

CSC309 Winter 2016

Lecture 2

Larry Zhang

Announcements

- Assignment 1 is out, due Jan 25, 10pm. Start Early!
 - Work in groups of 2, make groups on MarkUs.
 - Make sure you can login to MarkUs, if not let me know.
-
- Tutorial 1 this Wednesday, very important!
 - You will setup your personal web space on cs.utm servers.
 - You will exercise with HTML and CSS, which is what A1 is about.
 - Consider go to the 6-7pm section, fewer people, more help.
 - If you are from StG, email me your UTORID so I can create account for you on cs.utm servers.

Cascading **S**tyle **S**heets

Why CSS?

→ For each element that we have in the HTML file, we want to define some **style** for it.

- ◆ size
- ◆ color
- ◆ position
- ◆ margin
- ◆ alignment
- ◆ etc....

The primitive way of styling

Shashdot. News for **nerds!!** You will never, EVER be **BORED** here!

output

```
<p>
<font face="Arial">Shashdot.</font>
News for <b>nerds!!</b> You will <i>never</i>, <u>EVER</u>
be
<font size="+4" color="red">BORED</font> here!
</p>
```

HTML

What's bad about this?

Why CSS

We would like to **decouple** the code that describes

→ **What** information is displayed



HTML file

→ **How** the information is displayed

CSS file

→ Because of the separation, one CSS file can control the style of multiple HTML files.

→ **Decoupling** is an important idea in software engineering!

Using a stylesheet file, e.g., “style.css”

```
<head>
...
<link href="style.css" type="text/css" rel="stylesheet" />
...
</head>
```

One HTML file can include **multiple** stylesheets.

What if have **conflicts**? Good question. Will talk about it later.

Can also embed into HTML file

```
<head>  
<style type="text/css">  
p { font-family: sans-serif; color: red; }  
h2 { background-color: yellow; }  
</style>  
</head>
```

HTML

Avoid using in general. Only for quick-and-dirty tests.

demo

<http://www.csszengarden.com/>

Basic CSS Rule Syntax

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

The HTML elements selected by the selector have style defined by all the property-value pairs.

Selectors

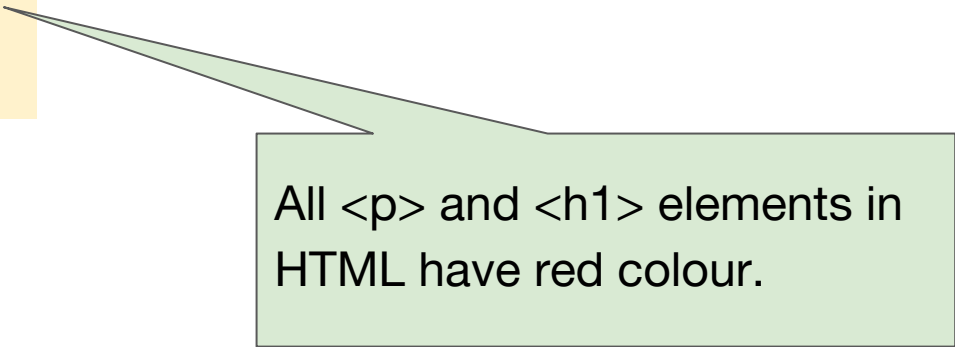
Select all elements of a kind

```
p {  
    font-size:100px;  
    color:red;  
    font-weight:bold;  
}
```

All <p> elements in HTML have font size 100px, red color and are boldface.

You can group different selectors (use **comma**)

```
p, h1 {  
    color:red;  
}
```



All <p> and <h1> elements in HTML have red colour.

Select A inside B (use **space**)

```
<p>The red paragraph</p>
<div>
  <p>I want to be blue!</p>
</div>
```

```
div p {
  color:blue;
}
```

All `<p>` inside `<div>` have blue colour.

Care about **immediate** parent (use “>”)

```
<div>
  <span><p>I won't turn blue!</p></span>
</div>
<div>
  <p>I am blue.</p>
</div>
```

```
div > p {
  color:blue;
}
```

select <p> whose **immediate**
parent is a <div>

Select according attribute (use **[attr]**)

```
<a target="_blank" href="x.html">I'm blue.</a>  
<a href="x.html">I don't turn blue</a>
```

```
a[target] {  
    color:blue;  
}
```

select <a> who has "target" attribute

Select elements under some “special state”

```
a:hover {  
    color:blue;  
}
```

<a> turns blue when hovered over by mouse.

