HTML, CSS, Bootstrap Framework
What we will talk about:

Front-end vs Back-end coding at bswift

Defining HTML, CSS and Javascript

How the client stylesheets work

Examples

Tools
Static Pages  /  Dynamic Pages
A static website is a group of self-contained, individual pages (or page), sent to the browser from the server one-page-at-a-time.
Dynamic web content is built when it is requested, by the user directly, or programmatically while a user is on a page (e.g., facebook updates).

Most websites contain both static and dynamic elements.
Can I have a webpage, please?

back-end “recipe”

.NET

SQL databases

SERVER REQUEST

SERVER RESPONSE

front-end

HTML
CSS
Javascript

thanks!
Server-side / Client-side
aka
Back End / Front-end
Client-side (front-end) coding includes HTML, CSS and Javascript. This just means that our code will be downloaded from the server and then compiled entirely in the browser.
HTML, CSS, Javascript
Three layers of web design:

Structure, Style, Behavior
BEHAVIOR
Javascript

PRESENTATION
CSS
Imagery

STRUCTURE
HTML markup
Site planning
HTML
Hyper Text + Markup Language
Hyper Text
A markup language is a set of markup tags.

The purpose of the tags is to group and describe page content.
Markup Language

Without any markup to give your content structure, the browser renders unformatted and unstyled text, also known as “plain text”.

Pitchfork Music Festival Set Times Revealed
Check out the schedule for all three days

June 8, 2012 at 12:35 p.m.

Pitchfork Music Festival 2012, which returns to Chicago's Union Park July 13-15, is just five weeks away! The festival will feature Vampire Weekend, Feist, Godspeed You! Black Emperor, Beach House, Dirty Projectors, Hot Chip, Sleigh Bells, Wild Flag, Flying Lotus, Japandroids, Chromatics, Real Estate, A$AP Rocky, Danny Brown, and many, many more. And now, we're happy to announce the complete three-day schedule for the festival. Below, find the set times for all of the bands playing.

Friday, July 13:
Gates at 3 p.m.

8:30 Feist (GREEN)  
8:20 Purity Ring (BLUE)  
7:20 Dirty Projectors (RED)  
7:15 Clams Casino (BLUE)  
6:25 Big K.R.I.T. (GREEN)  
6:15 Japandroids (BLUE)  
5:30 A$AP Rocky (RED)

Saturday, July 14:
Gates at 12 p.m.

8:40 Grimes (BLUE)  
8:30 Godspeed You! Black Emperor (GREEN)  
7:40 Danny Brown (BLUE)  
7:25 Hot Chip (RED)  
6:45 Chromatics (BLUE)  
6:15 Sleigh Bells (GREEN)  
5:45 Schoolboy Q (BLUE)
Pitchfork Music Festival Set Times Revealed Check out the schedule for all three days June 8, 2012 at 12:35 p.m. Pitchfork Music Festival 2012, which returns to Chicago's Union Park July 13-15, is just five weeks away! The festival will feature Vampire Weekend, Feist, Godspeed You! Black Emperor, Beach House, Dirty Projectors, Hot Chip, Sleigh Bells, Wild Flag, Flying Lotus, Japandroids, Chromatics, Real Estate, A$AP Rocky, Danny Brown, and many, many more. And now, we're happy to announce the complete three-day schedule for the festival. Below, find the set times for all of the bands playing. Friday, July 13: Gates at 3 p.m. 8:30 Feist (GREEN) 8:20 Purity Ring (BLUE) 7:20 Dirty Projectors (RED) 7:15 Clams Casino (BLUE) 6:25 Big K.R.I.T. (GREEN) 6:15 Japandroids (BLUE) 5:30 A$AP Rocky (RED) Saturday, July 14: Gates at 12 p.m. 8:40 Grimes (BLUE) 8:30 Godspeed You! Black Emperor (GREEN) 7:40 Danny Brown (BLUE) 7:25 Hot Chip (RED) 6:45 Chromatics (BLUE) 6:15 Sleigh Bells (GREEN) 5:45 Schoolboy Q (BLUE) Buy tickets here.
**Markup Language**

HTML tags give structure and meaning to your content. “Semantic markup” refers to the use of meaningful tags to describe content (e.g. using header tags for header content).

