# CSS, part 1

CS147L Lecture 2
Mike Krieger

# Welcome back!

# Week 1 Recap

- HTML provides content
- CSS provides style
- Javascript provides action

# HTML Recap

- Set of tags that enclose images, video, text, & other content
- <head>er and <body>
-  boxes,  around short text

# HTML Recap, 2

```
<html>
    <head></head>
    <body></body>
</html>
```

# HTML Recap, 2

```
<html>
   <head></head>
   <body>
      <div class='title'>Hello</div>
      <div class='bio'>Hi, this is a little
bit about me</div>
   </body>
</html>
```

# HTML Recap, 2

```
<html>
   <head></head>
   <body>
      <div class='title'>Hello</div>
      <div class='bio'><span
class='greeting'>Hi</span>, this is a little
bit about me</div>
   </body>
</html>
```

# Questions from Week 1?

- Simulator price
- How to load files into localhost (will talk about it after class if interested)

# By the end of today

- Know how to style text & boxes using CSS
- Understand how CSS positioning works

# Today's topics

- Styling text
- Styling elements like boxes
- Positioning boxes & text
- Display & z-ordering

# What is CSS?

- **C**ascading **S**tyle **S**heets
- Cascading because multiple styles can affect one element
- Style sheets because that's how elements are styled

# CSS intro

- Where? Either in your <head>er or a separate file
- Set of *selectors* and *rules*

# First things first

- In-line: <style>...</style> in your <head>
- External file: <link rel="stylesheet" href="yourstylesheet.css" />
- In-element: 
- Changing in Javascript: element.style.backgroundColor = '';

# First things first

- **In-line: <style>...</style> in your <head>**
- **External file: <link rel="stylesheet" href="yourstylesheet.css" />**
- In-element: 
- Changing in Javascript: element.style.backgroundColor = '';

# What's a selector?

- Referencing one or many objects in the DOM that got created from your HTML

# The 3 selectors to learn

- **#***name* (selects by id)
- **.***name* (selects by class)
- *tagname* (selects by tag)

# Let's start.

- Open up simplebox.html in this week's folder

# Simple.html

```
<html>
<head>
   <style></style>
</head>
<body>
   <div class='content'>
      All <a href='work.html'>work</a> and no
      <a href='play.html'>play</a> makes <span
      class='person-name'>Jack</span> a dull
      boy
   </div>
   <div class='citation'>-Author unknown</div>
```

# Playing with selectors

*selector:* **body**

*original*

```
<body>

<div class='content'>All <a
href='work.html'>work</a> and
no <a href='play.html'>play</
a> makes <span id='person-
name'>Jack</span> a dull
boy</div>

<div class='citation'>-Author
unknown</div>

</body>
```

*selected*

```
<body>

<div class='content'>All <a
href='work.html'>work</a> and
no <a href='play.html'>play</
a> makes <span id='person-
name'>Jack</span> a dull boy</
div>

<div class='citation'>-Author
unknown</div>

</body>
```

# Playing with selectors

*selector:* **div**

*original*

```
<body>

<div class='content'>All <a
href='work.html'>work</a> and
no <a href='play.html'>play</
a> makes <span id='person-
name'>Jack</span> a dull
boy</div>

<div class='citation'>-Author
unknown</div>

</body>
```

*selected*

```
<body>

<div class='content'>All <a
href='work.html'>work</a> and
no <a href='play.html'>play</
a> makes <span id='person-
name'>Jack</span> a dull
boy</div>

<div class='citation'>-Author
unknown</div>

</body>
```

# Playing with selectors

*selector:* **.content**

*original*

```
<body>

<div class='content'>All <a
href='work.html'>work</a> and
no <a href='play.html'>play</
a> makes <span id='person-
name'>Jack</span> a dull
boy</div>

<div class='citation'>-Author
unknown</div>

</body>
```

*selected*

```
<body>

<div class='content'>All <a
href='work.html'>work</a> and
no <a href='play.html'>play</
a> makes <span id='person-
name'>Jack</span> a dull
boy</div>

<div class='citation'>-Author
unknown</div>

</body>
```

# Playing with selectors

*selector:* **#person-name**

*original*

```
<body>

<div class='content'>All <a
href='work.html'>work</a> and
no <a href='play.html'>play</
a> makes <span id='person-
name'>Jack</span> a dull
boy</div>

<div class='citation'>-Author
unknown</div>

</body>
```

*selected*

```
<body>

<div class='content'>All <a
href='work.html'>work</a> and
no <a href='play.html'>play</
a> makes <span id='person-
name'>Jack</span> a dull
boy</div>

<div class='citation'>-Author
unknown</div>

</body>
```

# Playing with selectors

*selector:* **a**

*original*

```
<body>

<div class='content'>All <a
href='work.html'>work</a> and
no <a href='play.html'>play</
a> makes <span id='person-
name'>Jack</span> a dull
boy</div>

<div class='citation'>-Author
unknown</div>

</body>
```

*selected*

```
<body>

<div class='content'>All <a
href='work.html'>work</a> and
no <a href='play.html'>play</
a> makes <span id='person-
name'>Jack</span> a dull
boy</div>

<div class='citation'>-Author
unknown</div>

</body>
```

# Playing with selectors

*selector:* **#another**

*original*

```
<body>

<div class='content'>All <a
href='work.html'>work</a> and
no <a href='play.html'>play</
a> makes <span id='person-
name'>Jack</span> a dull
boy</div>

<div class='citation'>-Author
unknown</div>

</body>
```

*selected*

```
<body>

<div class='content'>All <a
href='work.html'>work</a> and
no <a href='play.html'>play</
a> makes <span id='person-
name'>Jack</span> a dull
boy</div>

<div class='citation'>-Author
unknown</div>

</body>
```

# What's a rule?

- Set a property on selected elements

# What are properties?

- Visual
  - font size, font family, color, background, border, padding
- Position & Size
- Animations & Transitions

# Setting a property

```
.classname {

}
```

# Setting a property

```
.classname {
    rule:
}
```

# Setting a property

```
.classname {
    property: value;
}
```

# Setting a property

```
.classname {
    property: value;
    prop2: differentvalue;
}
```

# Making all links black

```
a {
    color: black;
}
```

# styling text

# Styling fonts overview

- *picking a font*: font-family
- size: font-size
- **weight**: font-weight
- '*style*': font-style
- Color: color

```
<div class='text'>The <span class='emphasis'>quick</
span> <span class='brown'>brown</span> fox jumps over
the <a href='lazy.html'>lazy dog</a></div>
```

css

{{empty}}

result

The quick brown fox jumps over the lazy dog

# Font-family

- One or more font family names, separated by a comma
  - Browser will pick the first it finds on user's system
  - Keywords: *serif* and *sans-serif* are last-ditch fallbacks

```
<div class='text'>The <span class='emphasis'>quick</
span> <span class='brown'>brown</span> fox jumps over
the <a href='lazy.html'>lazy dog</a></div>
```

CSS

```
.text {

}
```

result

The quick brown fox jumps over the lazy dog

```
<div class='text'>The <span class='emphasis'>quick</
span> <span class='brown'>brown</span> fox jumps over
the <a href='lazy.html'>lazy dog</a></div>
```

CSS

```
.text {
   font-family: Helvetica;
}
```

result

The quick brown fox jumps over the lazy dog

```
<div class='text'>The <span class='emphasis'>quick</
span> <span class='brown'>brown</span> fox jumps over
the <a href='lazy.html'>lazy dog</a></div>
```

CSS

```
.text {
   font-family: Gotham, Helvetica;
}
```

result

The quick brown fox jumps over the lazy dog

*no change; I don't have Gotham*

```
<div class='text'>The <span class='emphasis'>quick</
span> <span class='brown'>brown</span> fox jumps over
the <a href='lazy.html'>lazy dog</a></div>
```

CSS

```
.text {
   font-family: Gotham, sans-serif;
}
```

result

The quick brown fox jumps over the lazy dog

*no change; "sans-serif" defaults to Helvetica on the Mac*

```
<div class='text'>The <span class='emphasis'>quick</
span> <span class='brown'>brown</span> fox jumps over
the <a href='lazy.html'>lazy dog</a></div>
```

CSS

```
.text {
   font-family: Gotham, serif;
}
```

result

The quick brown fox jumps over the lazy dog

# A note on Web fonts

- Limited selection!

# Microsoft Core Fonts

Andale Mono
Arial
**Comic Sans MS**
Courier New
Georgia
**Impact**
Times New Roman
Trebuchet MS
Verdana

# Fonts on the iPhone

American Typewriter

**American Typewriter Condensed**

Arial

**Arial Rounded MT Bold**

Courier New

Georgia

Helvetica

**Marker Felt**

Times New Roman

Trebuchet MS

Verdana

Zapfino

# Sizing fonts

- font-size takes measurement in pixels (px), ems (em), percentages (%), or absolute measures (cm, mm, pt)

# What's an *em*?

- Relative measure: 1em = the point size of the current font
  - at 16pt, 1em=16pt
  - 1em = 100%
  - set global (default) in px, scale using ems

```
<div class='text'>The <span class='emphasis'>quick</
span> <span class='brown'>brown</span> fox jumps over
the <a href='lazy.html'>lazy dog</a></div>
```

CSS

```
.text {
   font-family: Helvetica;
   font-size: 4px;
}
```

# result

The quick brown fox jumps over the lazy dog

```
<div class='text'>The <span class='emphasis'>quick</
span> <span class='brown'>brown</span> fox jumps over
the <a href='lazy.html'>lazy dog</a></div>
```

CSS

```
.text {
   font-family: Helvetica;
}
```

result

The quick brown fox jumps over the lazy dog

```
<div class='text'>The <span class='emphasis'>quick</
span> <span class='brown'>brown</span> fox jumps over
the <a href='lazy.html'>lazy dog</a></div>
```

CSS

```
.text {
   font-family: Helvetica;
   font-size: 1em;
}
```

result

The quick brown fox jumps over the lazy dog

```
<div class='text'>The <span class='emphasis'>quick</
span> <span class='brown'>brown</span> fox jumps over
the <a href='lazy.html'>lazy dog</a></div>
```

CSS

```
body { font-size: 10px; }
.text {
   font-family: Helvetica;
   font-size: 1em;
}
```

result

The quick brown fox jumps over the lazy dog

```
<div class='text'>The <span class='emphasis'>quick</
span> <span class='brown'>brown</span> fox jumps over
the <a href='lazy.html'>lazy dog</a></div>
```

CSS

```
body { font-size: 10px; }
.text {
   font-family: Helvetica;
   font-size: 1.6em;
}
```

result

The quick brown fox jumps over the lazy dog

(back where we started)

