
Welcome to HTML and CSS Class 4!

Slides at: <http://goo.gl/fPvVVS>

Please open your work from last week:

1. Open Sublime
2. Drop last week's folder on Sublime to open it as a project
3. Open index.html & style.css in Sublime
4. Open index.html in Chrome
5. Open the Chrome Inspector



HTML and CSS 101

Class 4

Web Safe Fonts

1. 13 Web Safe fonts
2. Example font stacks:
http://www.w3schools.com/cssref/css_websafe_fonts.a
3. Always end fontstack with serif, sans-serif, or monospace

```
p {  
  font-family: Tahoma, Geneva, sans-serif;  
}
```

Google Web Fonts

1. Easiest way to include fancy web fonts
2. Link to the google stylesheet in the head
3. Separate multiple Google fonts with pipe | character
4. Fonts do add load time, so use sparingly
5. Fonts will ALWAYS render differently across browsers
6. Add font stylesheets before your actual stylesheet or reset (cascade)

```
<head>
<link href='http://fonts.googleapis.com/css?family=Righteous|Cherry+Swash' rel='stylesheet' type='text/css'>
(reset and other stylesheets here)
</head>
```

```
.class1 {
  font-family: "Cherry Swash", times, serif;
}

.class2 {
  font-family: "Righteous", arial, sans-serif;
}
```

This is what Cherry Swash looks like.

This is what Righteous looks like.

Let's Develop It

Find two Google web fonts you would like to use & include on your site

1. **Google Web Fonts**
2. Include the stylesheet in your head section
3. Add the font to an element in your stylesheet

```
<link href='http://fonts.googleapis.com/css?family=Righteous|Cherry+Swash' rel='stylesheet' type='text/css'>
```

Static Positioning

HTML elements are positioned static by default.

Static elements are positioned in the normal flow of the page

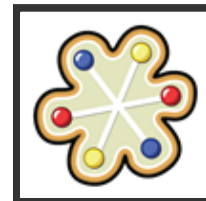
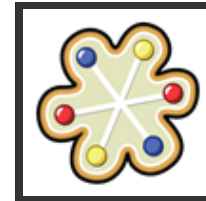
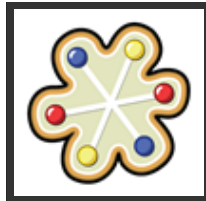
Static elements ignore top, bottom, right or left property specifications.

Static Positioning

In normal flow, inline boxes flow from left to right, wrapping to next line when needed.

```
  
  
  
...  
  

```



Static Positioning

In normal flow, block boxes flow from top to bottom, making a new line after every box.

```
<p>Greetings</p>  
<p>Hello</p>  
<p>Hi there!</p>
```

Greetings

Hello

Hi there!

Let's Develop It

Let's create some statically positioned divs

```
<div class="diva">A</div>  
<div class="divb">B</div>  
<div class="divc">C</div>
```

Let's give those divs some style! Make one red, gray, and blue. Give each the same margin, padding, and width per below.

```
.diva {  
  background-color: red;  
  color: white;  
  margin: 5%;  
  padding: 40px;  
  width: 20%;  
}
```

Relative Positioning

- Takes the element out of the normal flow, allowing it to be moved to the top, left, right or bottom.
- Does not affect the elements surrounding it.
- Makes an element a "positioning context" in which to position other elements relative to it.
- Relative positioning and absolute positioning are used together.

Relative Positioning

The "relative" value will still put the element in the normal flow, but then offset it according to top/left/right/bottom properties.

```
.relative{  
  position: relative;  
  left: 80px;  
  top: 20px;  
  height: 100px;  
  background-color: yellow;  
}
```

Hello, hi!

Let's Develop It

Let's make our second `.divb` relatively positioned.

```
.divb {  
  background-color: gray;  
  color: white;  
  margin: 5%;  
  padding: 40px;  
  position: relative;  
  top: -.5em;  
  right: 10%;  
  bottom: auto;  
  left: auto;  
  width: 20%;  
}
```

Once it is relatively positioned, we can add `top`, `right`, `bottom`, and `left` as indicated above to offset it from the normal flow.

Absolute Positioning

- Positions element outside of the normal flow.
- An absolutely positioned element is offset from its container block (positioned relative)
- Its container block is the first element that has a position other than static.
- If no such element is found, the container block is `<html>`.
- Other elements act as if it's not there.
- Determined by its offset values in the properties top, bottom, right and left.

Absolute Positioning

The "absolute" value will take the element out of the normal flow and position it in relation to the window (or the closest non-static element).

```
.top {  
  position: absolute;  
  top: -40px;  
  right: 10px;  
  background-color: yellow  
}
```

```
.bottom {  
  position: absolute;  
  bottom: -40px;  
  left: 60px;  
  background-color: green  
}
```

Let's Develop It

Let's make our third .divc absolutely positioned to the top of the containing element.