Pseudo classes

:active	an activated or selected element
:focus	an element that has the keyboard focus
:hover	an element that has the mouse over it
:link	a link that has not been visited
:visited	a link that has already been visited
:first-letter	the first letter of text inside an element
:first-line	the first line of text inside an element
:first-child	an element that is the first one to appear inside another

How to select this?

```
<p>A normal paragraph</p>  
<p>I want to be different!</p>
```

Hard to find a selector that distinguishes these two.

The ugly way

```
<p>A normal paragraph</p>  
<p style="color:blue;">I want to be different!</p>
```

The better way: use **id** or **class** attributes

```
<p>A normal paragraph</p>
```

```
<p id="different">I want to be different!</p>
```

```
#different {  
    color:blue;  
}
```

Select the element with id "different"

Use class

```
<p>A normal paragraph</p>
```

```
<p class="different">I want to be different!</p>
```

```
.different {  
    color:blue;  
}
```

Select the element with class "different"

id vs class

- id must be unique, i.e., one id can only be used once by a **single** element.
- one class name can be used by **multiple** elements

SELECT ALL THE THINGS



```
* {  
    color:blue;  
}
```

```
p * {  
    color:blue;  
}
```

Full list of CSS selectors

→ http://www.w3schools.com/cssref/css_selectors.asp

CSS selector (fun) exercises

→ <http://flukeout.github.io/>

Takeaway: what are the pros of using **selectors** compared to

```
<p>  
<font face="Arial">Shashdot.</font>  
News for <b>nerds!!</b> You will <i>never</i>, <u>EVER</u>  
be  
<font size="+4" color="red">BORED</font> here!  
</p>
```

HTML

- one selector can select multiple elements, so don't need write the same style multiple times.
- one element can be of multiple class and combine multiple styles
- In general, it is better organized and more flexible
- Javascript works well with selectors, too.

Cascading Style Sheets



What is **cascading** about?

Cascading

- More than one stylesheet rules could be applied to an HTML element
- We need a way to determine which rule actually has effect.
- The general rule is to cascade down from more **general** rules to more **specific** rules, i.e., **the more specific rule is chosen.**

Example

```
<div>  
  <p>What's my colour?</p>  
</div>
```

```
p {  
  color:red;  
}
```

```
div p {  
  color:blue;  
}
```

This rule is more specific, so the colour is blue.

Cascading: more details

- Later rules override earlier rules (with everything else being equal)
- Closer rules override farther rules
 - ◆ inline rules override...
 - ◆ rules embedded in the head, which overrides
 - ◆ rules defined in external stylesheet
- more specific rules override more general rules
- an element inherits the style of its container
- The exact calculation: <http://www.bloobery.com/indexdot/css/topics/cascade.htm>
- Your code should be so simple and clear that no complicated calculation is need to determine which one overrides.

We're done talking about selectors

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

Let's talk about properties and values.

the style properties that you can set

colours

```
p {  
  color: red;  
  background-color: yellow;  
}
```

CSS

This paragraph uses the style above

output

property	description
color	color of the element's text
background-color	color that will appear behind the element

Specifying colours

```
p { color: red; }  
h2 { color: rgb(128, 0, 196); }  
h4 { color: #FF8800; }
```

CSS

This paragraph uses the first style above

This h2 uses the second style above.

This h4 uses the third style above.

output

- **color names:** aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white (white), yellow
- **RGB codes:** red, green, and blue values from 0 (none) to 255 (full)
- **hex codes:** RGB values in base-16 from 00 (0, none) to FF (255, full)

Fonts

property	description
font-family	which font will be used
font-size	how large the letters will be drawn
font-style	used to enable/disable italic style
font-weight	used to enable/disable bold style

Full list of font properties: http://www.w3schools.com/css/css_font.asp

Font family

```
p {  
font-family: Georgia;  
}  
h2 {  
font-family: "Courier New";  
}
```

CSS

This paragraph uses the first style above.

This h2 uses the second style above.

output

- ❑ Enclose multi-word font names in quotes

Font family

```
p {  
  font-family: Garamond, "Times New Roman", serif;  
}
```

CSS

This paragraph uses the above style.

output

- We can specify multiple fonts from highest to lowest priority
- Generic font names:
 - ▣ serif, sans-serif, cursive, fantasy, monospace
- If the first font is not found on the user's computer, the next is tried
- Placing a generic font name at the end of your font-family value, ensures that every computer will use a valid font

Font size

```
p {  
    font-size: 24pt;  
}
```

CSS

This paragraph uses the style above.

output

- units: pixels (**px**) vs. point (**pt**) vs. m-size (**em**)

16px, 16pt, 1.16em

- vague font sizes: xx-small, x-small, small, medium, large, x-large, xx-large,
smaller, **larger**

- percentage font sizes, e.g.: 90%, **120%**

Font size

```
p {  
    font-size: 24pt;  
}
```

CSS

This paragraph uses the style above.

