The Box Model

WEB DEVELOPMENT FUNDAMENTALS

Introduction

- Understanding how to control a block's width and height
  - Floats and Clears
- Positioning elements out of the normal document position:
  - Relative
  - Absolute
  - Fixed
- Controlling Element visibility & dimensions
  - Min & max heights
  - Creating resizable elements
  - Hidden, display and visibility

2

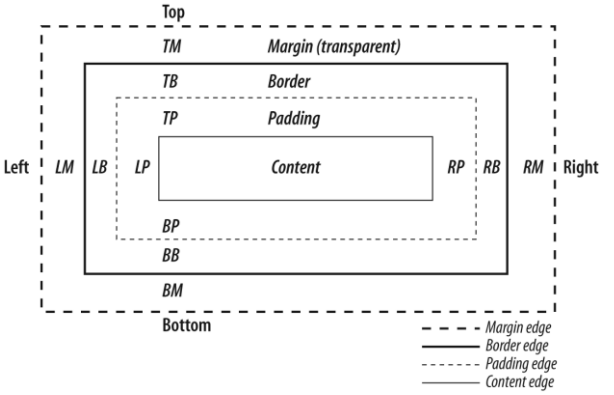
Containing blocks

- Containing block: the box in which an element lives
  - Defined differently depending on the position property
  - All block level elements are containing blocks
- Block level elements are static positioned
  - Meaning they are in natural flow
  - They can have their positioning changed.
- A CSS block level element is positioned by the Box Model
  - CSS3 adds a new model but we will consider the CSS2.1 model first
- A block level element takes up space in a layout controlled by its:
  - Height
  - Width
  - Margin
  - Padding

3

The Standard Box Model

- The standard box model is known as a `content-box where:`
  - width + padding + border = actual visible/rendered width of box
  - height + padding + border = actual visible/rendered height of box



4

Widths & Heights

- A browser by default sets the width and height of an element
  - The width defined by its parent's box-model
  - The height set by the required content

```
article>div{width: 50%; height: 450px;}
```

- The width or height can use any measurement unit
  - Also be set to auto – then calculated by browser
- Altering the width of an element reduces its available content area
  - But does not free up the space available
  - The 'gutter' of the block element remains filled
  - Block elements appear in their normal position and flow

5

Float and Clear

- Float will move an element and flow text around it
  - Treats the element as a block element and moves it left/right
  - Rest of the page flows around the floated element
    - The available box is shrunk by the amount the floats take up
- Clear will move an element to after the float
  - Adds clearance to the top margin to move it clear of the float
    - Moves top border edge below the bottom outer edge of the float
    - Unless the cleared element is also a float (line up outer edges)
- See
  - http://www.w3.org/TR/CSS21/visuren.html#propdef-float
  - http://www.w3.org/TR/CSS21/visuren.html#propdef-clear

6

Positioning Elements

- Position: relative | static
  - The content edge of the nearest block-level ancestor
- Position: absolute
  - The nearest positioned ancestor according to
    - The padding edge of the if the ancestor is block-level
    - The content edge of the first/last box if the ancestor is inline
- Position: fixed
  - The window/printed page

7

Relative positioning

- Relative positioning: offset from default position
  - I.e. moved from where it would have been
  - Offset not measured from containing block
- Next element flows as if the box hadn't been moved
  - Relative boxes take up space where they would have been
- Moved element has same size, as if it hadn't been moved
  - Hence specify only one of left/right and top/bottom
    - E.g. if you specify left and right this could change the width of the element, which is not allowed, hence one of left/right will be ignored
- See:
  - http://www.w3.org/TR/CSS21/visuren.html#relative-positioning

8

Absolute positioning

- Absolute positioning: offset from container's position
  - I.e. relative to container not page
- Offset measured from
  - Block level ancestor: the top, left of ancestor's padding box
    - I.e. outside of padding, inside of border
  - Inline ancestor: the top, left of the ancestor's content box
    - I.e. outside of content
- See:
  - http://www.w3.org/TR/CSS21/visuren.html#position-props
  - http://www.w3.org/TR/CSS21/visuren.html#absolutely-positioned

9

Overflow, Min & Max dimensions

- The width and height of an object can be constrained
  - With min-height/min-width and max-height/max-width
  - Once set, an element will never grow/shrink beyond these values
- The element is now smaller than the content it display
  - What happens to this content can be controlled with the overflow
  - Can be set to:
    - auto
    - visible
    - hidden
  - CSS3 allows overflow control on a specific axis overflow-x/y
  - In CSS3, we also have the hidden property

10

Controlling how an element displays

- Elements are primarily set to be block or inline as their display type
  - This behaviour can be changed in CSS
  - By modifying the display attribute
  - By setting an element to be display:none an element is hidden
    - The element is then removed from the flow
    - Can be accomplished with a hidden attribute in HTML5
    - Alternatively, there is the visible property
      - Does not remove the element from the document flow
- Elements can also be switched between inline and block display
  - Useful for advanced layout, as you will see in the lab

11

The border-box model

- The broken box model is a familiar tale of woe to most
  - CSS3 includes a new attribute called box-sizing
  - Set to content-box to get the traditional W3C box model

```
article { box-sizing: content-box;}
```

  - The total width of the element will be:
    - the width set on the element
    - plus the width of the borders and padding
  - If border-box borders and paddings include in the width

```
article { box-sizing: border-box;}
```

12

Review

- Positioning gives us another string in our bow, when working with the Box Model
  - It is a complementary tool
- Laying out pages with positioning and box models is the basis of contemporary design

13