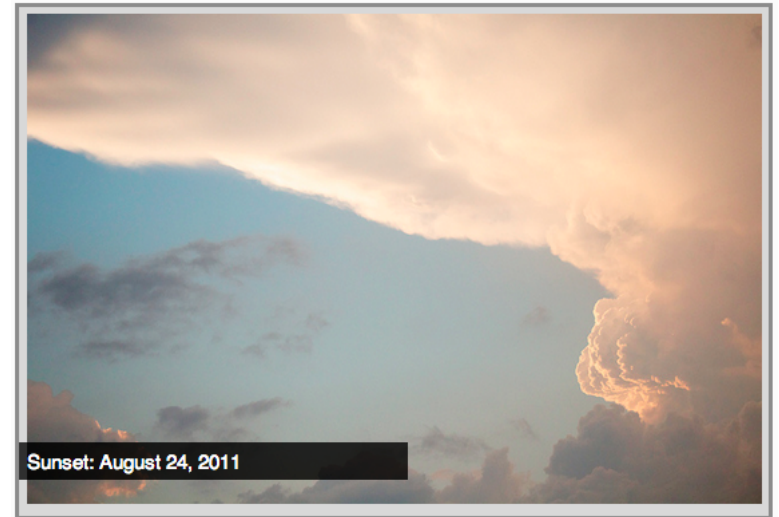
```
.divc {  
  background-color: blue;  
  color: white;  
  margin: 5%;  
  padding: 40px;  
  position: absolute;  
  top: 0;  
  right: auto;  
  bottom: auto;  
  left: auto;  
  width: 20%;  
}
```

If there is no containing element for divc, the containing element is <html>

More: Absolute Positioning

Here's an example of an image with a caption absolutely positioned over top of it.

The containing div has a position of relative, and the caption has a position of absolute.



Z-index

When you use positioning to move elements out of the normal flow of content, elements can overlap. You can change the order of overlapping with z-index.

The element with highest z-index goes on top.

```
.bottom{  
  position: absolute;  
  bottom: 10px;  
  left: 60px;  
  background-color: yellow;  
}
```

```
.top{  
  position: absolute;  
  bottom: 15px;  
  left: 60px;  
  background-color: green;  
  z-index: 2;  
}
```

Let's Develop it!

Let's add some more positioning.

1. Let's create a div with a class of "pos". Position the div relatively.
2. Add a medium sized image inside the div with a caption below. The caption should have a class of "caption".
3. Position that caption absolutely overtop the image.

```
.pos {  
  background-color: yellow;  
  padding: 0;  
  position: relative;  
  top: 0;  
  width: 20%;  
}  
  
.caption {  
  position: absolute;  
  top: -2em;  
  width: 80%;  
  z-index: 9;  
}
```

Float

- "Floating" an element takes it in the normal flow, as far to the left or right of it's containing element as possible.
- Any other elements, such as paragraphs or lists, will wrap around the floated element.
- Always specify a width when floating an element, otherwise the element is likely to take up the whole page and not appear floated.
- You can specify a) whether an element is floated or not, and b) which side it floats on.

Float: Example

Below a <blockquote> is floated to the left, allowing text to wrap around it on the right

That's when they discovered Are You a Human. After switching to PlayThru, HiredMyWay saw a 40% decrease in the amount of time it takes an average user to sign up for the site, from 4 minutes, 24 seconds to 2 minutes, 39 seconds. That's almost **two minutes saved** per user.

“The technology behind Are You a Human seemed like a breath of fresh air.”

—Wes Weiler, CMO

Even better, after implementing PlayThru, the percentage of HiredMyWay users who complete their profile after signing up went from 60% to 80%.

That's no surprise—CAPTCHA's effectiveness is well-documented in studies indicating that one in four people will

And PlayThru didn't just improve HiredMyWay's numbers—it also improved their overall customer experience. “Coming onto my site, and that's a bad experience,” says Wes. “Everything's an emotional roller coaster.”

Float

```
.float{  
  float:left;  
  width:200px;  
  background:yellow;  
}
```

Hi, I'm a yellow box
with black text.
I like to hang out on
the left side.