output

- ❑ **pt** specifies a number of points, where a point is 1/72 of an inch onscreen
- ❑ **px** specifies a number of pixels on the screen
- ❑ **em** specifies number of m-widths, where 1 em is equal to the font's current size

font-weight, font-style

```
p {  
  font-weight: bold;  
  font-style: italic;  
}
```

CSS

This paragraph uses the style above.

output

- ❑ Either of the above can be set to normal to turn them off

CSS properties for text

property	description
text-align	alignment of text within its element
text-decoration	decorations such as underlining
line-height, word-spacing, letter-spacing	gaps between the various portions of the text
text-indent	indents the first letter of each paragraph

Full list of text properties: http://www.w3schools.com/css/css_text.asp

text-align

```
blockquote { text-align: justify; }  
h2 { text-align: center; }
```

CSS

The Gollum's Quote

We wants it, we needs it. Must have the precious. They stole it from us.
Sneaky little hobbitses. Wicked, tricky, false!

- output*
- **text-align** can be left, right, center, or justify



text-decoration

```
p {  
  text-decoration: underline;  
}
```

CSS

This paragraph uses the style above.

output

- ❑ can also be overline, ~~line-through~~, blink, or none
- ❑ effects can be combined:

```
text-decoration: overline underline;
```

CSS properties for background

property	description
background-color	color to fill background
background-image	image to place in background
background-position	placement of bg image within element
background-repeat	whether/how bg image should be repeated
background-attachment	whether bg image scrolls with page
background	shorthand to set all background properties

background-image

```
body {  
  background-image: url("images/draft.jpg");  
}
```

CSS

This is the first paragraph

This is the second paragraph...

It occupies 2 lines

- background image/color fills the element's content area

background-repeat

```
body {  
background-image: url("images/draft.jpg");  
background-repeat: repeat-x;  
}
```

CSS

This is the first paragraph

This is the second paragraph..

It occupies 2 lines

- ❑ can be repeat (default), repeat-x, repeat-y, or no-repeat

background-position

```
body {  
  background-image: url("images/draft.jpg");  
  background-repeat: no-repeat;  
  background-position: 370px 20px;  
}
```

CSS

This is the first paragraph

This is the second paragraph...

It occupies 2 lines

DRAFT

- ❑ value consists of two tokens, each of which can be top, left, right, bottom, center, a percentage, or a length value in px, pt, etc.
- ❑ value can be negative to shift left/up by a given amount