# Font weight

- font-weight takes either:
  - number b/w 100 and 900 in 100-step increments
  - normal, bold, bolder, lighter
  - in practice, either normal or bold

```
<div class='text'>The <span class='emphasis'>quick</
span> <span class='brown'>brown</span> fox jumps over
the <a href='lazy.html'>lazy dog</a></div>
```

CSS

```
.text {
   font-family: Helvetica;
}
```

result

The quick brown fox jumps over the lazy dog

```
<div class='text'>The <span class='emphasis'>quick</
span> <span class='brown'>brown</span> fox jumps over
the <a href='lazy.html'>lazy dog</a></div>
```

CSS

```
.text {
   font-family: Helvetica;
   font-weight: bold;
}
```

result

**The quick brown fox jumps over the lazy dog**

# Setting font style

- font-style takes:
  - normal, oblique, italic
  - italic falls back to oblique

```
<div class='text'>The <span class='emphasis'>quick</
span> <span class='brown'>brown</span> fox jumps over
the <a href='lazy.html'>lazy dog</a></div>
```

CSS

```
.text {
   font-family: Helvetica;
   font-style: italic;
}
```

result

*The quick brown fox jumps over the lazy dog*

# Setting color

- color takes:
  - a hexadecimal value
  - a color word
  - an rgb() value

# Hexawhat?

- hex runs from 00 (nothing) to FF (full)
- so, 00...33..99..AA...DD...FF
- looks like: #RRGGBB
  - #0000FF (all blue)
  - #FFFF33 (yellow)
  - #c5d5ec  light blue

# Named colors

- black, blue, gray, red...
- huge list:
- http://www.w3schools.com/HTML/html_colornames.asp

# rgb()

- rgb(redval, greenval, blueval)
- Runs from 0 to 255
  - rgb(0, 0, 255) all blue
  - rgb(103, 233, 40) lime green
  - #RRGGBB -> rgb(rv, gv, bl)

```
<div class='text'>The <span class='emphasis'>quick</
span> <span class='brown'>brown</span> fox jumps over
the <a href='lazy.html'>lazy dog</a></div>
```

CSS

```
.brown {
   color: brown;
}
```

result

The quick brown fox jumps over the lazy dog

```
<div class='text'>The <span class='emphasis'>quick</
span> <span class='brown'>brown</span> fox jumps over
the <a href='lazy.html'>lazy dog</a></div>
```

CSS

```
.brown {
   color: #763017;
}
```

result

The quick brown fox jumps over the lazy dog

# Aligning text

- text-align:
  - left, right, center, justify

```
<div class='text'>The <span class='emphasis'>quick</
span> <span class='brown'>brown</span> fox jumps over
the <a href='lazy.html'>lazy dog</a></div>
```

CSS

```
.text {
   text-align: right;
}
```

result

The quick brown fox jumps over the lazy dog

# Underlining

- text-decoration:
  - none, underline, ~~strikethrough~~

# Cascading rules

# Cascading styles

- brown in inside the . What happens if we already have a color defined?

# Overridden styles

```
.text {
    color: gold;
}

.brown {
    color: brown;
}
```

The quick brown fox jumps over the lazy dog

# Cascade

- elements inherit from their parents
- specific styles on an element override anything on their parent

# Same rule, Same Element, Different places

- If same style for same element is defined in multiple places:
  - First check styles in the HTML for that element (<div style=...)
  - Then check CSS in <style> tags and loaded <link>ed stylesheets in reverse order

# Example

```
<link rel="stylesheet" href="quickbrown.css"
type="text/css" media="screen" title="no
title" charset="utf-8">
<style>
    .brown {
        color: yellow;
    }
</style>
```

The quick brown fox jumps over the lazy dog

even though stylesheet specifies .brown { color:brown;}, <style> takes precedence

# !important

```
<link rel="stylesheet" href="quickbrown.css"
type="text/css" media="screen" title="no
title" charset="utf-8">
<style>
    .brown { color: green !important }
    .brown {
        color: yellow;
    }
</style>
```

The quick brown fox jumps over the lazy dog

an !important declaration will jump out of the cascade to the top

# Example

```
<link rel="stylesheet" href="quickbrown.css"
type="text/css" media="screen" title="no
title" charset="utf-8">
<style>
    .brown {
        color: yellow;
    }
</style>
...
<span style="color:lightblue"
class='brown'>brown</span>
```

The quick brown fox jumps over the lazy dog

Style defined on the element takes precedence

# Firebug

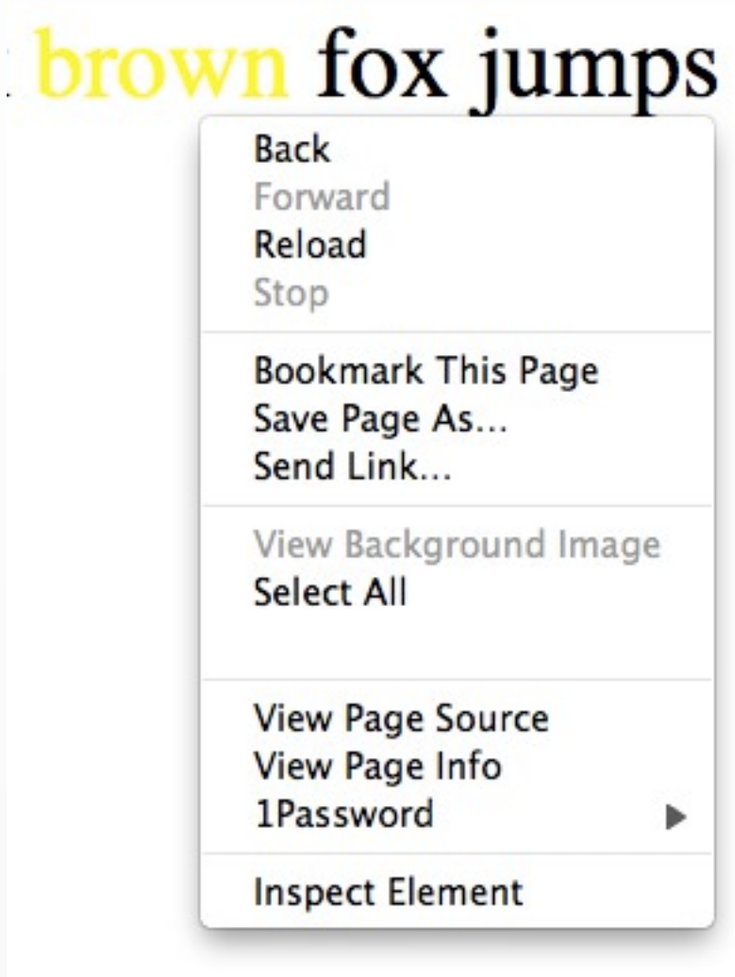
- Essential tool for Web development
- Developed by Joe Hewitt, now Facebook's lead iPhone dev
- Download at: http://getfirebug.com/

# Firebug

- Inspect any element to see the styles defined on it
- Lots of Javascript features too, we'll revisit in coming weeks

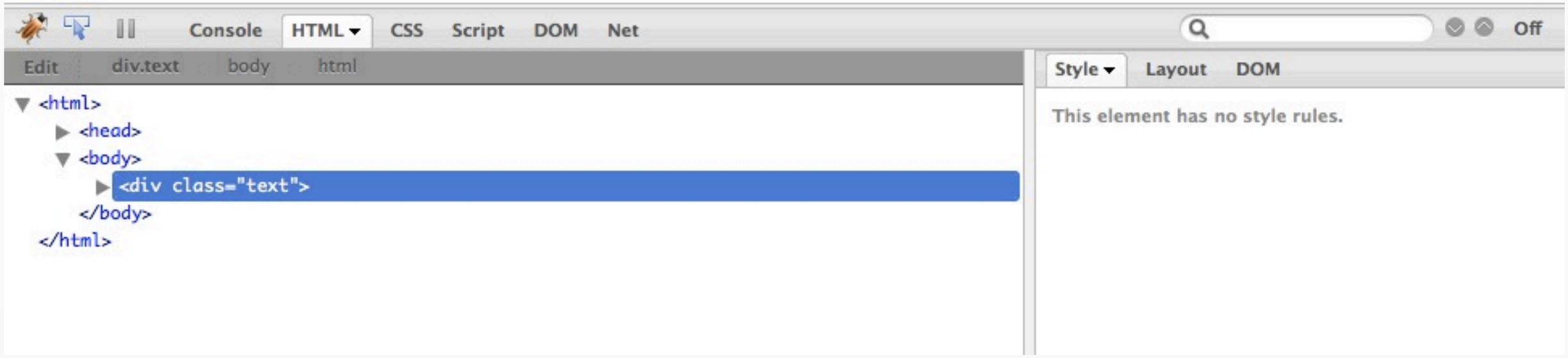
# Firebug How-To

- Right click anywhere in your page
- Click "Inspect Element"

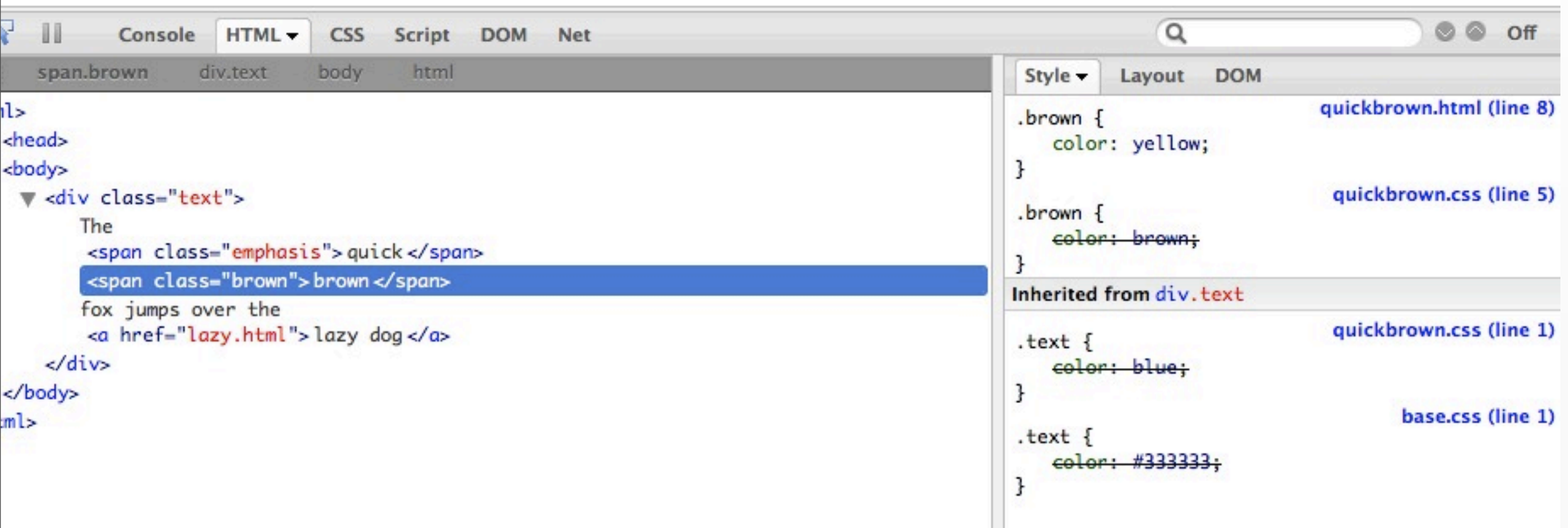


# Firebug

- Javascript console
- DOM Inspector
- CSS Editor
- Script debugger
- Network analyzer



