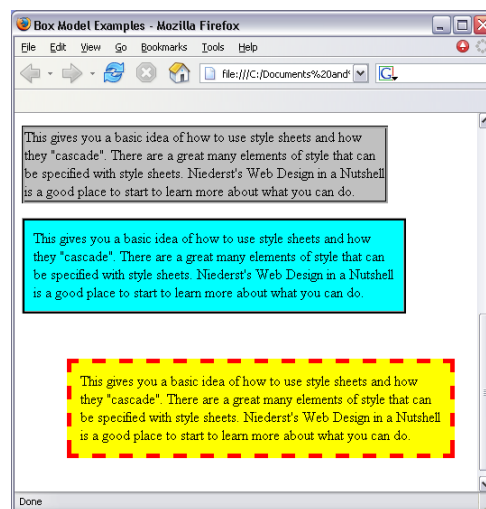
- e.g. `div { border-style: dashed; border-width: thick; border-color: gray; }`
- Shorthand: e.g. `div { border: grooved thin black; }`
- Can target individual sides: e.g. `div { border-top: 6px solid green; }`
- Border style is required; thickness and colour are optional

Margins and Padding

- e.g. `p { padding: 10px; margin: 30px; }`
- Can target individual sides: e.g. `div { padding-top: 6px; margin-left: 3em; }`
- Can set all sides to different values: e.g. `p { margin: 3px 20px 2px 20px; }`
 - Ordering is: top, right, bottom, left

Examples

```
p.bord { width: 400px;
background-color: silver;
border-style: groove; }
p.bordpad { width: 400px;
background-color: aqua;
border-style: ridge;
padding: 10px; }
p.bordmarg { width: 400px;
background-color: yellow;
border: dashed red thick;
padding: 10px;
margin: 50px; }
```



Background Colour and Images

```
body { background-image: url(clouds.jpg); }
p.back { width: 200px; height: 200px;
  background-color: aqua; border-style: ridge; }
p.backimg { width: 400px;
  background-image: url(backgr1.gif); }
```

to use style sheets and how they "casc
style that can be specified with style s
It is a good place to start to learn mor

they "cascade". There are a great
many elements of style that can
be specified with style sheets.
Niederst's Web Design in a
Nutshell is a good place to start
to learn more about what you
can do.

This gives you a basic idea of how to use style sheets and how
they "cascade". There are a great many elements of style that can
be specified with style sheets. Niederst's Web Design in a
Nutshell is a good place to start to learn more about what you can do.

Done

Measurement Units

CODE	UNIT
px	Pixel
pt	Point
pc	Pica
em	Em
ex	EX
in	Inches
mm	Millimeters
cm	centimeters

Positioning: Aligning

- Text aligned with `text-align` style
 - left, right, center, justify
- Block elements aligned using margins
 - E.g. right align: `div { margin-left: auto; margin-right: 0px; width: 200px; }`
 - E.g. centering: `p { margin-left: auto; margin-right: auto; width: 200px; }`

Aligning Examples

```
.txtcentre { text-align: center; }  
div.right { width: 300px;  
            margin-left: auto;  
            margin-right: 0px; }  
p.centre { width: 300px;  
           text-align: justify;  
           margin-left: auto;  
           margin-right: auto; }
```

Alignment Examples

Just a Heading

This gives you a basic idea of how to use style sheets and how they "cascade". There are a great many elements of style that can be specified with style sheets. Niederst's Web Design in a Nutshell is a good place to start to learn more about what you can do.

This gives you a basic idea of how to use style sheets and how they "cascade". There are a great many elements of style that can be specified with style sheets. Niederst's Web Design in a Nutshell is a good place to start to learn more about what you can do.

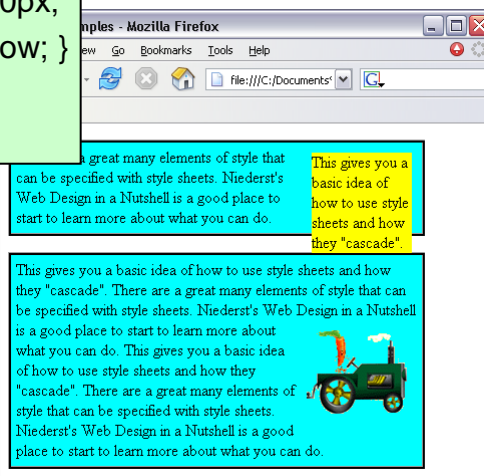
Positioning: Floating

- Normal flow
 - Block-level elements stack on top of each other
 - Inline elements fill the available space in their block
- Can float both block and inline elements
 - Element floats to left or right edge of content area of containing block
 - Text flows around floated element
 - E.g. `div { float: left; }`
 - E.g. `img { float: right; }`

