



Introduction to Internet Technology and Web Programming

Computer Science 103
Boston University
Vahid Azadeh-Ranjbar

These notes are based on the lecture notes provided by [Professor Susan Worst](#).



Lecture 7-1

BU CS 103, Spring 2019




Assignments

❖ Deadlines

- **HW 3** due was today at 6:00 pm.
- **Lab 5** due is on Friday 03/08 at midnight.

❖ Grades

- ✓ **HW 2** was posted on BB
- ✓ **Lab 4** Assignment was posted on BB.
- ✓ **Lab 5** Challenge was posted on BB.



Midterm Exam 1

- The exam may include up to 40 questions.
- You will see more coding questions in the midterm exam 1 in comparison to the practice exam.
- The midterm exam 1 will be fairly harder than the practice exam.

Midterm Exam 1

- The exam will be held at PHO 206 at Wed March 6, 2019 from 6:35pm to 7:35pm.



Today's Materials!

- Multicolumn columns layout:
 - ✓ CSS Positioning
 - ✓ Bootstrap for Multicolumn Layout
- Combining CSS selectors for more power



Limitations of Float

- ❖ It can only float elements left or right, not up or down



Next step

- ❖ What if you want more than two columns?
- ❖ Can we do it only by float/clear property?
- ❖ If not, how is it doable?

A Demonstration: 3 Columns Layout with Float and Clear Properties



3 Columns Layout with Float and Clear Properties:

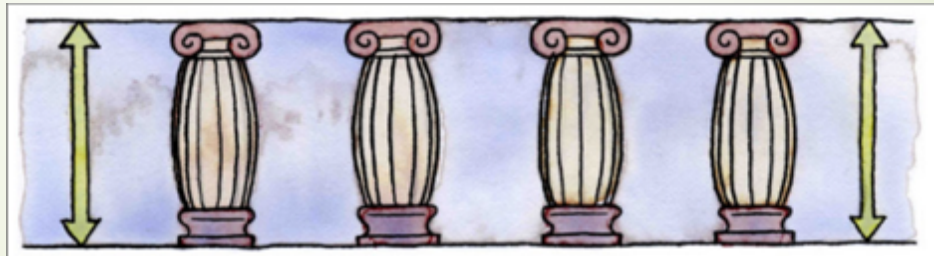
❖ Problem:

- Float property is sensitive to amount of the content. So the 3 columns layout will have 3 columns with different heights.



3 Columns Layout with Positioning

Equal Height Columns



3 Columns Layout with Positioning

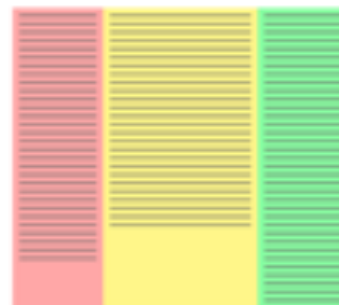
The Problem

Columns are all different heights depending on the content in each



The Solution

The height of all columns is equal to the height of the tallest column



3 Columns Layout with Positioning

1. Adding extra nested containers

```
<div id="container1">
  <div id="col1"> <p>Column 1C
  <div id="col2"><p>Column
  <div id="col3"><p>Column
</div>
```

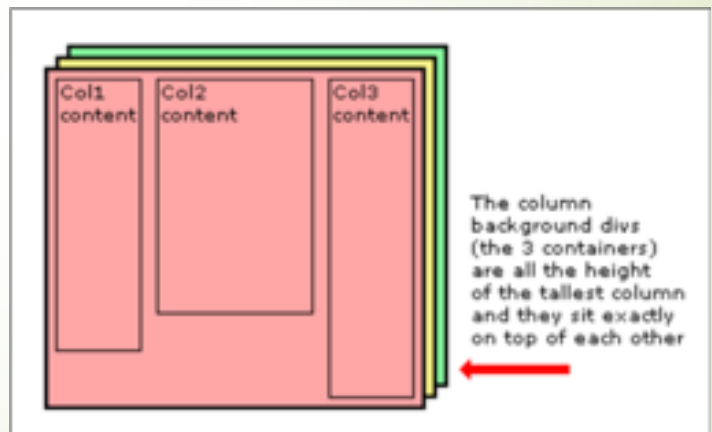


```
<div id="container3">
  <div id="container2">
    <div id="container1">
      <div id="col1"> <p>Column 1C
      <div id="col2"><p>Column 2Co
      <div id="col3"><p>Column 3Co
    </div>
  </div>
</div>
```