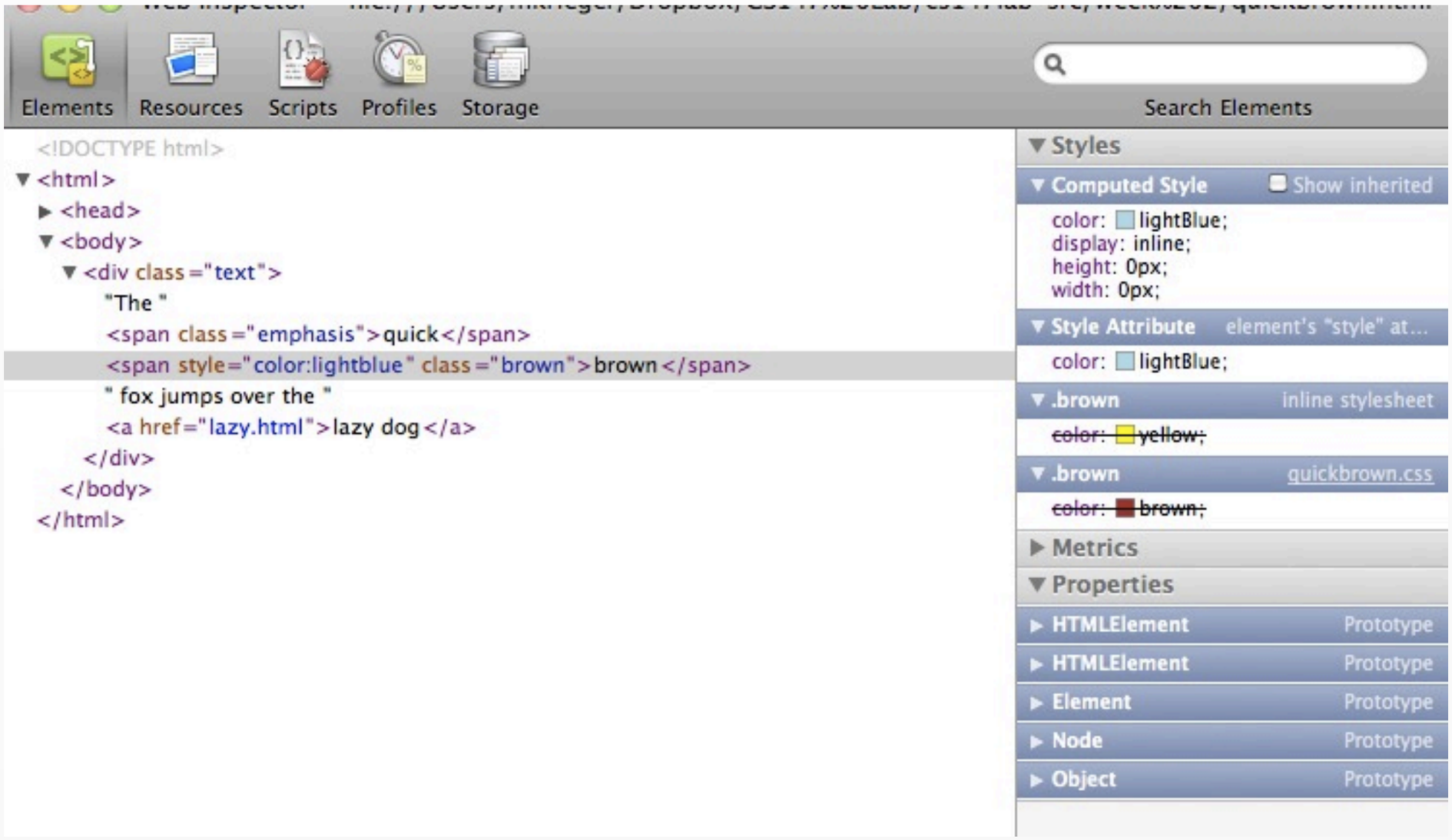
# Inspecting the 'brown' element



# Changing styles on the fly


# Safari 4 Inspector

- Mostly on par with Firebug



# Summary

- Use either Firebug or the Safari Inspector
- They'll be a huge time-saver!
- We'll see more of them soon

# Styling elements

# Properties

- border
- size
- margin
- padding
- display
- background
- opacity

# Border

- Either separately:
  - border-width, border-style, border-color
    - (specify at least style)
- Or together:
  - border: *width style color*

# Border style

- border-style
  - solid, dotted, dashed, hidden, groove, ridge, inset...
  - most of the time: solid, dotted, dashed

# Border color

- border-color (same color arguments as before)

# border-width

- Takes measurement(s) (px, em, etc)
- Overloaded:
  - border-width: 1px; (all sides 1px)
  - border-width: 2px 1px (top and bottom 2px, left and right 1px)
  - border-width: 2px 3px 4px 3px (top, right, bottom, left)
- Or, specify border-width-top, border-width-bottom, etc

```
<div id='box'>I'm in a box!</div>
```

CSS

```
#box {
   border-width: 5px;
}
```

(no change; we need to specify at least a width and a style)

result

I'm in a box!

```
<div id='box'>I'm in a box!</div>
```

CSS

```
#box {
    border-width: 5px;
    border-style: dotted;
}
```

result

I'm in a box!

```
<div id='box'>I'm in a box!</div>
```

CSS

```
#box {
    border-width: 5px 60px 4px 8px;
    border-style: solid;
}
```

result

```
<div id='box'>I'm in a box!</div>
```

CSS

```
#box {
    border-width: 5px;
    border-style: dotted;
    border-color: #ffccee;
}
```

result

I'm in a box!

```
<div id='box'>I'm in a box!</div>
```

CSS

```
#box {
    border: 5px dotted #ffccee;
}
```

result

I'm in a box!

# Sizing divs

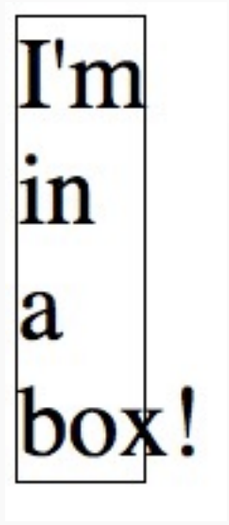
- width:
- height:
  - both take sizes or percentages
  - width: defaults to 100% of parent element
  - height: as big as the text needs to be

```
<div id='box'>I'm in a box!</div>
```

CSS

```
#box {
    width: 50px;
    border: 1px solid black;
}
```

result

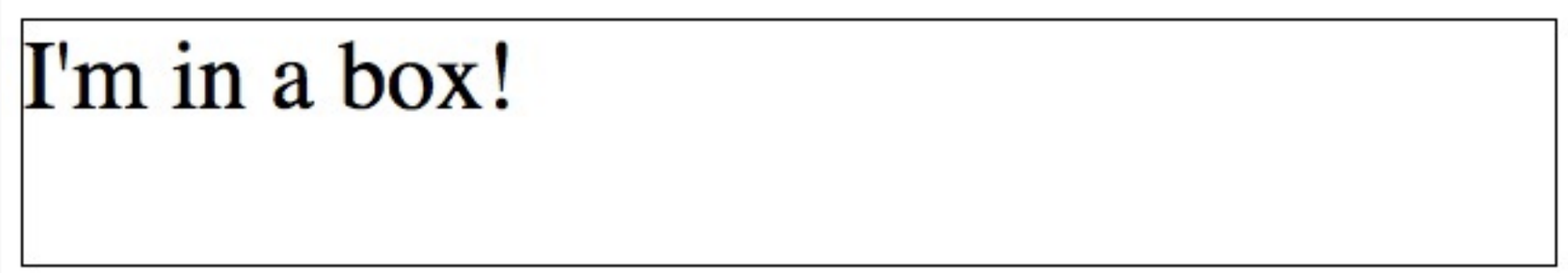


```
<div id='box'>I'm in a box!</div>
```

CSS

```
#box {
   height: 100px;
   border: 1px solid black;
}
```

result

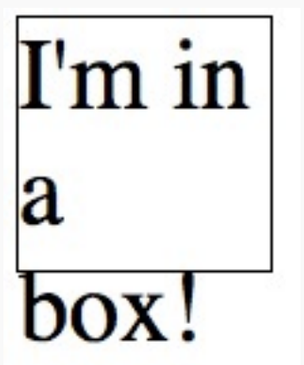


```
<div id='box'>I'm in a box!</div>
```

CSS

```
#box {
    height: 100px;
    width: 100px;
    border: 1px solid black;
}
```

result




text will overflow

# Overflow

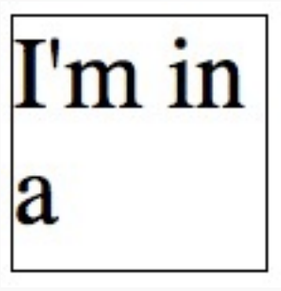
- **overflow** sets what happens when stuff goes outside the box
  - visible (default), hidden, scroll, auto

```
<div id='box'>I'm in a box!</div>
```

CSS

```
#box {
    height: 100px;
    width: 100px;
    border: 1px solid black;
    overflow: hidden;
}
```

result



```
<div id='box'>I'm in a box!</div>
```

CSS

```
#box {
    height: 100px;
    width: 100px;
    border: 1px solid black;
    overflow: scroll;
}
```

result

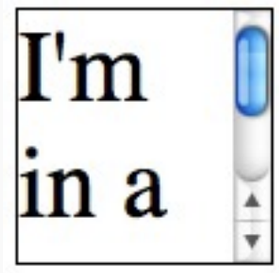

always shows a scroll box

```
<div id='box'>I'm in a box!</div>
```

CSS

```
#box {
    height: 100px;
    width: 100px;
    border: 1px solid black;
    overflow: auto;
}
```

result




only show scroll if needed

# Display

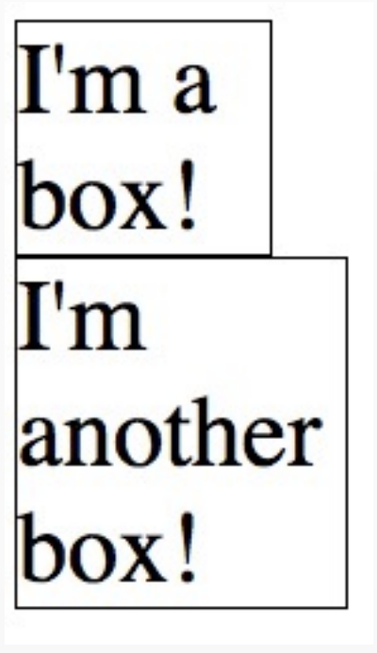
- **display** sets how the DOM element behaves:
  - none (hides the element)
  - block (has a width & height, line-break after)
  - inline (no width & height, no line-break)

```
<div id='firstbox'>I'm a box!</div>
<div id='secondbox'>I'm another box!</div>
```

