The CSS Box Model

CSE 190 M (Web Programming), Spring 2008
University of Washington
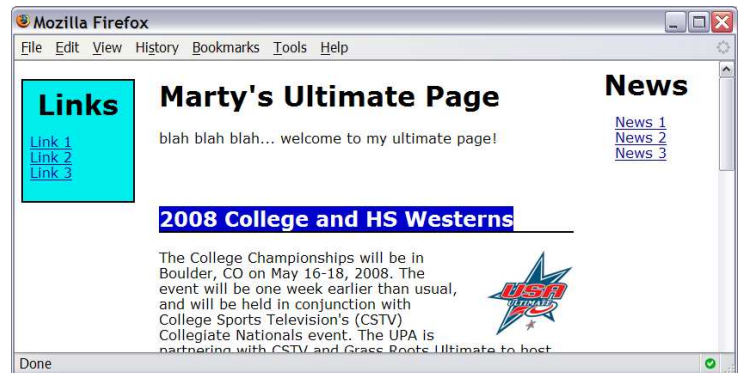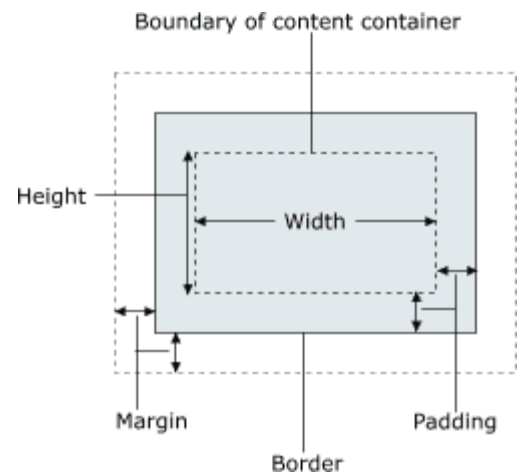
Reading: Chapter 2, sections

Motivation

- want to be able to style individual elements, groups of elements, sections of text or of the page
- want to create **complex page layouts**

CSS Box Model

- for layout purposes, every element is composed of:
    - the actual element's **content**
    - a **border** around the element
    - **padding** between the content and the border (*inside*)
    - a **margin** between the border and other content (*outside*)
- width = content width + L/R padding + L/R border + L/R margin
  height = content height + T/B padding + T/B border + T/B margin
    - IE6 doesn't do this right, and sucks

Document flow - block elements

```
<html>
    <head>...</head>
    <body>
        <h1>...</h1>
        <h2>...</h2>
        <p>...</p>
        <h2>...</h2>
        <p>...</p>
        <p>...</p>
        <p>...</p>
    </body>
</html>
```
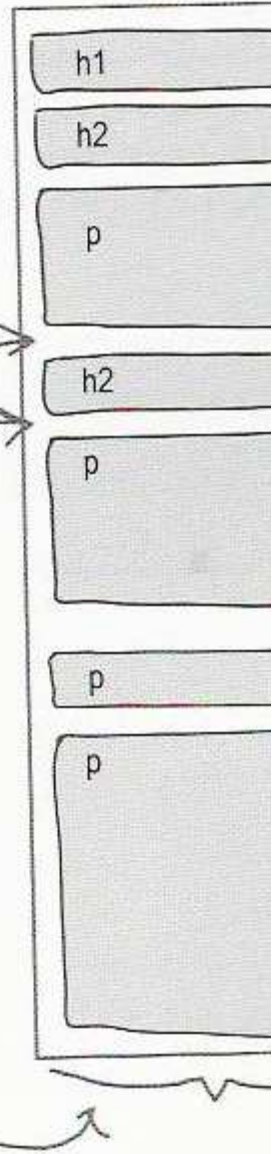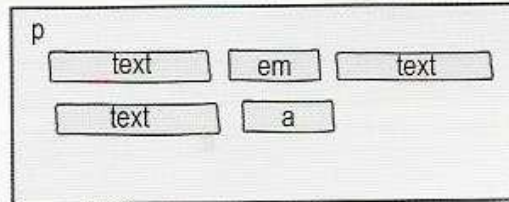
Each block element is taken in the order it appears in the markup, and placed on the page.

| h1 |
| h2 |
| p |

Each new block element causes a linebreak.

| h2 |
| p |

| p |
| p |

Notice that elements take up the full width of the page.