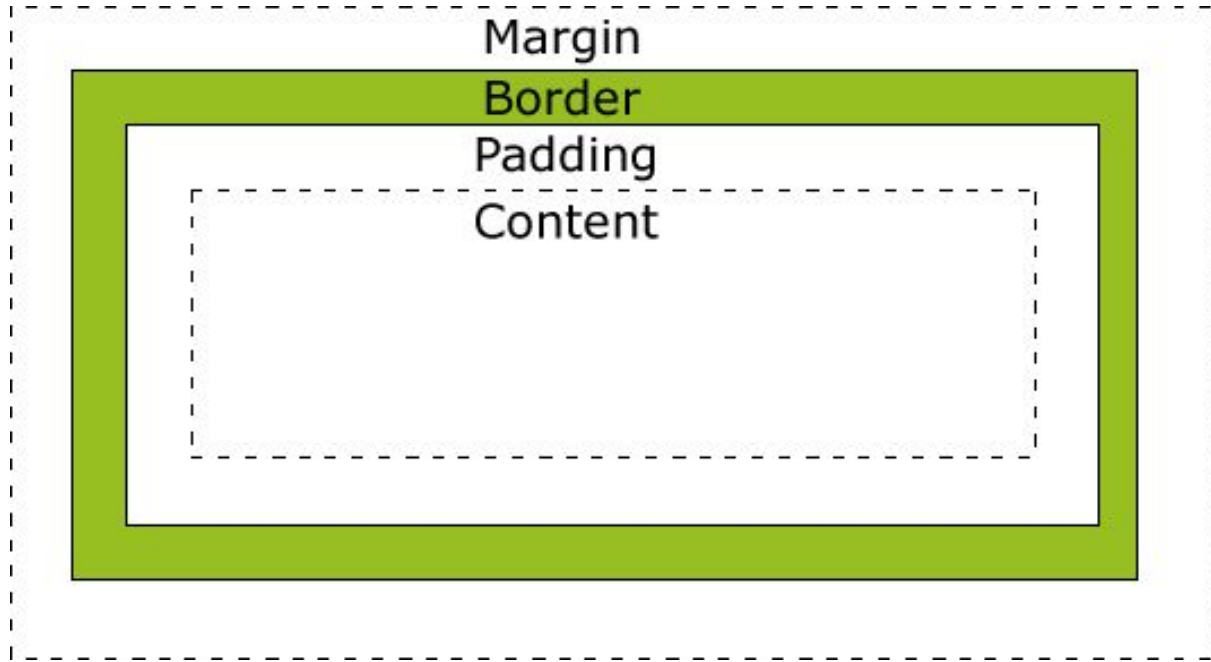


Just look it up here when needed

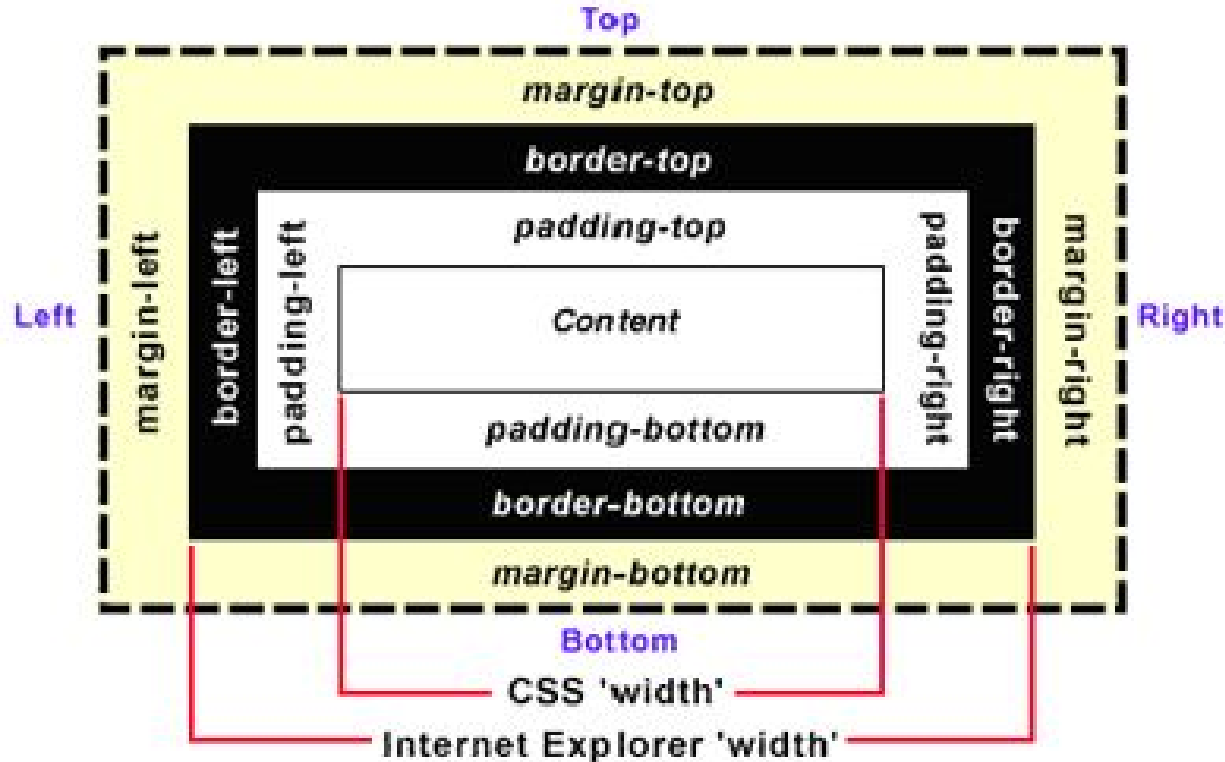
<http://www.w3schools.com/css/default.asp>

Page Layout with CSS

The CSS Box Model



The CSS Box Model: more details



Related properties of the box model

```
padding //for all 4 sides  
padding-left  
padding-right  
padding-top  
padding-bottom
```

```
margin //for all 4 sides  
margin-left  
margin-right  
margin-top  
margin-bottom
```

```
border: 5px solid red;
```

width, height: note this is only for the content part, the real width/height of the whole box is more than the defined value (+padding+border+margin)

More about width

```
p { width: 350px; background-color: yellow; }  
h2 { width: 50%; background-color: aqua; }
```