## CSS

```
#firstbox {
    display: block;
    border: 1px solid black;
    width: 100px;
}
#secondbox {
    display: block;
    border: 1px solid black;
    width: 130px;
}
```

## result

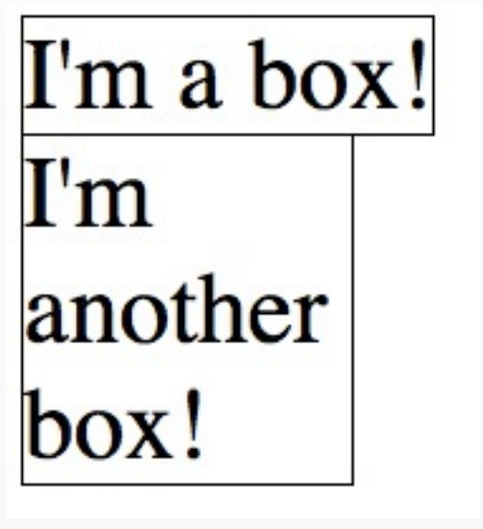


```
<div id='firstbox'>I'm a box!</div>
<div id='secondbox'>I'm another box!</div>
```

## CSS

```
#firstbox {
    display: inline;
    border: 1px solid black;
    width: 100px;
}
#secondbox {
    display: block;
    border: 1px solid black;
    width: 130px;
}
```

## result

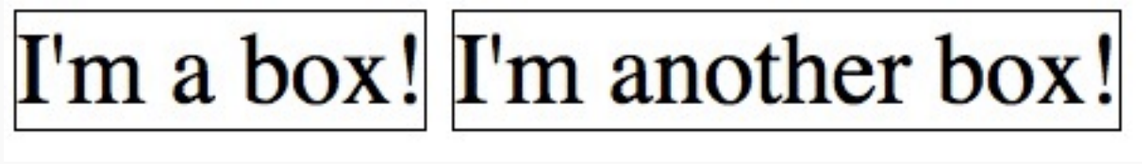


```
<div id='firstbox'>I'm a box!</div>
<div id='secondbox'>I'm another box!</div>
```

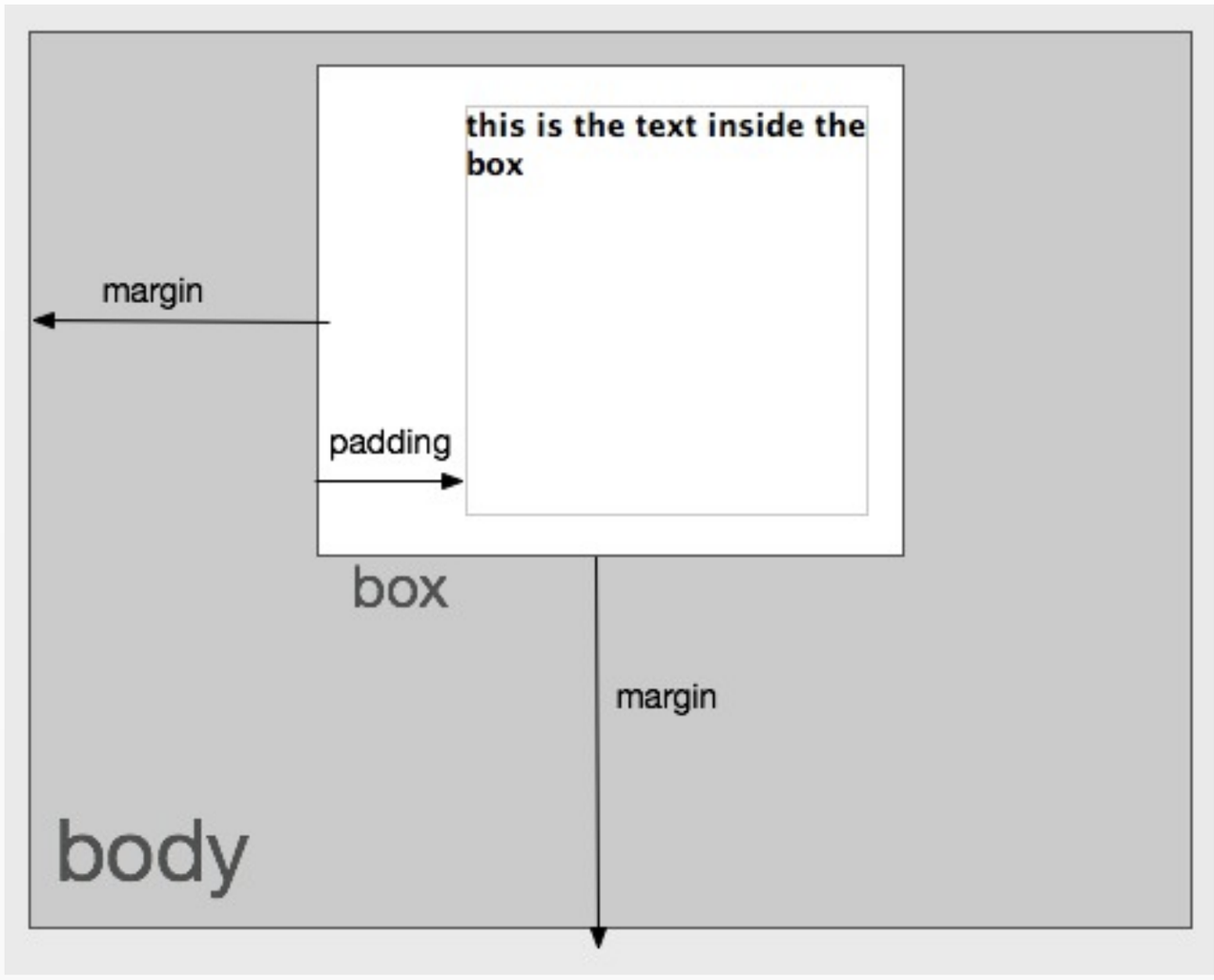
## CSS

```
#firstbox {
    display: inline;
    border: 1px solid black;
    width: 100px;
}
#secondbox {
    display: inline;
    border: 1px solid black;
    width: 130px;
}
```

## result



# Margin & Padding



# Margin & Padding

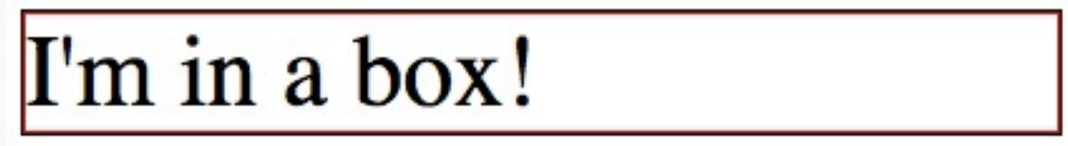
- Margin is distance from element from its parent
- Padding is distance from edge of element to its contents

```
<div id='outerbox'><div id='box'>I'm in a box!</div></
div>
```

## CSS

```
#outerbox {
    border: 1px solid black;
}

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

## result



```
<div id='outerbox'><div id='box'>I'm in a box!</div></
div>
```

## CSS

```
#outerbox {
    border: 1px solid black;
}

#box {
    border: 1px solid red;
    margin: 3px 10px;
}
```

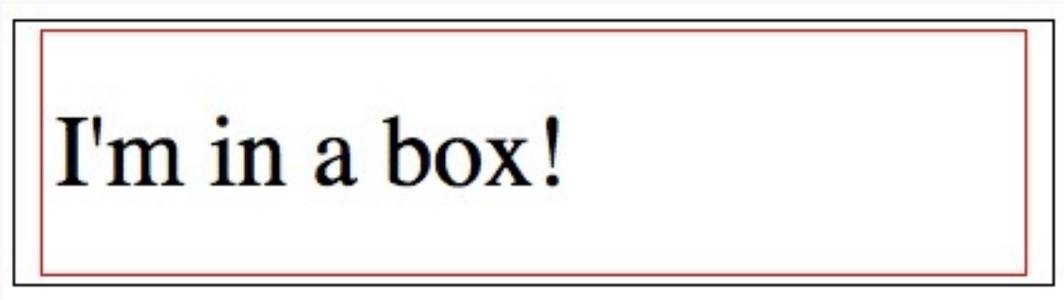
## result