3 Columns Layout with Positioning

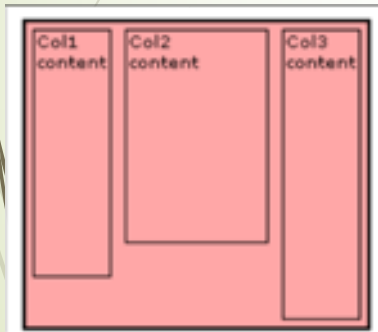
1. Adding extra nested containers

```
<div id="container3">
  <div id="container2">
    <div id="container1">
      <div id="col1"> <p>Column 1C
      <div id="col2"><p>Column 2Co
      <div id="col3"><p>Column 3Co
    </div>
  </div>
</div>
```



3 Columns Layout with Positioning

2. Specifying the colors for containers not columns

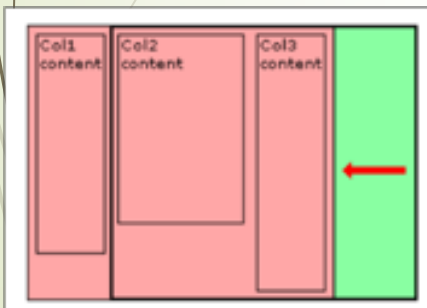


```
<div id="container3">
  <div id="container2">
    <div id="container1">
      <div id="col1"><p>Column 1C
      <div id="col2"><p>Column 2C
      <div id="col3"><p>Column 3C
    </div>
  </div>
</div>
```

```
#container3 {
  float:left;
  width:100%;
  background:green;
}
#container2 {
  float:left;
  width:100%;
  background:yellow;
}
#container1 {
  float:left;
  width:100%;
  background:red;
}
#col1 {
  float:left;
  width:30%;
}
#col2 {
  float:left;
  width:40%;
}
#col3 {
  float:left;
  width:30%;
}
```

3 Columns Layout with Positioning

3. Moving container 2 to the left by 30% using relative positioning

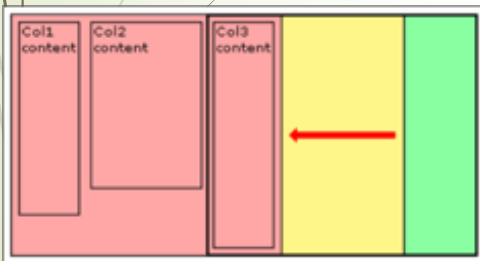


```
<div id="container3">
  <div id="container2">
    <div id="container1">
      <div id="col1"><p>Column 1C
      <div id="col2"><p>Column 2C
      <div id="col3"><p>Column 3C
    </div>
  </div>
</div>
```

```
#container3 {
  float:left;
  width:100%;
  background:green;
}
#container2 {
  float:left;
  width:100%;
  background:yellow;
  position:relative;
  right:30%;
}
#container1 {
  float:left;
  width:100%;
  background:red;
}
#col1 {
  float:left;
  width:30%;
}
#col2 {
  float:left;
  width:40%;
}
#col3 {
  float:left;
  width:30%;
}
```


3 Columns Layout with Positioning

4. Moving container 1 to the left by 40% using relative positioning

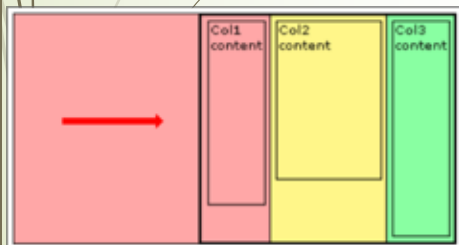


```
<div id="container3">
  <div id="container2">
    <div id="container1">
      <div id="col1"> <p>Column 1C</p>
      <div id="col2"> <p>Column 2C</p>
      <div id="col3"> <p>Column 3C</p>
    </div>
  </div>
</div>
```

```
#container3 {
  float:left;
  width:100%;
  background:green;
}
#container2 {
  float:left;
  width:100%;
  background:yellow;
  position:relative;
  right:30%;
}
#container1 {
  float:left;
  width:100%;
  background:red;
  position:relative;
  right:40%;
}
#col1 {
  float:left;
  width:30%;
}
#col2 {
  float:left;
  width:40%;
}
#col3 {
  float:left;
  width:30%;
}
```