CSS

width changes with
browser size

This paragraph uses the first style above

An h2 heading

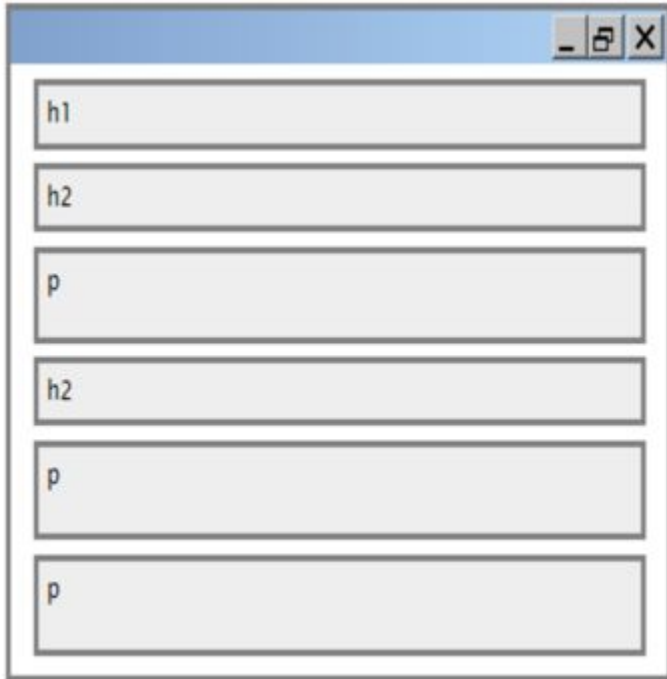
output

limit the range of
change

property	description
width, height	how wide or tall to make this element (block elements only)
max-width, max-height, min-width, min-height	max/min size of this element in given dimension

Float

The normal document flow

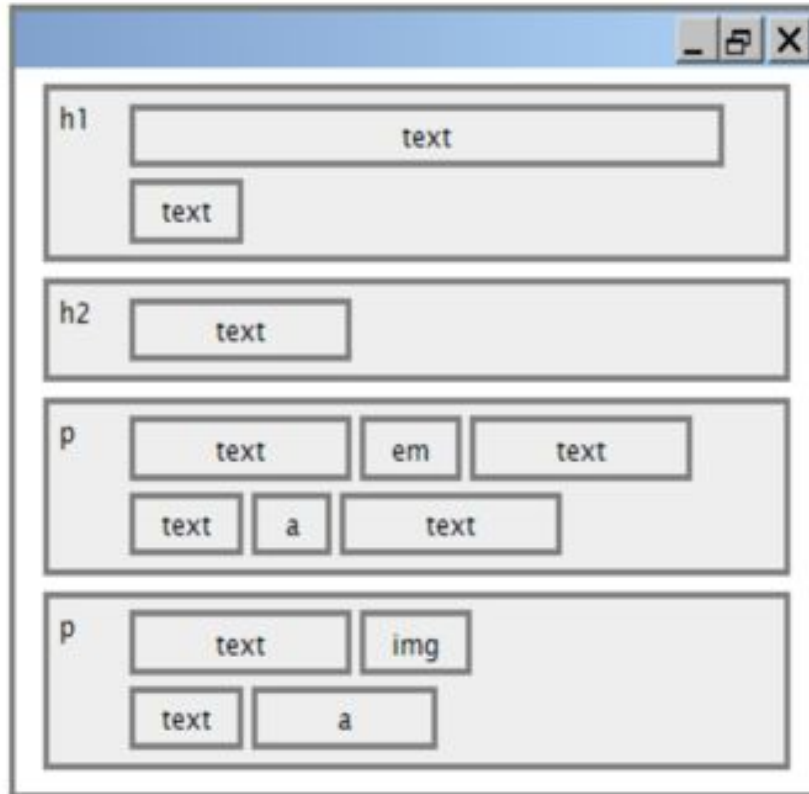


block elements



inline elements

The normal document flow: with both



Float

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

CSS

Ghostbusters is a 1984 American science fiction comedy film written by co-stars Dan Aykroyd and Harold Ramis about three eccentric New York City parapsychologists-turned-ghost capturers.