```
<div id='outerbox'><div id='box'>I'm in a box!</div></
div>
```

## CSS

```
#outerbox {
    border: 1px solid black;
}

#box {
    border: 1px solid red;
    margin: 3px 10px;
    padding: 25px 5px;
}
```

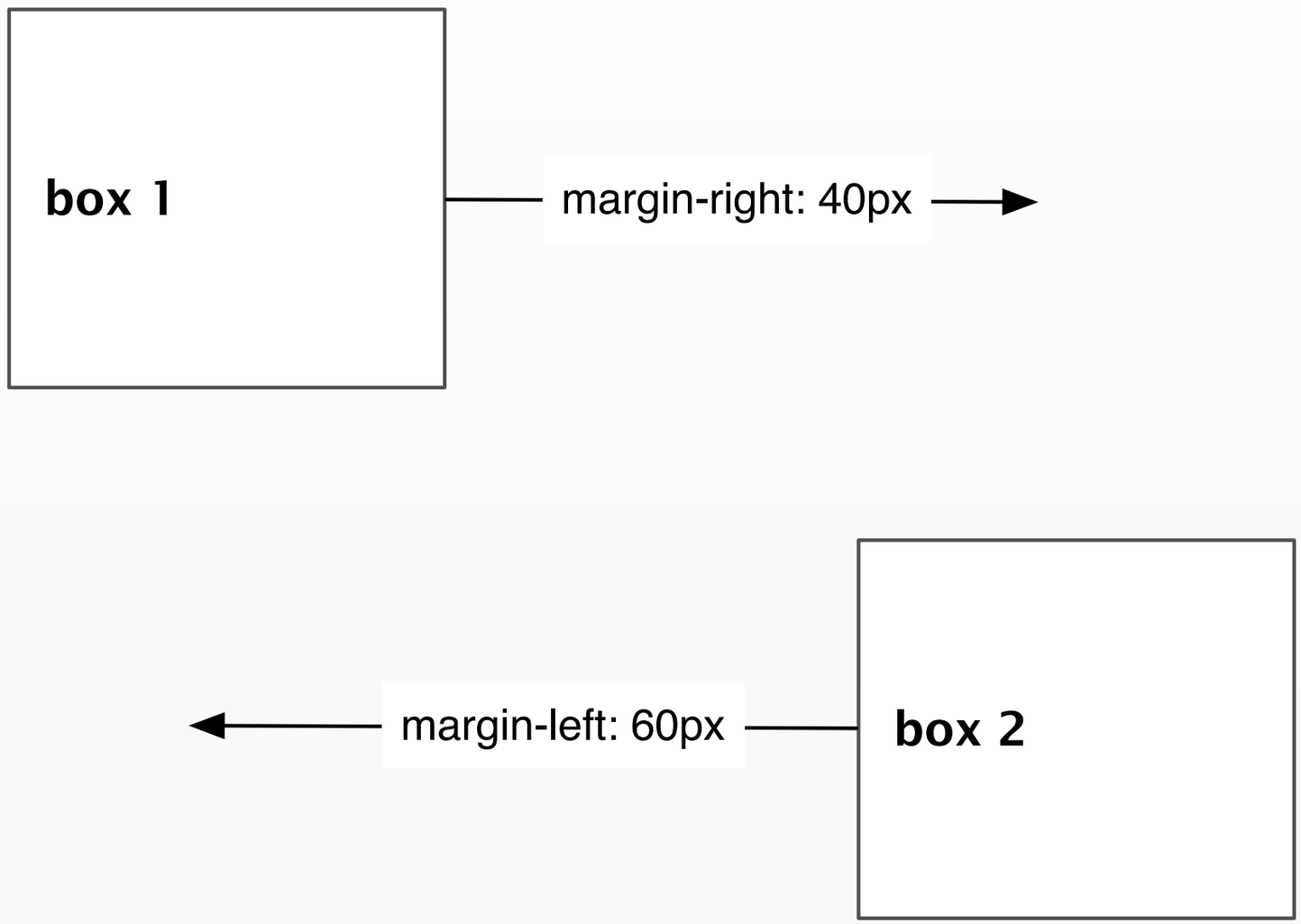
## result



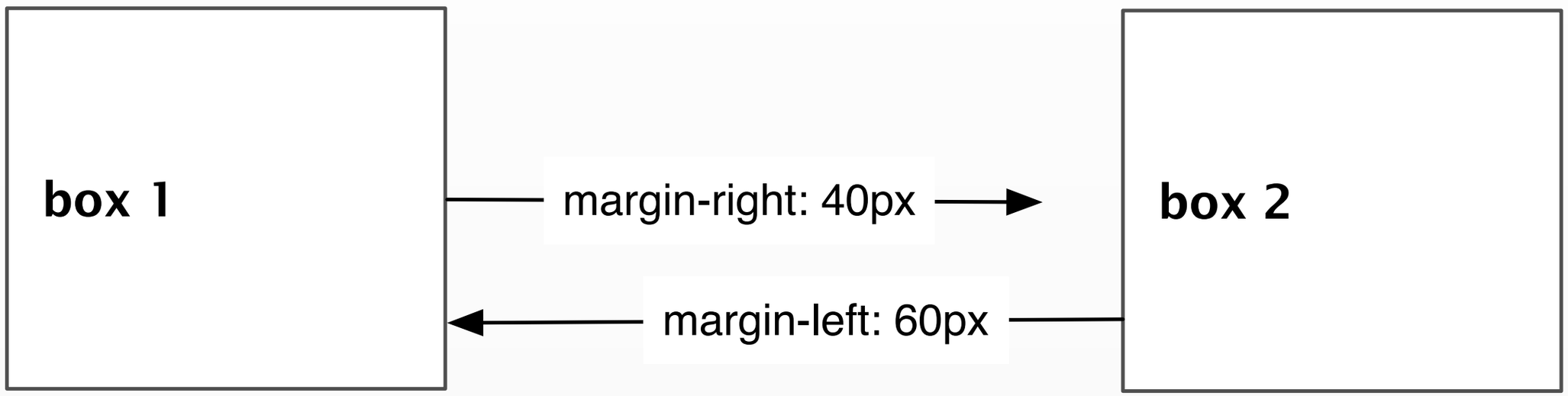
# Overlapping margins

- CSS will find the *minimum* margin between two elements that satisfies their margin requirements

# Example



# If they were next to each other



# Centering an element

```
margin: 0 auto; /* or margin: 10px auto, etc */
```

telling the browser:
0px top and bottom margin
automatic left/right margin to center element

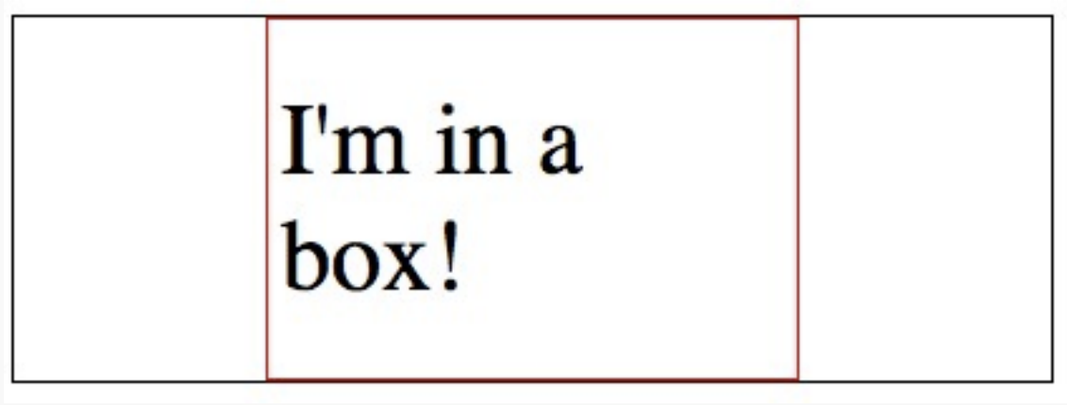
*note:* a width needs to be set

```
<div id='outerbox'><div id='box'>I'm in a box!</div></
div>
```

## CSS

```
#outerbox {
    border: 1px solid black;
}

#box {
    border: 1px solid red;
    margin: 0px auto;
    width: 200px;
    padding: 25px 5px;
}
```

## result



# Background

- background properties:
  - background-image
  - background-color
  - background-repeat

# Background color

- background-color takes:
  - a color spec as before

```
<div id='outerbox'><div id='box'>I'm in a box!</div></
div>
```

## CSS

```
#outerbox {
    border: 1px solid black;
}

#box {
    margin: 10px;
    background-color: red;
}
```

## result

# background-image

- takes a url( ) parameter with the link to the file

```
<div id='outerbox'><div id='box'>I'm in a box!</div></
div>
```

## CSS

```
#outerbox {
    border: 1px solid black;
}

#box {
    margin: 10px;
    background-color: red;
    background-image: url(http://dryicons.com/files/graphics_previews/
decorative_background.jpg);
```

## result

# background-repeat

- can specify:
  - repeat-y, repeat-x, or just repeat (will repeat both x and y)

# Opacity

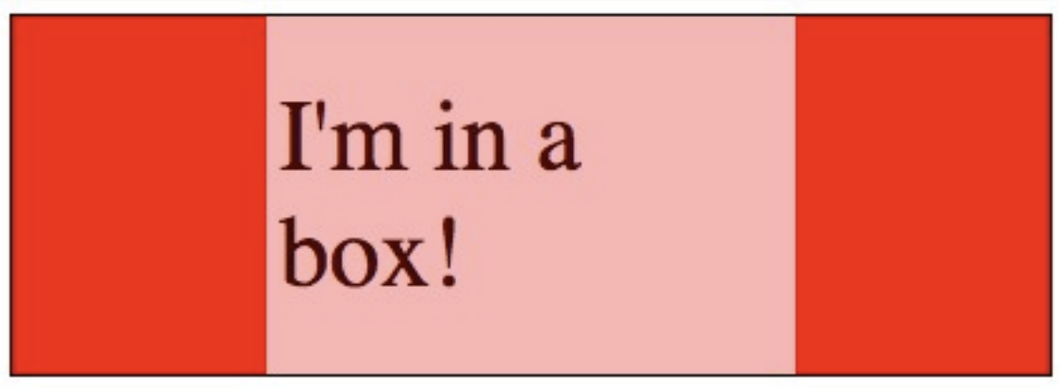
- takes a number between 0 and 1.0 and sets how opaque the div **and its contents** will be

```
<div id='outerbox'><div id='box'>I'm in a box!</div></
div>
```

## CSS

```
#outerbox {
    border: 1px solid black;
    background-color: red;
}

#box {
    margin: 0 auto;
    width: 200px;
    padding: 25px 5px;
    background-color: white;
    opacity: 0.7;
}
```

## result



# Positioning boxes & text

# Positioning Elements

- Two ways:
  - absolute & relative positioning
  - floats

# Absolute & Relative

- Changing an element's **absolute** position takes it out of the normal flow of the DOM (other elements act like it was never there)
- Changing an element's **relative** position makes the DOM act like the element is still in its starting spot

# Specifying position

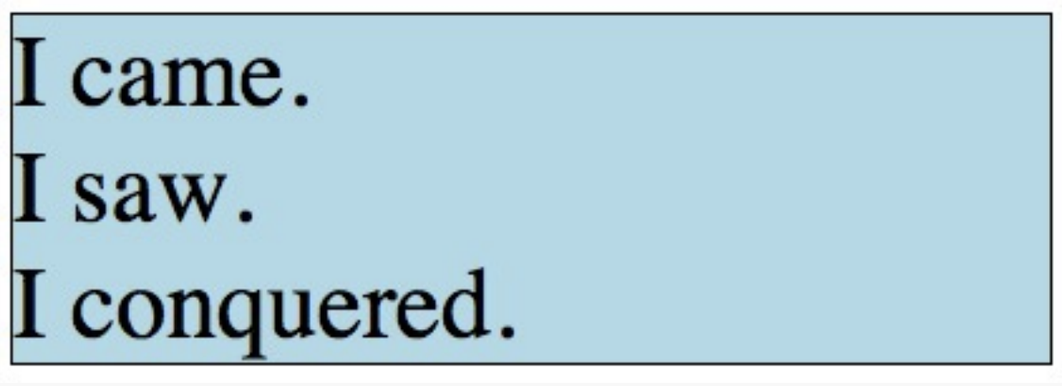
- **position:** relative|absolute|normal;
- **left/right/top/bottom:** size of how much to displace for any of these directions

```
<div id='outerbox'><div id='firstbox'>I came.</div><div
id='secondbox'>I saw.</div><div id='thirdbox'>I
conquered.</div></div>
```

## CSS