3 Columns Layout with Positioning

5. Moving the content back into each column by 70% using relative positioning



```
<div id="container3">
  <div id="container2">
    <div id="container1">
      <div id="col1"> <p>Column 1C</p>
      <div id="col2"> <p>Column 2C</p>
      <div id="col3"> <p>Column 3C</p>
    </div>
  </div>
</div>
```

```
#container3 {
  float:left;
  width:100%;
  background:green;
}
#container2 {
  float:left;
  width:100%;
  background:yellow;
  position:relative;
  right:30%;
}
#container1 {
  float:left;
  width:100%;
  background:red;
  position:relative;
  right:40%;
}
#col1 {
  float:left;
  width:30%;
  position:relative;
  left:70%;
}
#col2 {
  float:left;
  width:40%;
  position:relative;
  left:70%;
}
#col3 {
  float:left;
  width:30%;
  position:relative;
  left:70%;
}
```

A Demonstration: 3 Columns Layout with Positioning



Bootstrap for Multicolumn Layout

- Last class, we showed that making a **two-column layout** might be easy (only by using float and clear properties)
- But making a multicolumn layout is very sophisticated
 - ✓ **Bootstrap will help us to make a multicolumn layout without taking care of the details.**

What is Bootstrap?

- ❖ Bootstrap is an open source toolkit for developing with HTML, CSS, and JS.
- ❖ Applications:
 - multicolumn layout
 - responsive design
 - extensive prebuilt components, and
 - powerful plugins built on jQuery.

<https://getbootstrap.com/>

Bootstrap for Multicolumn Layout

We can do it in four steps:

Step 1: Link bootstrap CSS file to your HTML file

- Go to **<https://getbootstrap.com/>**
- Click on **BootstrapCDN** section
- Go to **complete CSS** section
- Copy the link tag (from HTML section) and paste it into the head of your html file.

Bootstrap for Multicolumn Layout

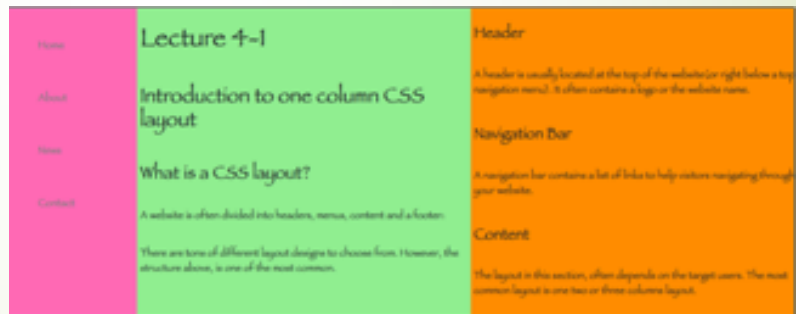
Step 2: HTML file

- **Mark up** the elements that you want to make a multicolumn layout for them.

```
<nav>
...
</nav>

<div id="content1">
...
</div>

</div id="content2">
...
</div>
```



Bootstrap for Multicolumn Layout

Step 3: HTML file

- ❖ Wrap them up with 2 div elements with **predefine** classes called **class="container-fluid"** and **class="row"**

```
<div class="container-fluid">
  <div class="row">
    <nav>... </nav>
    <div id="content1"> ... </div>
    <div id="content2"> ... </div>
  </div>
</div>
```

Bootstrap for Multicolumn Layout

Step 4: HTML file

- ❖ Now give some predefined classes to the columns
- ❖ They are written in the following form:
`class="col-x"`
- ❖ Where x can be a number between **1** and **12**.
- ❖ Needs careful calculation

```
<div class="container-fluid">
<div class="row">
<nav class="col-2"> ...</nav>
<div id="content1" class="col-5"> ... </div>
<div id="content2" class="col-5"> ... </div>
</div>
</div>
```

Bootstrap for Multicolumn Layout

A demonstration for making a **two** column layout

What the question marks have to be?

```
<div class="container-fluid">
<div class="row">
<nav class="col-?"> ...</nav>
<div class="col-?"> ... </div>
</div>
</div>
```

Bootstrap for Multicolumn Layout

A demonstration for making a **two** column layout

What the question marks have to be?

For example: **3** and **9**

```
<div class="container-fluid">
<div class="row">
<nav class="col-3"> ...</nav>
<div class="col-9"> ... </div>
</div>
</div>
```

Bootstrap for Multicolumn Layout

A demonstration for making a **three** column layout

What the question marks have to be?

```
<div class="container-fluid">
<div class="row">
<nav class="col-?"> ...</nav>
<div id ="content1" class="col-?"> ... </div>
<div id ="content2" class="col-?"> ... </div>
</div>
</div>
```

Bootstrap for Multicolumn Layout

A demonstration for making a **three** column layout

What the question marks have to be?