Document flow - inline elements

```
<p>
Join us <em>any evening</em> for
these and all our other wonderful <a
href="beverages/elixir.html" title="Head
First Lounge Elixirs">elixirs</a>.
</p>
```

```
p
  [ text ]  [ em ]  [ text ]
  [ text ]  [ a ]
```

Document flow - a larger example

```
h1  [      text      ]
    [ text ]

h2  [      text      ]

p   [ span ] [ em ] [ span ]
    [ em ]

p   [ img ]    [ img ]
    [ img ]    [ img ]
```
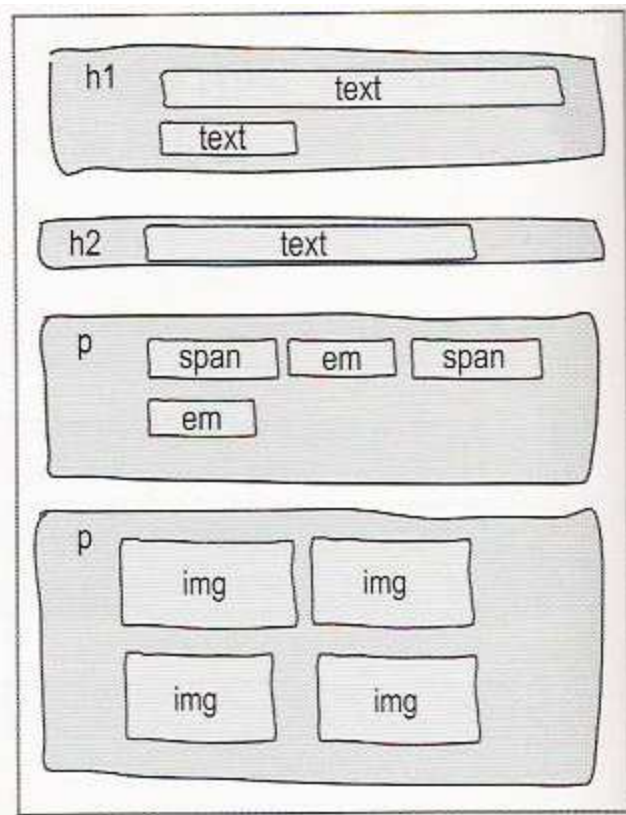
CSS properties for borders

```css
h2 { border: 5px solid red; }                                    CSS
```

This is a heading.

- `border`: all properties of border on all 4 sides
- a border is specified as three items:
  - its **thickness** (specified in `px`, `pt`, `em`, or one of the following general widths: `thin`, `medium`, `thick`)
  - its **style** (one of `none`, `hidden`, `dotted`, `dashed`, `double`, `groove`, `inset`, `outset`, `ridge`, `solid`)
  - its **color** (specified as seen previously for text and background colors)

More border properties

- `border-color`, `border-width`, `border-style`: specific properties of border on all 4 sides
- `border-bottom`, `border-left`, `border-right`, `border-top`: all properties of border on a particular side
- `border-bottom-color`, `border-bottom-style`, `border-bottom-width`, `border-left-color`, `border-left-style`, `border-left-width`, `border-right-color`, `border-right-style`, `border-right-width`, `border-top-color`, `border-top-style`, `border-top-width`: specific properties of border on a particular side
- Complete list of border properties

Border example 2

```css
h2 {
  border-left: thick dotted #CC0088;
  border-bottom-color: rgb(0, 128, 128);
  border-bottom-style: double;
}                                                                CSS
```

This is a heading.