```
#outerbox {
    border: 1px solid black;
    background: lightblue;
}
```

## result

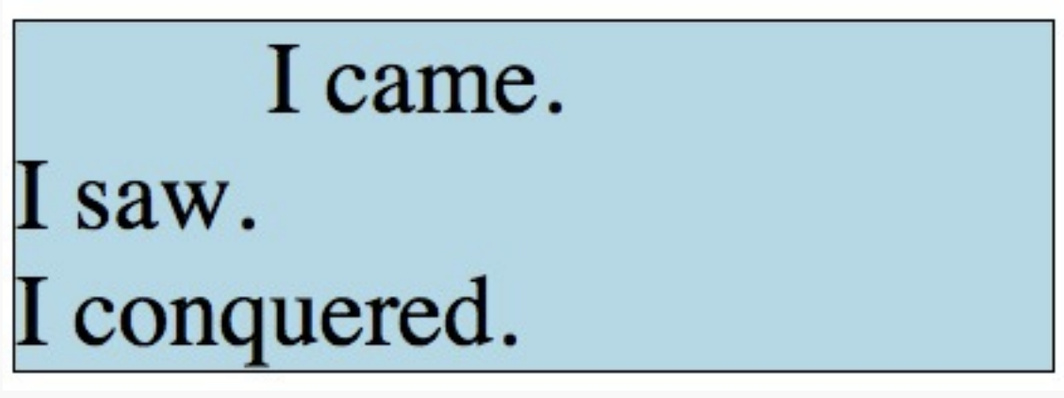


```
<div id='outerbox'><div id='firstbox'>I came.</div><div
id='secondbox'>I saw.</div><div id='thirdbox'>I
conquered.</div></div>
```

## CSS

```
#outerbox {
    border: 1px solid black;
    background: lightblue;
}

#firstbox{
    position: relative;
    left: 100px;
}
```

## result



```
<div id='outerbox'><div id='firstbox'>I came.</div><div
id='secondbox'>I saw.</div><div id='thirdbox'>I
conquered.</div></div>
```

## CSS

```
#outerbox {
    border: 1px solid black;
    background: lightblue;
}

#firstbox{
    position: relative;
    left: 100px;
}

#secondbox {
    position: absolute;
    left: 220px;
}
```

## result

I came.
I conquered. I saw.

# Positioned relative to what?

- Its **first positioned parent**, or the **body** if nothing is positioned relative or absolute
- In other words, walk up the DOM tree until it finds a node that has been **position**ed.

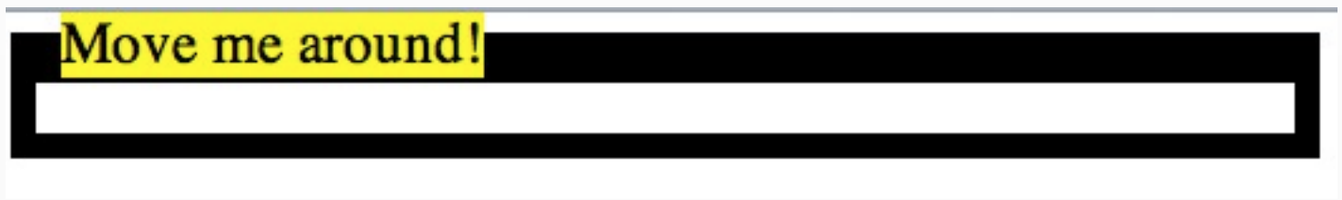
# Example

```
<div id='grandparent'>
    <div id='firstparent'>
        <div id='movemearound'>
            Move me around!
        </div>
    </div>
</div>
```

# Nothing positioned

# So if we position the movemearound box

```
#movemearound{
    position: absolute;
    background: yellow;
    top: 0;
}
```




...it's positioned relative to the whole body, not any of its parents

# Now let's position the grandparent

```
#grandparent {
     background: black;
     padding: 10px;
     width: 500px;
     position: absolute;
     left: 30px;
     top: 20px;
}

#firstparent {
     background: white;
     margin-top: 10px;
}

#movemearound{
     // disabling this positioning for now
}
```

# Now grandparent is positioned

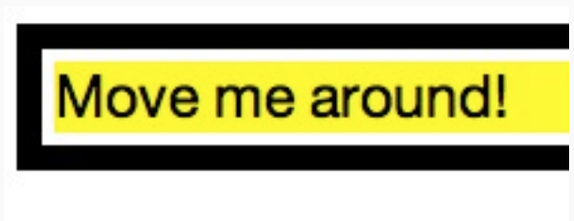
# So if we position movemearound again...

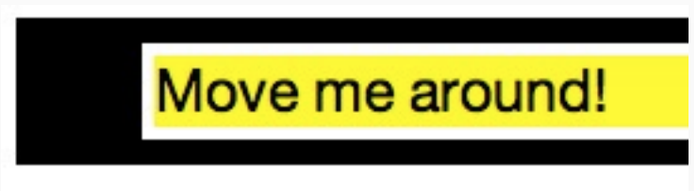
```
#grandparent {
     background: black;
     padding: 10px;
     width: 500px;
     position: absolute;
     left: 30px;
     top: 20px;
}

#firstparent {
     background: white;
     margin-top: 10px;
}

#movemearound{
     position: absolute;
     background: yellow;
     top: 0;
}
```


it's positioned relative to the grandparent

# Now let's position its direct parent

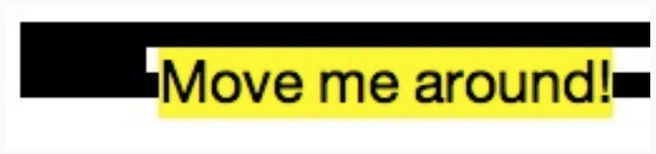
```
#firstparent {
    background: white;
    position: relative;
    left: 40px;
    padding: 5px;

}
```



# So, if we now position movemearound...

```
#movemearound{
    position: absolute;
    background: yellow;
    top: 0px;

}
```




...it's positioned relative to #firstparent, not #grandparent or the body

# Position Summary

- position: relative doesn't affect the elements around it; DOM acts like it's still there
- position: absolute takes element out of the DOM flow
- left/right/top/bottom properties are relative to the first parent from the element that has a position attached...or the body

# float

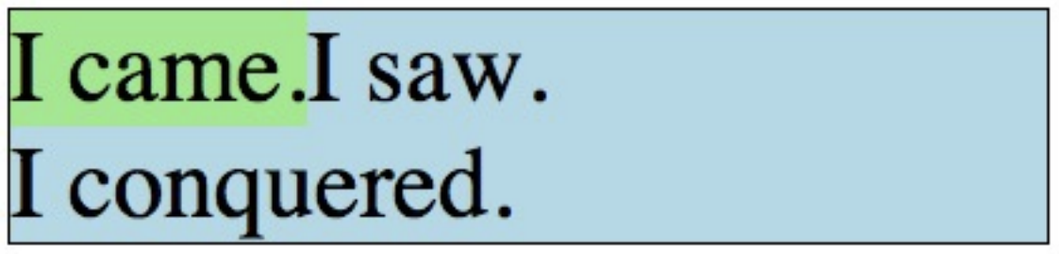
- **float** right/left will move an element as far in that direction in its bounding box as possible. Everything else in that box (other than other floated elements) flows around it.

```
<div id='outerbox'><div id='firstbox'>I came.</div><div
id='secondbox'>I saw.</div><div id='thirdbox'>I
conquered.</div></div>
```

## CSS

```
#outerbox {
    border: 1px solid black;
    background: lightblue;
}

#firstbox{
    background: lightgreen;
    float: left;
}
```

## result

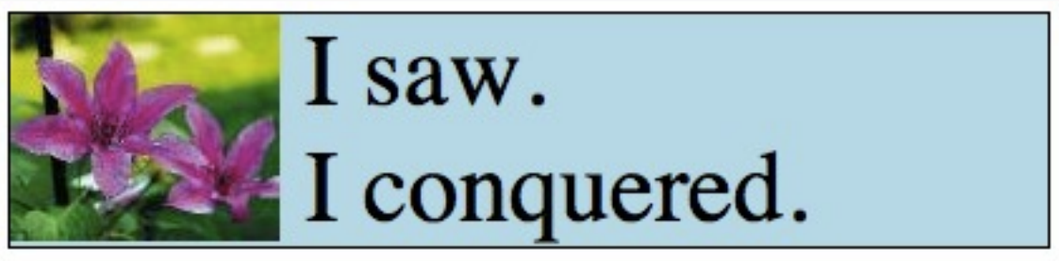


```
<div id='outerbox'><img id="flowerimage" src="http://www.w3schools.com/CSS/klematis_small.jpg"/> <div id='secondbox'>I saw.</div><div id='thirdbox'>I conquered.</div></div>
```

## CSS

```
#outerbox {
    border: 1px solid black;
    background: lightblue;
}

#flowerimage{
    float: left;
    margin-right: 10px;
}
```

## result

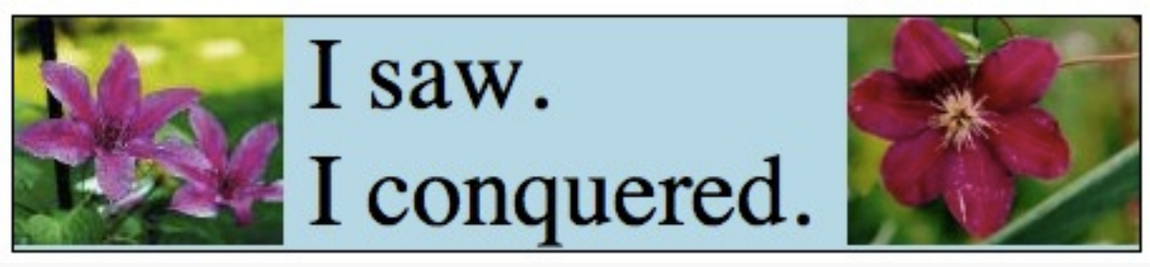


```
<div id='outerbox'><img id="flowerimage" src="http://
www.w3schools.com/CSS/klematis_small.jpg"/><img
id="otherflower" src="http://www.w3schools.com/CSS/
klematis3_small.jpg"/> <div id='secondbox'>I saw.</
```

CSS