Floating Examples

```
span.goright { float: right;  
margin: 5px; width: 100px;  
background-color: yellow; }
```

```
img.rightside { float: right;  
margin: 10px; }
```



Positioning: Specific

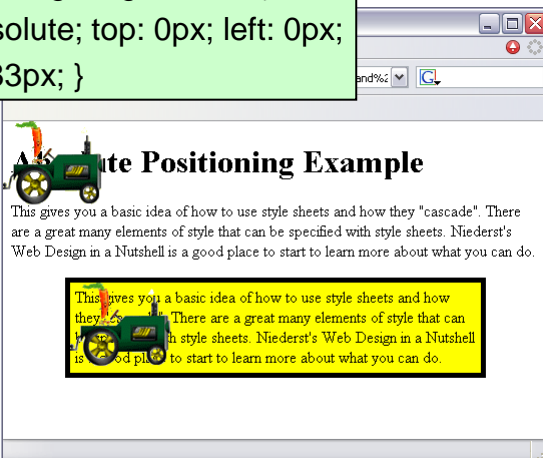
- Element removed from normal flow
- Given specific position
 - Relative to containing block
 - Containing block must be explicitly positioned
 - Top level container is the web page body
- Types of specific position
 - relative or absolute
- Specifying position by offset
 - top, right, bottom, left

Positioning Examples

```
p.contain { position: relative; width: 400px;
margin-left: auto; margin-right: auto; }
img.topleft { position: absolute; top: 0px; left: 0px;
width: 100px; height: 83px; }
```

```
<p>
This gives you a basic idea of how to use style sheets and how they "cascade". There
are a great many elements of style that can be specified with style sheets. Niederst's
Web Design in a Nutshell is a good place to start to learn more about what you can do.
</p>
<p class="contain">

This gives you a basic idea of how to use style sheets and how they "cascade". There
are a great many elements of style that can be specified with style sheets. Niederst's
Web Design in a Nutshell is a good place to start to learn more about what you can do.
</p>
```



More HTML

Semantic Tags

What are Semantic Tags?

- Semantic tags are elements with meaning
- Examples of non-semantic elements:
 - `<div>` and `` - say nothing about the content.
- Examples of semantic elements:
 - `<h1>`, `<p>`, `<table>` - clearly define the type of their structure and content.

Limits of HTML pre HTML5

- Many web pages and documents have semantic elements that are not represented by HTML tags before HTML5:
 - header, footer, footnote, aside
- Many web sites contain HTML code like:

```
<div id="nav"> ... </div>
<div class="header"> ... </div>
<div class="footer"> ... </div>
```

to indicate navigation (links), header, and footer.

New Semantic Tags in HTML5

- HTML5 offers new semantic elements to define different parts of a web page:
<article>, <aside>, <details>, <figcaption>, <figure>, <footer>, <header>, <main>, <mark>, <nav>, <section>, <summary>, <time>
- They may or may not affect the visual appearance of a page, but the semantics allows better searching and style application.

Header <header>

- This element represents a container for introductory content.
- It may contain:
 - Section's headings
 - Section's table of contents
 - A search form
 - Bylines
 - Logos

```
<article>
  <header>
    <h1>Heading here</h1>
    <h3>Subheading here</h3>
    <p>More info here</p>
  </header>
  <p>Lorem Ipsum....</p>
</article>
```

Footer <footer>

- The **<footer>** element specifies a footer for a document or section.
- A **<footer>** element typically contains the author of the document, copyright information, links to terms of use, contact information, etc.

```
<footer>
  <p>Posted by: Fred Bloggs</p>
  <p>Contact information: Planet Earth</p>
</footer>
```

Navigation <nav>

- The **<nav>** element defines a set of navigation links.
- Not all groups of links on a web page need to be contained within a **<nav>** element;
- The element is primarily intended for sections that consist of major navigation blocks;
- The element can appear more than once on any given web page.
- Typically, the **<nav>** element can be found within the **<header>** structural element.

Sectional Elements

- **<article>**: specifies independent, self-contained content. Potential sources for the **<article>** element are news story, blog posts, forum posts.
- **<section>**: defines a section in a document. Several examples include chapters of a book or numbered sections of a thesis.
- **<aside>**: defines some content aside from the content it is placed in (like a sidebar). The aside content should be related to the surrounding content.
- **<hgroup>**: group any set of **<h1 - h6>** elements when the heading has multiple levels.

Example

- <http://www.cs.stir.ac.uk/courses/CSCU9B2/resources/HTMLExamples/cake-extras.html>

End of Lecture

Next Web Tech lecture on JavaScript