- each side's border properties can be set individually
- if you omit some properties, they receive default values (e.g. `border-bottom-width` above)

CSS properties for padding

- <u>padding</u>: padding on all 4 sides
- <u>padding-bottom</u>: padding on bottom side only
- <u>padding-left</u>: padding on left side only
- <u>padding-right</u>: padding on right side only
- <u>padding-top</u>: padding on top side only
- <u>Complete list of padding properties</u>

Padding example 1

```css
p { padding: 20px; border: 3px solid black; }
h2 { padding: 0px; background-color: yellow; }
```
CSS

This is the first paragraph

This is the second paragraph

This is a heading

Padding example 2

```css
p {
  padding-left: 200px; padding-top: 30px;
  background-color: fuchsia;
}
```
CSS

This is the first paragraph

This is the second paragraph

- each side's padding can be set individually
- notice that padding shares the background color of the element

CSS properties for margins

- <u>margin</u>: margin on all 4 sides
- <u>margin-bottom</u>: margin on bottom side only
- <u>margin-left</u>: margin on left side only
- <u>margin-right</u>: margin on right side only
- <u>margin-top</u>: margin on top side only
- <u>Complete list of margin properties</u>

Margin example 1

```css
p {
  margin: 70px;
  background-color: fuchsia;
}
```
CSS

This is the first paragraph

This is the second paragraph

- notice that margins are always transparent
  (they don't contain the element's background color, etc.)

Margin example 2

```css
p {
  margin-left: 200px;
  background-color: fuchsia;
}
```
CSS

This is the first paragraph

This is the second paragraph

- each side's margin can be set individually

CSS properties for dimensions

```css
p { width: 400px; background-color: yellow; }
h2 { width: 50%; background-color: aqua; }
```
CSS

This paragraph uses the first style above.

This heading uses the second style above.

- `width`, `height`: how wide or tall to make this element
- `max-width`, `max-height`, `min-width`, `min-height`: the maximum or minimum size of this element in the given dimension
- all of these apply only to block elements; ignored for inline elements
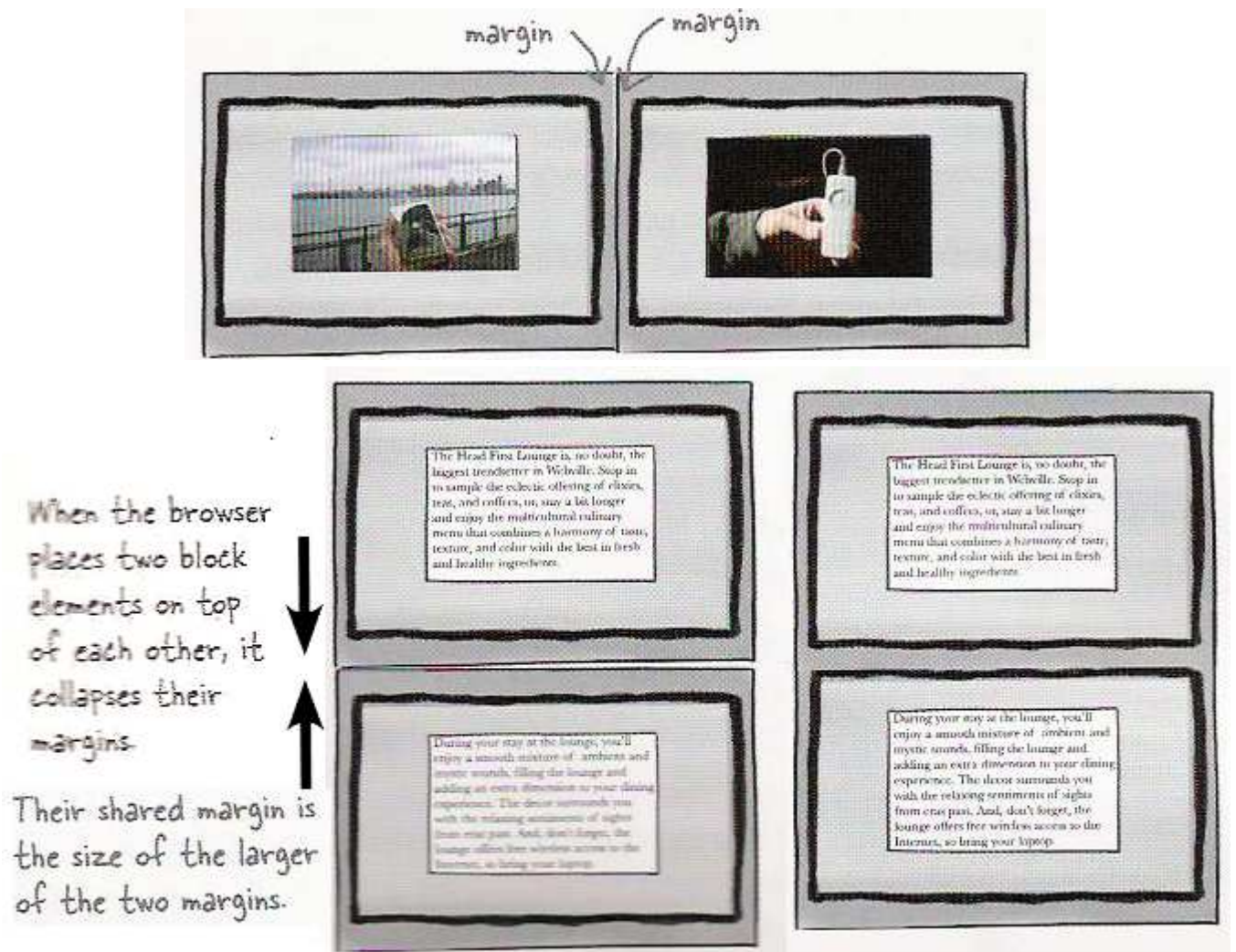
Centering a block element: `auto` margins

```css
p {
  margin-left: auto; margin-right: auto;
  width: 750px;
}
```
CSS