```
#outerbox {
    border: 1px solid black;
    background: lightblue;
}

#flowerimage{
    float: left;
    margin-right: 10px;
}

#otherflower{
    float: right;
}
```

result

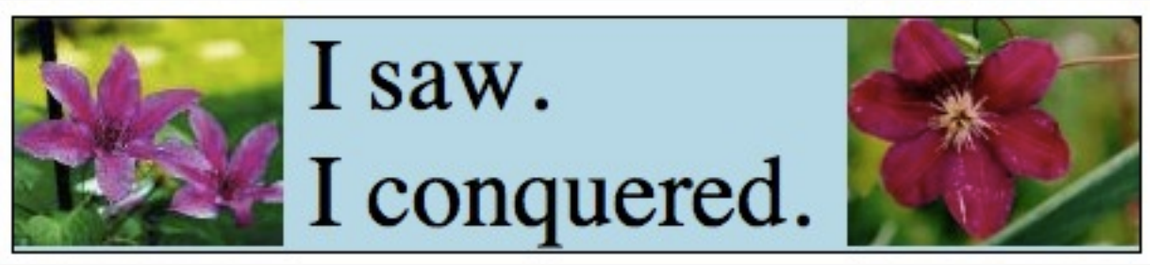


```
<div id='outerbox'><img id="flowerimage" src="http://www.w3schools.com/CSS/klematis_small.jpg"/><img id="otherflower" src="http://www.w3schools.com/CSS/klematis3_small.jpg"/> <div id='secondbox'>I saw.</div><div id='thirdbox'>I conquered.</div></div>
```

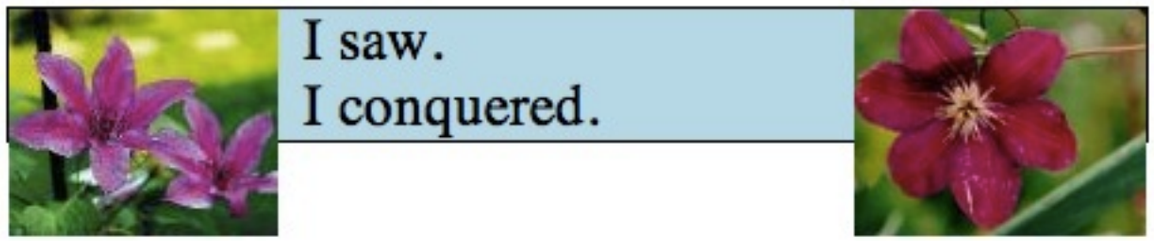
CSS

```
#outerbox {
    border: 1px solid black;
    background: lightblue;
}

#flowerimage{
    float: left;
    margin-right: 10px;
}

#otherflower{
    float: right;
}
```

result



# Floated elements don't flow



# Clearing a float

- Setting clear: left|right|both on an element will put it after the full height of floated elements on the left|right or both sides

# Clearing floats

```
#outerbox {
    border: 1px solid black;
    background: lightblue;
}

#flowerimage{
    float: left;
    margin-right: 10px;
}

#otherflower{
    float: right;
}

#thirdbox {
    clear: both;
}
```

# Putting it in practice

# 2 ways of doing a left-side menu with right content

- using **position**
- using **float**
- let's pretend we're building Axess

# Our starter code

```
<div id="nav">
    <ul>
        <li>Home</li>
        <li>Add/Edit Courses</li>
        <li>Order books</li>
        <li>Make a payment</li>
    </ul>
</div>
<div id="content">
    Lorem ipsum...
</div>
```

# Looks like this:

- Home
- Add/Edit Courses
- Order books
- Make a payment

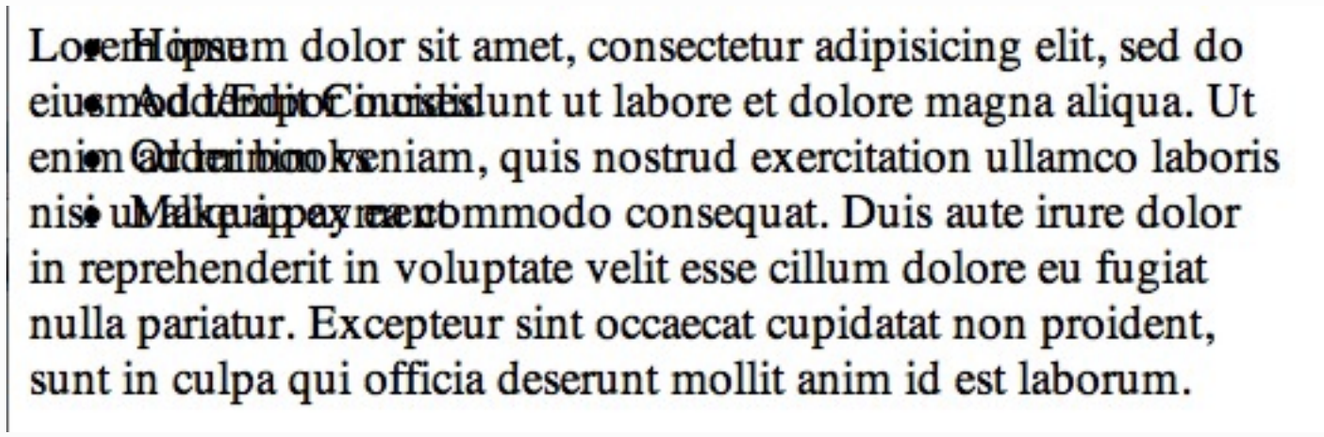
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

# method 1 : position

- Strategy: pull the left menu out of the DOM flow using **position: absolute**
- Then move the content over by using **position: relative**

# step 1

```
#nav {
    position: absolute;
}
```




#nav is now out of the flow, so the #content is overlapping it

# step 2

```
#nav {
    position: absolute;
    width: 200px;
}

ul {
    margin: 0px;
}

#content {
    position: relative;
    left: 220px;
}
```

- Home
- Add/Edit Courses
- Order books
- Make a payment

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

# This looks good, but...

- What happens when the right side is shorter than the left?

# with a footer

```
<div id="nav">
    <ul>
        <li>Home</li>
        <li>Add/Edit Courses</li>
        <li>Order books</li>
        <li>Make a payment</li>
    </ul>
</div>
<div id="content">
    If the content is short...
</div>
<div id="footer">What if we wanted more information like a
footer?</div>
```

- Home
- About
- Courses
- Order books
- Make a payment

If the content is short...

What if we wanted more information like a footer?

# Uh oh…

- We need a different approach if we want a footer

# Method 2: float

- Strategy: float both elements left

# Step 1

```
#nav {
    float: left;
    width: 200px;
}

body {
    width: 800px;
}

ul {
    margin: 0px;
}

#content {
    width: 500px;
}
```

- Home
- Add/Edit Courses
- Order books
- Make a payment

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

# Step 2

```
#nav {
    float: left;
    width: 200px;
}

body {
    width: 800px;
}

ul {
    margin: 0px;
}

#content {
    float: left;
    width: 500px;
}
```

- Home
- Add/Edit Courses
- Order books
- Make a payment

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

# And now for a footer...

- We can use **clear**!

# first, adding the footer

- Home
- Add/Edit Courses
- Order books
- Make a payment

If the content is short...
we wanted more information like a footer?

What if

# now add clear

```
#nav {
     float: left;
     width: 200px;
}

body { width: 800px;}
ul { margin: 0px;}

#content {
     float: left;
     width: 500px;
}

#footer {
     clear: both;
}
```

- Home
- Add/Edit Courses
- Order books
- Make a payment

If the content is short...

What if we wanted more information like a footer?

# Success!

# Portfolio Part 2

```
<!DOCTYPE html>
<html>
     <head>
          <title>CS147 Portfolio, Day 1</title>
          <meta name="viewport" content="width=device-width, user-scalable=no" />
          <link rel="stylesheet" href="week02.css" type="text/css" media="screen" title="no title"
charset="utf-8">
     </head>
     <body>
          <div class='content'>
               <div id="info">
                    <div id='fullname'>Joe <span id='last-name'>Student</span></div>
                    <div id="tagline">joe@student.com<br/>Interface Designer</div>
               </div>
               <div id='bio'>
               <h2>Bio</h2>
                    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation
ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in
voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non
proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</div>
               <div id='assignments'>
                    <h2>Course work</h2>
                    <ul>
                         <li><a href="how-might-we.html">How Might We?</a></li>
                         <li><a href="inspiration.html">Inspiration</a></li>
                         <li><a href="discovery.html">Discovery</a></li>
                    </ul>
               </div>
               <!-- we need this for our float layout -->
               <div class='clear'></div>
          </div>
     </body>
</html>
```

# Looks like...

Joe Student
joe@student.com
Interface Designer

## Bio

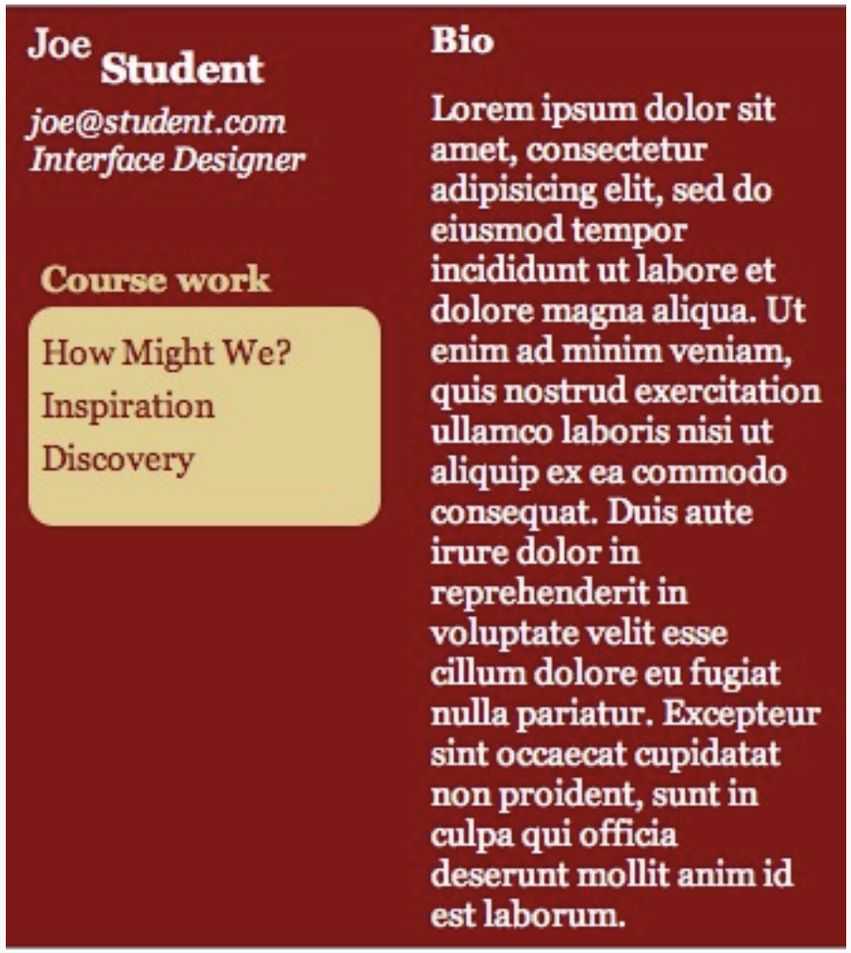
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

## Course work

- How Might We?
- Inspiration
- Discovery

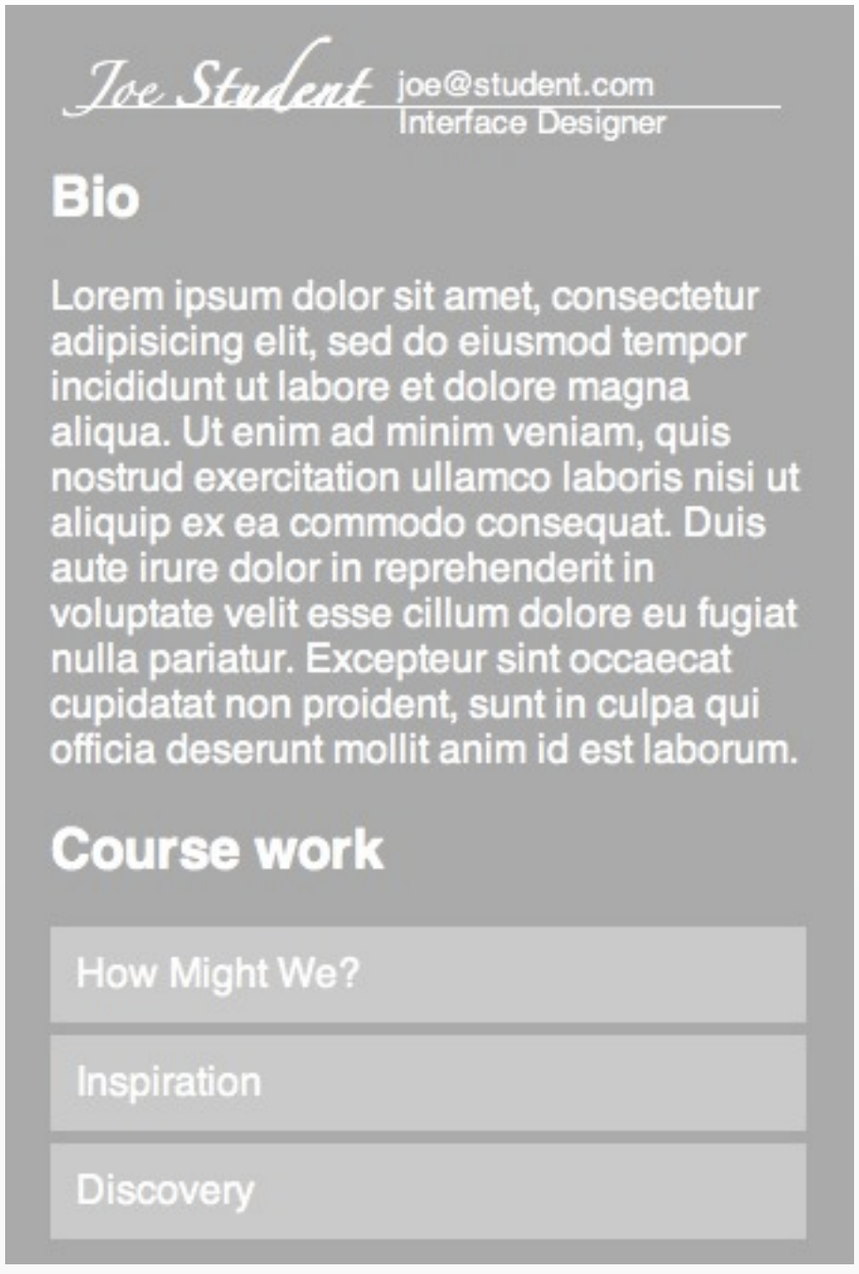
```
body {
	font-family: Georgia;
	margin: 0px;
}