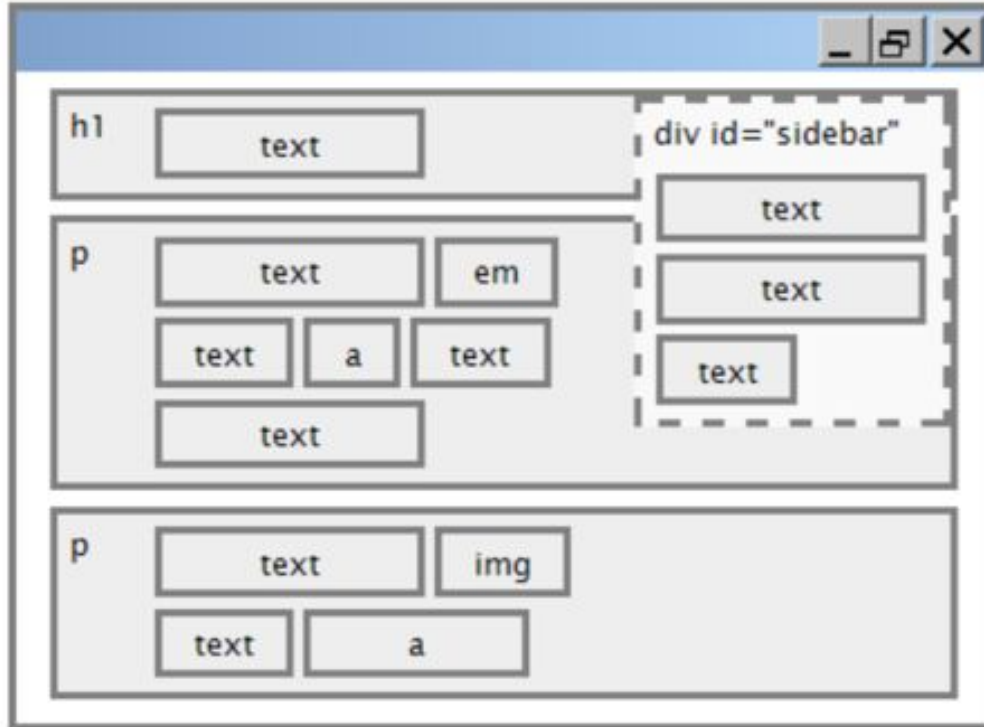


output

property	description
float	side to hover on; can be left, right, or none (default)

removed from normal document flow; underlying text wraps around as necessary

document flow with float



clear

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

Mario is a fictional character in his video game series. Serving as Nintendo's mascot and the main protagonist of the series, Mario has appeared in over 200 video games since his creation



Super Mario Fan Site!

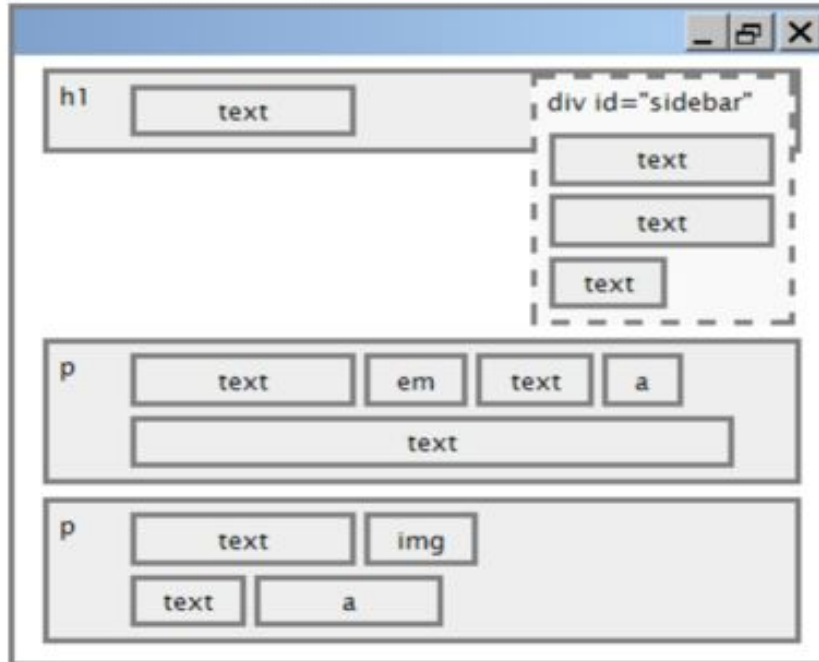
output

disallows the floating element to overlap with the element that “clears”

document flow with clear

```
div#sidebar { float: right; }  
p { clear: right; }
```

CSS



common bug: float with no 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

common bug: container too short

```
<p>
Mario is a fictional character in his video game
series.....</p>
```

HTML

```
p { border: 2px dashed black; }
img { float: right; }
```

CSS

Mario is a fictional character in his video game series.
Serving as Nintendo's mascot and the main protagonist
of the series, Mario has appeared in over 200 video
games since his creation.



output

fix it: using overflow: hidden (or auto)

```
p { border: 2px dashed black; overflow: hidden; }
```

^{CSS}
Mario is a fictional character in his video game series. Serving as Nintendo's mascot and the main protagonist of the series, Mario has appeared in over 200 video games since his creation.



output

Multi-column layout

```
<div>
  <p>first paragraph</p>
  <p>second paragraph</p>
  <p>third paragraph</p>
  Some other text that is important
</div>
```

HTML

```
p { float: right; width: 25%; margin: 0.5em;
border: 2px solid black; }
div { border: 3px dotted green; overflow: hidden; }
```