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore mag
aliqua.

- only works if `width` is set
  (otherwise, element occupies entire width of page)
- to center inline elements within a block element,
  use `text-align: center;` instead

Top/bottom margin collapse

When the browser places two block elements on top of each other, it collapses their margins.

Their shared margin is the size of the larger of the two margins.

- when two block elements appear on top of each other, their margins are collapsed
- their shared margin is the larger of the two individual margins

Floating elements

The CSS `float` and `clear` properties

The CSS <u>float</u> property (<u>reference</u>)

```css
img.headericon {  float: right; width: 130px;  }
```
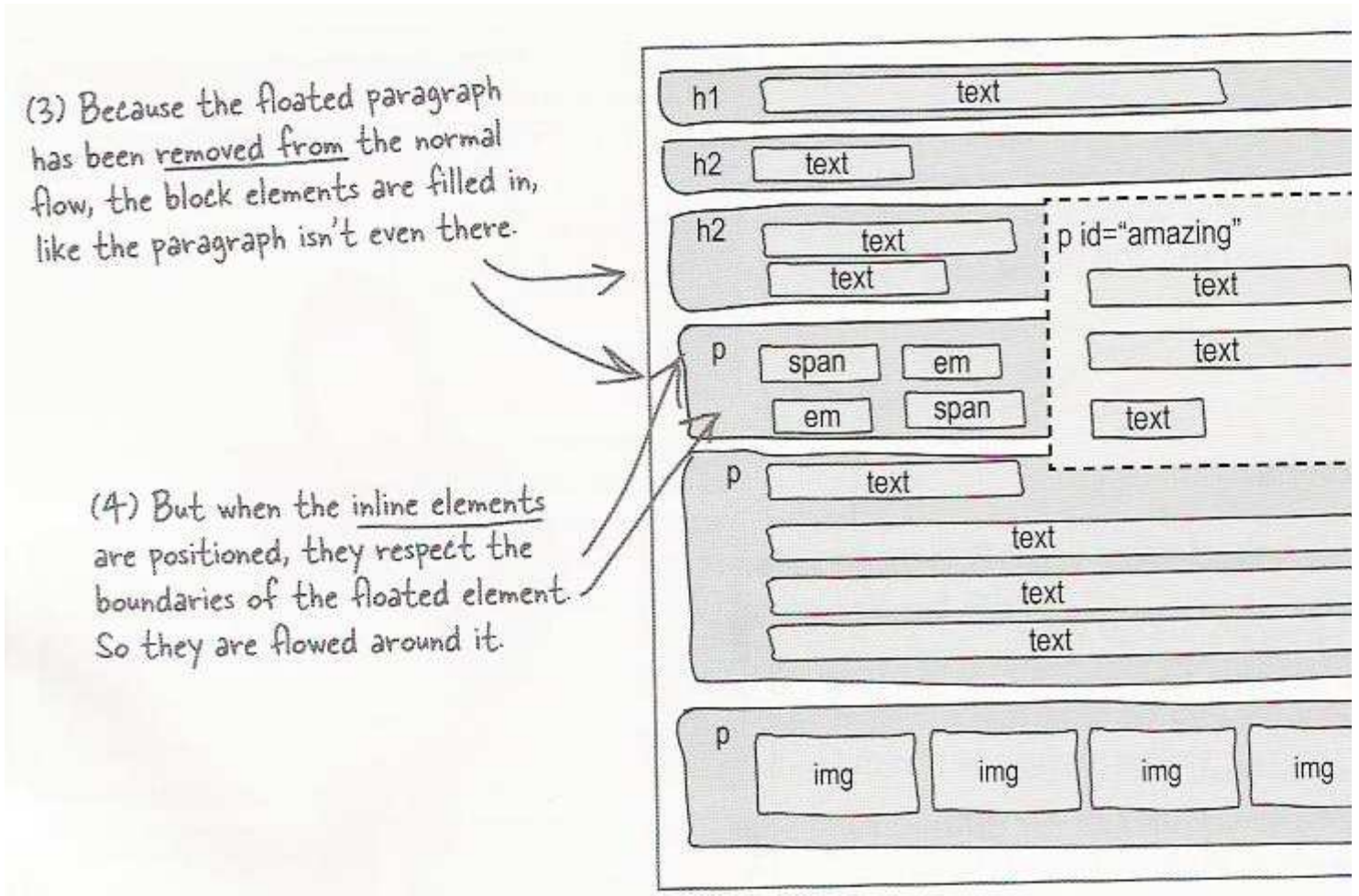*CSS*