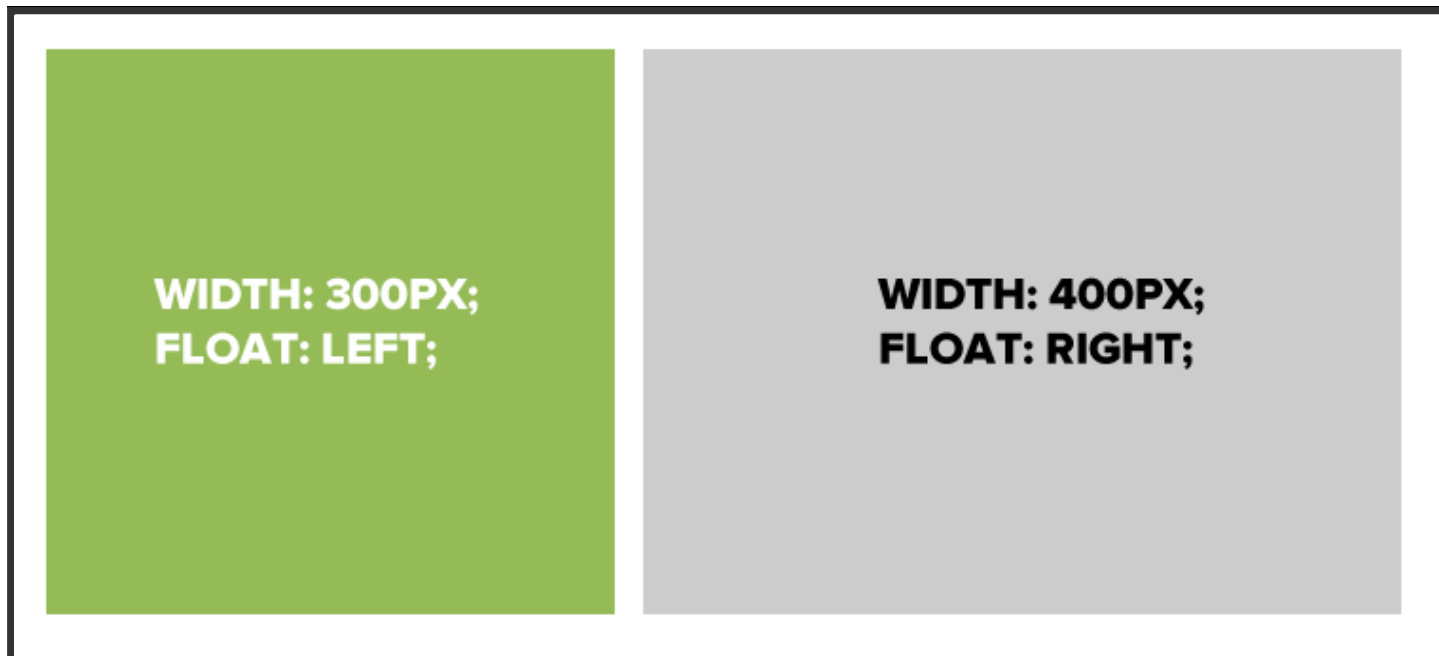
Not floating element

Not floating element

Not floating element with wrapping Lorem
ipsum dolor sit amet, consectetur adipisicing elit, sed do
eiusmod tempor incididunt ut labore et dolore magna
aliqua.

Using floats to place elements side by side

If you want two block level elements to be side by side, you may float both elements. One left, and one right.



Using float: left to place two or more elements side by side

If you want two or more block level elements to be side by side, you can float: left all elements.



But how
do you
stop the

floating?

Clear

- Clearing tells the element on which side (right, left, both) other elements **cannot** appear.
- If you had an image floated left, and you did not want the paragraph to appear next to it, you would add **clear: left;** to the paragraph.
- Clearing both sides makes sure floats don't flow past the clear element.

```
clear: right;  
clear: left;  
clear: both;
```

Clear

```
.float{  
  float:left;  
  width:100px;  
  background:yellow;  
}  
.clear-left{  
  clear:left  
}
```

hi Not floating element

hi Not floating element

hi

Non-floating element with a class of **.clear-left**

Let's Develop It

Let's float our sidebar and content areas so that they are side-by-side

1. Remember $\text{width} = \text{element width} + \text{padding} + \text{border}$
2. You may have to make the width smaller so everything fits
3. The header div should appear above and the footer div below the sidebar and content area.

Questions?



Keep in Touch

GDILA.org | [Facebook](#) | [Twitter](#)

More Learning

Upcoming Girl Develop It Classes

You are all **AWESOME!**

Give yourself a hand for making it through
HTML & CSS 101!



Survey

Please let us know how you liked the class and any suggestions you may have before you leave. Thank you! :)

<https://goo.gl/oRUYso>

I will send an email in the next day with the example files and PDF. Thank you for being a part of this class!

Class files and slides will be up for a month:

<http://goo.gl/fPvVVS>



Top
Down here