For example: 2, 5, 5

```
<div class="container-fluid">
<div class="row">
<div class="col-2"> ...</div>
<div id="content1" class="col-5"> ... </div>
<div id="content2" class="col-5"> ... </div>
</div>
</div>
```

Bootstrap for Multicolumn Layout

A demonstration for making a **five** column layout

What the question marks have to be?

```
<div class="container-fluid">
<div class="row">
<div class="col-?"> ...</div>
<div id="content1" class="col-?"> ... </div>
<div id="content2" class="col-?"> ... </div>
<div id="content3" class="col-?"> ... </div>
<div id="content4" class="col-?"> ... </div>
</div>
</div>
```

Bootstrap for Multicolumn Layout

A demonstration for making a **Five** column layout

What the question marks have to be?

For example: 2, 2, 3, 3, 2

```
<div class="container-fluid">
<div class="row">
<nav class="col-2"> ...</nav>
<div id="content1" class="col-2"> ... </div>
<div id="content2" class="col-3"> ... </div>
<div id="content3" class="col-3"> ... </div>
<div id="content4" class="col-2"> ... </div>
</div>
</div>
```

Bootstrap for Multicolumn Layout & Responsive Design

We only need to manipulate step 4

Step 4: HTML file

- ❖ Give some other predefined classes to the columns

`class="col-##-x"`

- ❖ **x** can be a number between **1** and **12**.
- ❖ **##** can be **xs** or **sm** or **md** or **lg** corresponding to different screen sizes

<code>.col-xs-\$</code>	Extra Small	Phones Less than 768px
<code>.col-sm-\$</code>	Small Devices	Tablets 768px and Up
<code>.col-md-\$</code>	Medium Devices	Desktops 992px and Up
<code>.col-lg-\$</code>	Large Devices	Large Desktops 1200px and Up

```
<div class="container-fluid">
<div class="row">
<nav class="col-lg-2"> ...</nav>
<div id="content1" class="col-lg-5"> ... </div>
<div id="content2" class="col-lg-5"> ... </div>
</div>
</div>
```


Bootstrap for Multicolumn Layout & Responsive Design

Even more degree of freedom!!!

```
<div class="container-fluid">
<div class="row">
<nav class="col-md-2 col-sm-12"> ...</nav>
<div id="content1" class="col-md-5 col-sm-6"> ... </div>
<div id="content2" class="col-md-5 col-sm-6"> ... </div>
</div>
</div>
```

Combining CSS Selectors for More Power

- ❖ CSS Grouping
- ❖ CSS Nesting
- ❖ Combining HTML and Class Selectors
- ❖ Combining HTML and ID Selectors



CSS Grouping

- ❖ Putting a comma between two selectors applies the same CSS rule to both selectors.

Instead of,

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

```
h2 { color: red; background-color: pink; }
```

It is better to say:

```
p, h2 { color: red; background-color: pink; }
```



CSS Nesting

- ❖ Using two selectors together without a comma or other punctuation in between
- ❖ It will apply the rule to the second selector if and only if it is nested inside the first selector

```
ul li { color: white; background-color: red; }
```

```
header h1 { font-weight: bold; }
```

Combining HTML and Class Selectors

- ❖ HTML selectors can be used with class selectors to provide extra specificity.

.column1

any element with a class of column1

h1.column1

a level-one heading with a class of column1

.column1 h1

a level-one heading nested inside an element with a class of column1

Combining HTML and Class Selectors

- ❖ HTML selectors can be used with class selectors to provide extra specificity.

.column1

any element with a class of column1

h1.column1

a level-one heading with a class of column1

.column1 h1

a level-one heading nested inside an element with a class of column1

<header>

<h1 class="column1"> Name of the website will sit here </h1>

</header>

<div class="column1">

<p> This is a paragraph. </p>

<h1> This is a level-one heading. </h1>

</div>

Combining HTML and ID Selectors

- ❖ HTML selectors can be used with ID selectors to increase specificity.


#column2 any element with an ID of column2
p#column2 a paragraph with an ID of column2
#column2 p a paragraph nested inside an element with an ID of column2

```


<header>
  <p ID="column2"> Tag line can be sit here </p>
</header>
<div ID="column2">
  <p> This is a paragraph. </p>
  <h1> This is a level-one heading. </h1>
</div>
  
```


Note: It is actually wrong to use the same value (here "column2") twice in the same html page.

Review from Last Lecture




To create columns on an HTML page, use which property?

- A. Column
- B. Placement
-  C. Float
- D. Position
- E. Alignment



To stop text from floating, use which property?

- A. Flow
- B. Stop
- C. No-float
-  D. Clear