<u>Borat</u> Sagdiyev (born July 30, 1972) is a fictional Kazakhstani journalist played by British-Jewish comedian Sacha Baron Cohen. He is the main character portrayed in the controversial and successful film Borat: Cultural Learnings of America for Make Benefit Glorious Nation of Kazakhstan. Borat ...

- `float` can be `left`, `right`, or `none` (default)
- floating elements are removed from normal document flow
- underlying text wraps around floating element as necessary

Floating elements diagram

(3) Because the floated paragraph has been underlined removed from the normal flow, the block elements are filled in, like the paragraph isn't even there.

(4) But when the inline elements are positioned, they respect the boundaries of the floated element. So they are flowed around it.

Common `float` bug: missing width

I am not floating, no width

I am floating right, no width

I am not floating, 45% width

I am floating right, 45% width

- often floating block elements must have a `width` property value
  - if no `width` is specified, the floating element may occupy 100% of the page width, so no content can wrap around it

The `clear` property

```css
p { background-color: fuchsia; }
h2 { clear: right; background-color: yellow; }
```
CSS

Homestar Runner is a Flash animated Internet cartoon. It mixes surreal humour with references to 1980s and 1990s pop culture, notably video games, classic television and popular music.
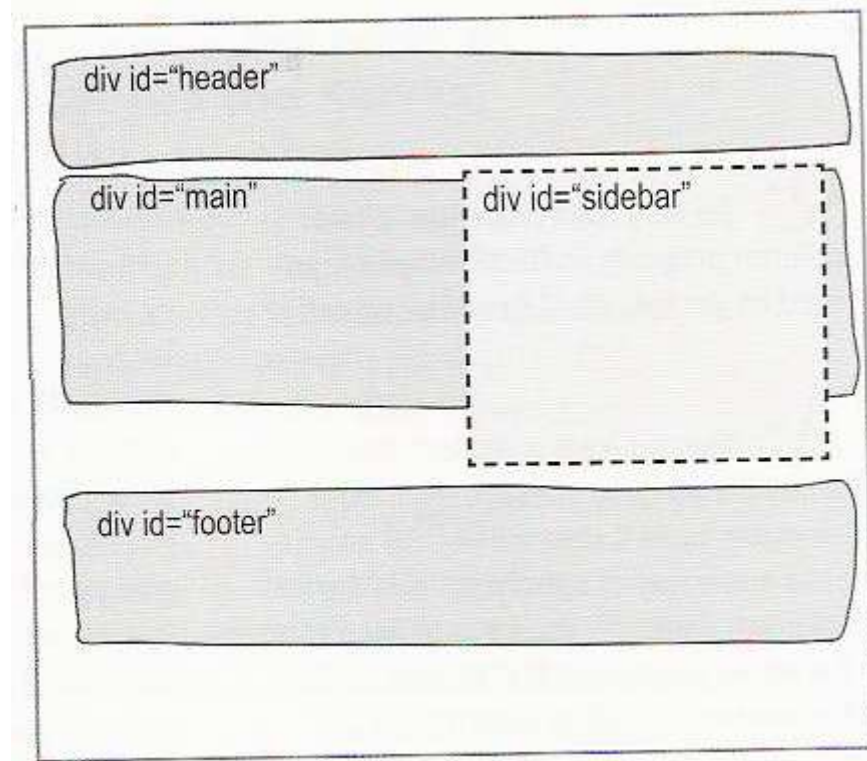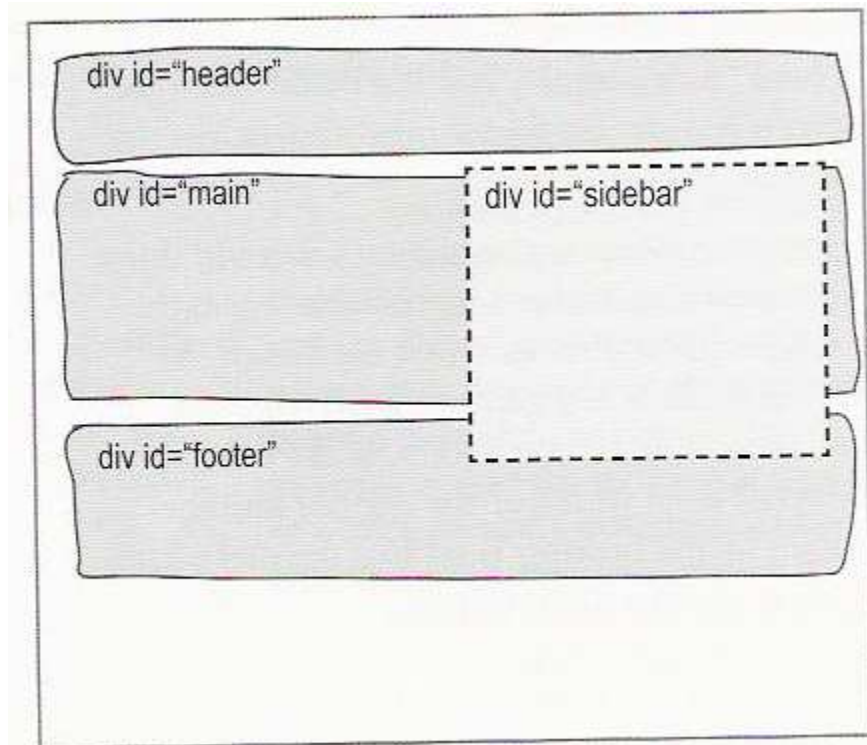
My Homestar Runner Fan Site

- disallows any floating elements from overlapping this element
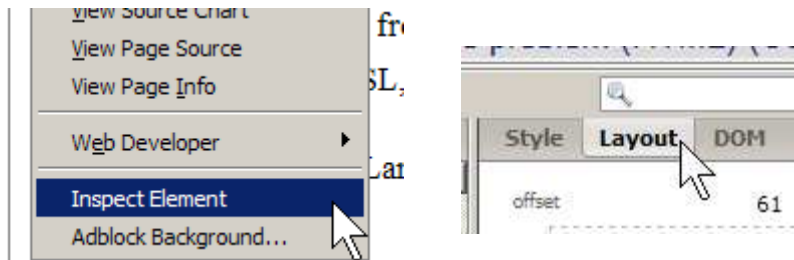- `clear` can be `left`, `right`, `both`, or `none` (default)

Clear diagram

```css
div#sidebar { float: right; }
div#footer { clear: right; }
```

CSS

div id="header"

div id="main"

div id="sidebar"

div id="footer"

div id="header"

div id="main"

div id="sidebar"

div id="footer"

Recall: Firefox Firebug extension