Layout and Grids

No screens

Prof. Lydia Chilton
COMS 4170
30 January 2019

Say your name
THEY DON’T EVEN HAVE A COLOR TO DESCRIBE HOW COLD IT’S GOING TO BE
COMS 4170 Goal 1

Build websites that suit the needs and abilities of users

The main goal of many websites is to display information users need.
How to guide users’ attention

Content

Metadata: title, description

Watch it next

Menu

Emails

Contact/chat
Layout in HTML

How to construct the layout of information in HTML and CSS
Web Browsers are in charge of rendering HTML
When you add elements, They either obey the line-wrap layout or break it.

SPAN elements {obey} line-wrap layout
Do `<div>` elements obey or break line-wrap layout?

`<div>` elements **break** line-wrap layout
When you use a new element, ask: does it obey or break line-wrap layout?

The best way to know what a layout will look like is to run it.
Today’s goal:
Remake this information layout in HTML and CSS
"I finished coding up my website..."

"It just doesn’t work yet!"
Iterative Style of Programming

What is the smallest unit of progress I can make?

Iteration

```html
<html>
  <body>
    Hello world!
  </body>
</html>
```

Goal

Does it look ok?
Step 0. Make a html page, connect some CSS

```html
<!DOCTYPE html>
<html>
<head>
<style>
  #test{
    color: blue;
  }
</style>
</head>
<body>
  <div id="test">Hello World!</div>
</body>
</html>
```
What should we do first?

Make a container
Step 1. Make a container div

```html
<div id="container">
</div>
```

```css
#container{
  width: 50px;
  height: 50px;
  background-color: grey;
}
```
Iterative Style of Programming

What’s the most **basic** thing I can do to make **necessary** progress towards my goal?

---

Iteration

<html>
<body>
    Hello world!
</body>
</html>

---

Goal

Does it look ok?
Does it look ok? What’s next?
Container div has realistic width.

```html
<div id="container">
</div>
```

```css
#container{
  width: 350px;
  height: 50px;
  margin: 50px;
  background-color: grey;
}
```
What’s next?
Blue facebook header

```html
<div id="container">
  <div id="fb_brand">
    FACEBOOK
  </div>
</div>
```

```css
#container{
  width: 350px;
  margin: 50px;
}

#fb_brand{
  background-color: blue;
  color: white;
}
```
What next?
Blue facebook header

```html
<div id="container">
  <div id="fb_brand">
    FACEBOOK
  </div>
</div>
```

```css
#container{
  width: 350px;
  margin: 50px;
}

#fb_brand{
  background-color: blue;
  color: white;
  padding-left: 40px;
  padding-top: 20px;
  padding-bottom: 20px;
}
```
What next?
Add the first post

```html
1 <div id="container">
2   <div id="fb_brand">
3     FACEBOOK
4   </div>
5   <div id="top_post">
6     FIRST POST
7   </div>
8 </div>
```
What next?
Add internal structure to the first post

```html
<div id="top_post">
  <div id="top_post_image">
  </div>
  <div id="top_post_text">
    TOP POST TEXT
  </div>
</div>

#top_post_image{
  height: 50px;
  width: 50px;
  background-color: green;
  display: inline-block;
}

#top_post_text{
  height: 30px;
  width: 250px;
  padding: 10px;
  border: 1px solid black;
  display: inline-block;
  vertical-align: top;
}
```
By default, DIV elements will flow down the page. display: block
To get them to flow across the page, you must set display: inline-block

```css
#top_post_image{
    height: 50px;
    width: 50px;
    background-color: green;
    /*display: inline-block;*/
}

#top_post_text{
    height: 30px;
    width: 250px;
    padding: 10px;
    border: 1px solid black;
    /*display: inline-block;*/
    vertical-align: top;
}
```
What next?
Add padding on the top post

```html
6  <div id="top_post">
7    <div id="top_post_image">
8    </div>
9    <div id="top_post_text">
10       TOP POST TEXT
11    </div>
12  </div>
```

```css
#top_post{
    padding: 10px;
}
```
What next?
Replies

```html
13  <div id="reply_container">
14      <div class="reply">
15          REPLY
16      </div>
17      <div class="reply">
18          REPLY
19      </div>
20      <div class="reply">
21          REPLY
22      </div>
23  </div>
```
What next?
Reply substructure: image and text

```html
14  <div class="reply">
15     <div class="reply_image">
16         </div>
17     <div class="reply_text">
18         REPLY 1
19     </div>
20     </div>

35  .reply_image{
36      height: 50px;
37      width: 50px;
38      background-color: green;
39      display: inline-block;
40      vertical-align: top;
41  }
42
43  .reply_text{
44      height: 30px;
45      width: 200px;
46      padding: 10px;
47      border: 1px solid black;
48      display: inline-block;
49      vertical-align: top;
50  }
```
What next?