#assignments {
	float: left;
	width: 130px;
	margin-top: 25px;
	padding: 5px 5px;
	-webkit-border-radius: 10px;
	background-color: #e4d18f;
	font-size: 0.9em;
	position: relative;
	top: 40px;
}

#assignments h2 {
	position: absolute;
	top: -19px;
	color: #e4d18f;

}

#assignments a {
	text-decoration: none;
	color: #8a000e;
}

.content {
	border: 1px solid gray;
	background: #8a000e;
	color: white;
	margin: 0px auto;
	width: 315px;
	padding: 5px 10px;
}

#info {
	width: 140px;
	float: left;
}

#last-name {
	font-weight: bold;
	position: relative;
	top: 10px;
}

#tagline {
	font-size: 0.8em;
	position: relative;
	left: 1px;
	font-style: italic;
	top: 14px;

}

h2 { margin-top: 0px; font-size: 1.0em; }


#bio {
	float: right;
	width: 155px;;
	font-size: 0.9em;
}

ul {
	margin-left: 0px;
	padding-left: 0px;
	margin-top: 0px;
	list-style-type: none;
}

li { margin-bottom: 5px; margin-top: 5px;  }
.clear { clear: both; }
```

# Looks like...


Joe Student
joe@student.com
Interface Designer
Course work
How Might We?
Inspiration
Discovery
Bio
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

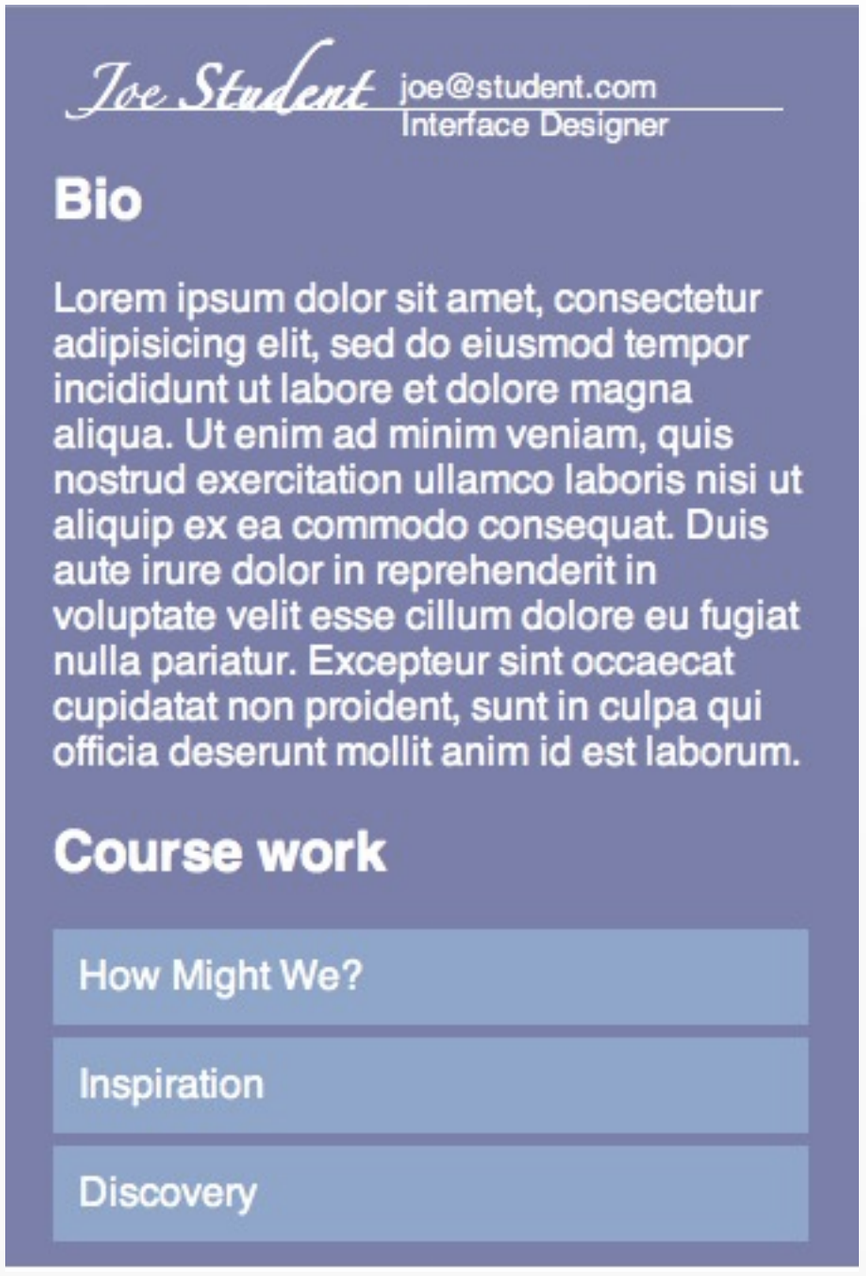
```
body {
     background-color: white;
     color: white;
     font-family: Helvetica;
     margin: 0px;

}

.content {
     background-color: #AAA;
     margin: 0px auto;
     width: 300px;
     padding: 10px 20px;
     height: 480px;
}

#info {
     width: 280px;
     margin: 0px auto;
     border-bottom: 1px solid white;
}

#last-name {
     font-family: Zapfino;
     font-weight: bold;
}

#fullname {
     float: left;
     font-family: Zapfino;
}

#tagline {
     font-size: 0.8em;
     position: relative;
     left: 14px;
     top: 14px;

}

h2 { font-size: 1.4em; }

#bio { clear: both; }

ul {
     margin-left: 0px;
     padding-left: 0px;
     list-style-type: none;
}
li {
     background-color: #cacaca;
     padding: 10px;
     margin-bottom: 5px;
}

a {
     text-decoration: none;
     color: white;
}
```

# Looks like...


Joe Student
joe@student.com
Interface Designer
Bio
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
Course work
How Might We?
Inspiration
Discovery

```
body {
	background-color: white;
	color: white;
	font-family: Helvetica;
	margin: 0px;

}

.content {
	background-color: #7a7cab;
	margin: 0px auto;
	width: 300px;
	padding: 10px 20px;
	height: 480px;
}

#info {
	width: 280px;
	margin: 0px auto;
	border-bottom: 1px solid white;

}

#last-name {
	font-family: Zapfino;
	font-weight: bold;
}

#fullname {
	width: ;
	float: left;
	font-family: Zapfino;
}

#tagline {
	font-size: 0.8em;
	position: relative;
	left: 14px;
	top: 14px;

}

h2 {
	font-size: 1.4em;
}

#bio {
	clear: both;
}

ul {
	margin-left: 0px;
	padding-left: 0px;
	list-style-type: none;
}
li {
	background-color: #8aa5cb;
	padding: 10px;
	margin-bottom: 5px;
}

a {
	text-decoration: none;
	color: white;
}
```

# Looks like...


Joe Student
joe@student.com
Interface Designer
Bio
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
Course work
How Might We?
Inspiration
Discovery

# What we did today

- How to add CSS to your HTML
- What CSS looks like
- How the "cascade" works
- How to style fonts & boxes
- How to position elements