CSS

Some other text that is important

third paragraph

second paragraph

first paragraph

output

Position

The position property

```
div#ad {  
    position: fixed;  
    right: 10%;  
    top: 45%;  
}
```

CSS

property	value	description
position	static	default position
	relative	offset from its normal static position
	absolute	a fixed position within its containing element
	fixed	a fixed position within the browser window
top, bottom, left, right	positions of box's corners	

MUST READ & TRY

great tutorial with positioning examples

<http://www.barelyfitz.com/screencast/html-training/css/positioning/>

display and visibility

display

```
h2 { display: inline; background-color: yellow; }
```

CSS

This is a heading **This is another heading**

output

property	description
display	sets the type of CSS box model an element is displayed with

- **values:** none, inline, block, run-in, compact, ...

visibility

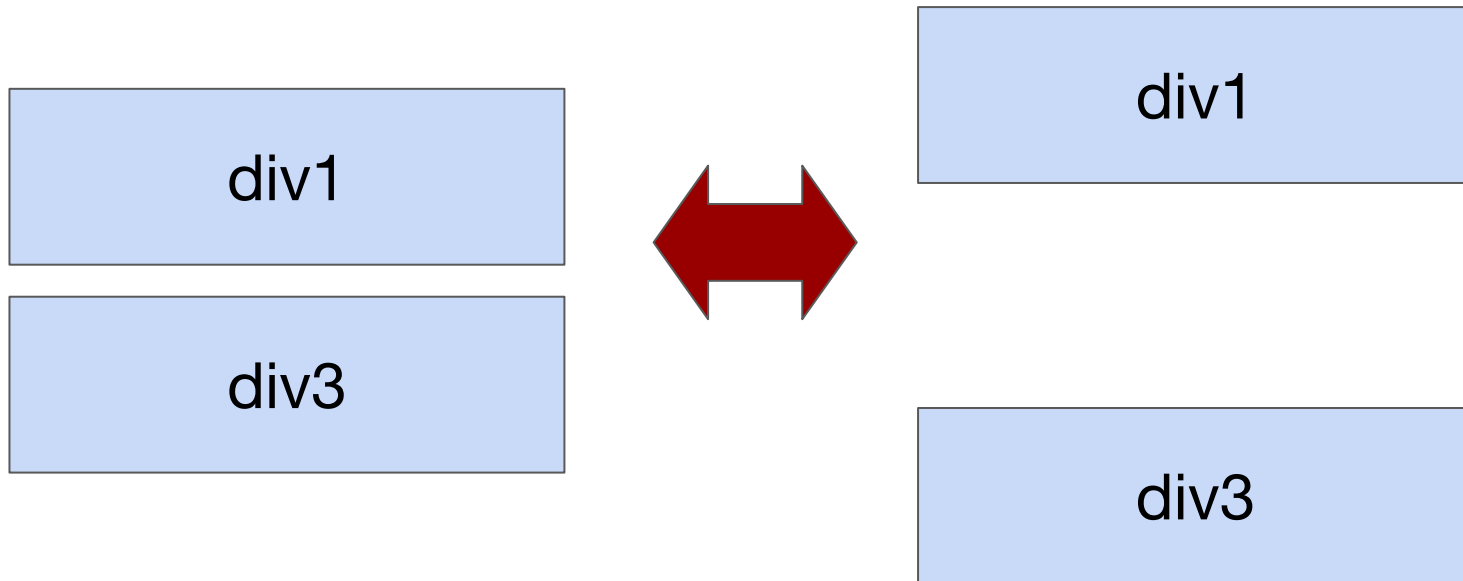
```
p.secret {  
    visibility: hidden;  
}
```

CSS

output

display:none; vs visibility:hidden;

- display:none; makes an element take NO space
- visibility:hidden; still takes the space but just you can't see it



Takeaway: box model, doc flow...

Before you learn these things, you may have had your CSS kind-of working by try-change-it-and-see-what-happens (change margin or padding? don't care), that typically leads to broken ugly layouts, without you realizing it.

Now you learned these things, you should do everything in the **principled** way.

CSS3

the latest standard of CSS

CSS3

- CSS3 is completely backward-compatible with old version of CSS, and added a bunch of new features, such as
 - ◆ round corners
 - ◆ gradients
 - ◆ shadows
 - ◆ animation
 - ◆ media queries

Rounded corner

We used to have to do this:

```
-moz-border-radius: 10px 5px;  
-webkit-border-top-left-radius: 10px;  
-webkit-border-top-right-radius: 5px;  
-webkit-border-bottom-right-radius: 10px;  
-webkit-border-bottom-left-radius: 5px;  
border-radius: 10px 5px;
```

Now we do this

```
border-radius: 10px 5px;
```

Rounded corners!