Facebook

Top Post Text

Reply 1

Reply 2

Reply 3
Reply Container positioning

```html
13  <div id="reply_container">
14      <div class="reply">
15          <div class="reply_image">
16          </div>
17          <div class="reply_text">
18              REPLY 1
19          </div>
20      </div>
21  </div>
```

```
#reply_container{
    padding-left: 60px;
}
```
What next?
Reply Class background-color

```html
13  <div id="reply_container">
14    <div class="reply">
15      <div class="reply_image">
16      </div>
17      <div class="reply_text">
18        REPLY 1
19      </div>
20    </div>
21  </div>

39  .reply{
40    background-color: lightblue;
41    padding: 10px;
42    margin: 10px;
43  }
```
Reduce Reply Text Width

```html
13  <div id="reply_container">
14    <div class="reply">
15      <div class="reply_image">
16      </div>
17      <div class="reply_text">
18        REPLY 1
19      </div>
20    </div>
```
What next? Am I done?
<div class="reply">
  <div class="reply_image">
  </div>
  <div class="reply_text">
    REPLY 1 REPLY 1 REPLY 1 REPLY 1 REPLY 1 REPLY 1
    REPLY 1 REPLY 1
  </div>
</div>

.reply_text{
  height: 30px;
  width: 160px;
  padding: 10px;
  border: 1px solid black;
  display: inline-block;
  vertical-align: top;
}
<div class="reply">
  <div class="reply_image">
  </div>
  <div class="reply_text">
    REPLY 1 REPLY 1 REPLY 1 REPLY 1 REPLY 1 REPLY 1 REPLY 1 REPLY 1
    REPLY 1 REPLY 1
  </div>
</div>

.reply_text{
  /*height: 30px;*/
  width: 160px;
  padding: 10px;
  border: 1px solid black;
  display: inline-block;
  vertical-align: top;
}
Remove borders

```css
#top_post_text{
  height: 30px;
  width: 250px;
  padding: 10px;
  /*border: 1px solid black;*/
  display: inline-block;
  vertical-align: top;
}

.reply_text{
  width: 160px;
  padding: 10px;
  /*border: 1px solid black;*/
  display: inline-block;
  vertical-align: top;
}
```
After layout... what next?
Iterative Style of Programming helps you build a mental model of your code.

What’s the small unit of progress I can make?

```html
<html>
<body>
  Hello world!
</body>
</html>
```

Does it look ok?

Hello World!
Iterative Style of Programming helped me recover from errors one at a time (as I created them)

- Error 1: display inline-block
- Error 2: reply div padding
- Error 3: Reply div height
Once you have too many bugs, you won’t fix it.
Grids

Currently the dominant way to layout information
That worked BUT it would be much easier if we had a grid.
How to guide users attention:

We can implement conceptual groupings in a grid
Old school information layout

Symmetrical
Before: Symmetric Layout

Now: Asymmetric Grids Layout
The Bauhaus School of Design
Typographer Jan Tschichold made layouts inspired by Bauhaus.
You can convey importance of elements within the grid
You can also convey importance by changing the grid.
Not all the ideas of the Bauhaus school are new.

But there is freedom from the symmetric layout.
Twitter Bootstrap

An implementation of grid layout
How many **columns** do you see?
How wide is the middle (compared to the sides)?
All content goes in the main container
The container is divided into **12 columns**
You get to apportion those 12 columns
12 columns: split in 3, 6, 3
Bootstrap grid. How many columns?
Do the columns have the same rows?
Is there alignment within the column?
Can you use the original grid?
Can you use the original grid? NO
Grids within grids!!!
Bootstrap grid. Apportion 12 columns.
Add Bootstrap to your HTML

CSS & JS in `<head>` tag, before your own CSS
1. What will this produce?

```html
<div class="container">
  <div class="row">
    <div class="col-md-3">
      <img src="./columbia.jpg">
    </div>
    <div class="col-md-3">
      <img src="./columbia.jpg">
    </div>
    <div class="col-md-3">
      <img src="./columbia.jpg">
    </div>
    <div class="col-md-3">
      <img src="./columbia.jpg">
    </div>
  </div>
</div>
```
2. What will this produce?

```html
<div class="container">
  <div class="row">
    <div class="col-md-2">
      <img src="./columbia.jpg">
    </div>
    <div class="col-md-2">
      <img src="./columbia.jpg">
    </div>
    <div class="col-md-4">
      <img src="./columbia.jpg">
    </div>
    <div class="col-md-4">
      <img src="./columbia.jpg">
    </div>
  </div>
</div>
```
3. What will this produce?

```html
<div class="container">
  <div class="row">
    <div class="col-md-3">
      <div class="row">
        <div class="col-md-6">
          <img src="/cu.jpg">
        </div>
        <div class="col-md-6">
          <img src="/cu.jpg">
        </div>
      </div>
    </div>
    <div class="col-md-3">
      <img src="/cu.jpg">
    </div>
    <div class="col-md-6">
    </div>
  </div>
</div>
```
Bootstrap solves another problem
Responsive Design
Four Device sizes

- Large: 1200px
- Medium: 992px
- Small: 768px
- Extra-Small: < 768px

- Desktops
- Laptops
- Tablets
- Phones
What will this do on a phone?

```html
<div class="row">
  <div class="col-md-3 col-xs-6">
    <img src="/columbia.jpg">
  </div>
  <div class="col-md-3 col-xs-6">
    <img src="/columbia.jpg">
  </div>
  <div class="col-md-3 col-xs-6">
    <img src="/columbia.jpg">
  </div>
  <div class="col-md-3 col-xs-6">
    <img src="/columbia.jpg">
  </div>
</div>
```
What’s the difference in layout?

class="col-md-3"

class="col-md-3 col-xs-6"
In summary...
Web Browsers are in charge of rendering HTML.

Their default rendered is to do line-wrapping, with some elements breaking the line.
Iterative Style of Programming helps you build a mental model of your code.

What’s the **smallest** unit of progress I can make?

```html
<html>
  <body>
    Hello world!
  </body>
</html>
```

Does it look ok?
Grids are a layout help you convey importance.
Not all the ideas of the Bauhaus school are new.

But there is freedom from the symmetric layout.
Twitter Bootstrap is a widely used grid system.
Bootstrap allocates width into **twelve columns**
Responsive design helps you specify the layout for different sized screens.
Announcements

• Homework 2 due Friday @4pm on Courseworks
• Record Participation: Piazza -> Google Form