Media Query

Useful for **responsive design**, i.e., change the page style dynamically according to the screen size.

For example:

```
@media screen and (min-width: 480px) {  
    body {  
        background-color: lightgreen;  
    }  
}
```

demo link: http://www.w3schools.com/css/css3_mediaqueries.asp

CSS3 Animation

No need of Javascript.

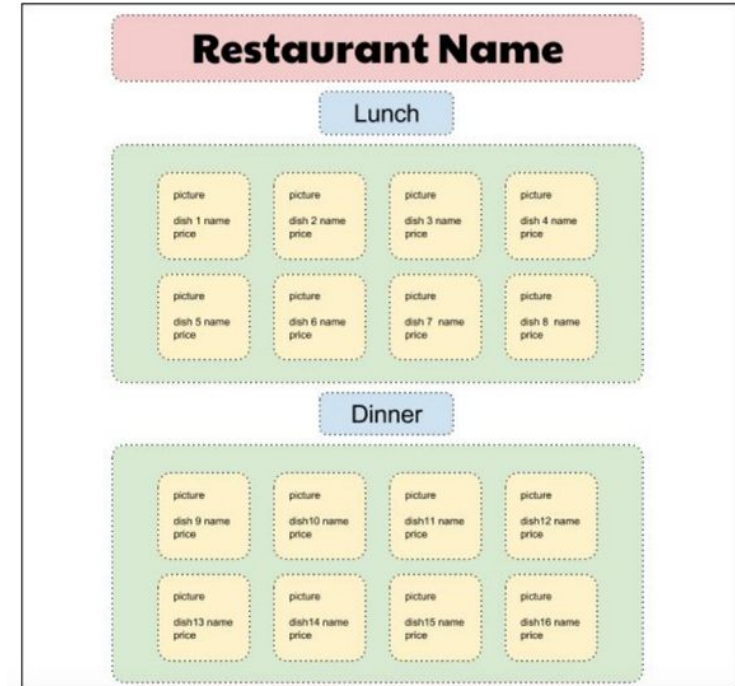
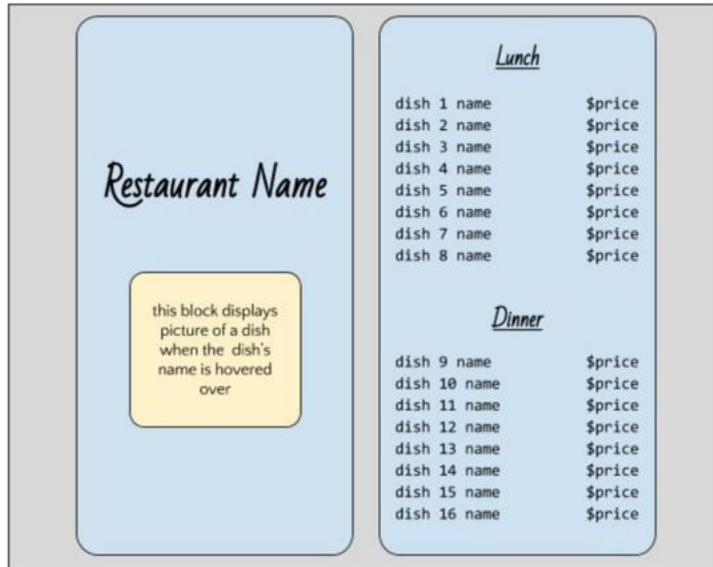
Demo

<https://daneden.github.io/animate.css/>

CSS
IS
AWESOME

Assignment 1

→ One HTML, two stylesheets



Assignment 1

- make sure to satisfy all the requirement
 - try to make it good-looking
 - can add additional features, without violating any requirements
 - Ask me whenever there is something unclear.
-
- For source code management, the department can only do SVN right now, which is not good
 - So, recommendation: use **Bitbucket** (<http://bitbucket.org>) to create a shared **GIT** or **Mercurial** repository with your partner.
 - Make you project **PRIVATE!**

The minimalist GIT tutorial

The GIT book: <https://git-scm.com/book/en/v2>

Why you should use git:

<https://www.youtube.com/watch?v=4XpnKHJAok8>

GIT	SVN
<code>git clone <repo URL></code>	<code>svn checkout <repo URL></code>
<code>git pull origin master</code>	<code>svn update</code>
<code>git add <filename></code>	<code>svn add <filename></code>
<code>git commit -am "<log text>"</code> <code>git push origin master</code>	<code>svn commit -m "<log text>"</code>
<code>git status</code>	<code>svn status</code>

Google Font

Easy way to get fancy fonts.

<https://www.google.com/fonts>

You can use it for Assignment 1.

Today we learned

→ CSS

Next week:

→ Javascript