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
</head>
<body>

<h1>Pitchfork Music Festival Set Times Revealed</h1>
<h2>Check out the schedule for all three days</h2>

<em>June 8, 2012 at 12:35 p.m.</em>

<p>Pitchfork Music Festival 2012, which returns to Chicago’s Union Park July 13-15, is just five weeks away! The festival will fe</p>

<h3>Friday, July 13:</h3>
<b>Gates at 3 p.m.</b>

<ul>
  <li>8:30 Feist (GREEN)</li>
  <li>8:20 Purity Ring (BLUE)</li>
  <li>7:20 Dirty Projectors (RED)</li>
  <li>7:15 Clams Casino (BLUE)</li>
  <li>6:15 Japandroids (BLUE)</li>
  <li>5:30 ASAP Rocky (RED)</li>
</ul>

<h3>Saturday, July 14:</h3>
<b>Gates at 12 p.m.</b>

<ul>
  <li>8:40 Grimes (BLUE)</li>
  <li>8:30 Godspeed You! Black Emperor (GREEN)</li>
</ul>
```
Once your content is marked up, the browser applies built-in default styles to the tags. While you can override these styles with css, your marked up, non-css styled document should be readable and have a clear hierarchy.

Pitchfork Music Festival Set Times Revealed

Check out the schedule for all three days

June 8, 2012 at 12:35 p.m.

Pitchfork Music Festival 2012, which returns to Chicago's Union Park July 13-15, is just five weeks away! The festival will feature Vampire Weekend, Feist, Godspeed You! Black Emperor, Beach House, Dirty Projectors, Hot Chip, Sleigh Bells, Wild Flag, Flying Lotus, Japandroids, Chromatics, Real Estate, ASAP Rocky, Danny Brown, and many, many more. And now, we're happy to announce the complete three-day schedule for the festival. Below, find the set times for all of the bands playing.

Friday, July 13:

Gates at 3 p.m.

- 8:30 Feist (GREEN)
- 8:20 Purity Ring (BLUE)
- 7:20 Dirty Projectors (RED)
- 7:15 Clams Casino (BLUE)
- 6:25 Big K.R.I.T. (GREEN)
- 6:15 Japandroids (BLUE)
- 5:30 ASAP Rocky (RED)

Saturday, July 14:

Gates at 12 p.m.
doctype
html
head
body
<!DOCTYPE html>

The doctype is not actually a tag, but a declaration, telling the browser what kind of html you are using. The doctype above declares HTML 5.
The `<html>` element defines the whole HTML document.
The `<head>` element contains special elements that instruct the browser where to find stylesheets, provide meta info, and more.
The `<body>` element contains the document content (what is shown inside the browser window).
The use of our first three tags (html, head and body), introduces an important concept: **Nesting**, which is when tags “wrap” other tags. When you create markup, you should indicate nesting by indenting the nested tags with 2 spaces (preferred) or a tab.

```html
<html>
  <head> </head>
  <body>
    <h1></h1>
    <p></p>
  </body>
</html>
```
Document Hierarchy: Parents, children and siblings

Just as in a genealogy tree, the family hierarchy is described in terms of relationships. All elements in the document have a parent (up to ‘document’, which is at the top), and may have children (nested inside) or siblings (placed alongside).
The ‘address’ of an element

The document hierarchy provides us with an ‘address’ for each element.

```html
<div class="client-text-container">
  <h1>Welcome to the Exchange!</h1>
  <h2>Applying the exchange is fast and easy</h2>
  <div class="exchange-application-callout">
    <table>
      <tr>
        <td class="exchange-application-welcome-body">
          <p>Suspendisse vestibulum dignissim quam. Integer vel augue. Phasellus nulla purus, interdum ac, volutpat id, viverra nec, semper turpis. Donec.</p>
        </td>
      </tr>
      <tr>
        <td class="exchange-application-welcome-body">
          <h4 style="text-transform: uppercase; margin-bottom: 10px;">You will need:</h4>
          <ul>
            <li>A list of your employees</li>
            <li>A list of your employees</li>
            <li>A list of your employees</li>
          </ul>
        </td>
      </tr>
      </table>
    </div>
</div>
```

in the div with class “client-text-container”, make all of the h2 elements orange and 24px.
HTML Elements
Anatomy of an Element

<tag>Content</tag>

An HTML element includes both the HTML tag and everything between the tag (the content).
Anatomy of an Element

<tag>Content</tag>

Tags normally come in pairs. The first tag is the **start tag**, and the second tag is the **end tag**.
Anatomy of an Element

<h1>Main Headline</h1>

HTML has a defined set of tag names (also called keywords) that the browser understands.
## The essential element tags

<table>
<thead>
<tr>
<th>Primary Structure</th>
<th>Structural Elements (block)</th>
<th>Formatting Elements (inline)</th>
</tr>
</thead>
<tbody>
<tr>
<td>html</td>
<td>p</td>
<td>em</td>
</tr>
<tr>
<td>head</td>
<td>br</td>
<td>i</td>
</tr>
<tr>
<td>body</td>
<td>h1 – h6</td>
<td>strong</td>
</tr>
<tr>
<td>title</td>
<td>ul</td>
<td>b</td>
</tr>
<tr>
<td>meta</td>
<td>ol</td>
<td>q</td>
</tr>
<tr>
<td>link</td>
<td>a</td>
<td>blockquote</td>
</tr>
<tr>
<td></td>
<td>img</td>
<td>(span)</td>
</tr>
</tbody>
</table>
<div>
  <p>Content</p>
</div>
<!DOCTYPE HTML>
<html>
  <head></head>
  <body></body>
</html>
<!DOCTYPE HTML>
<html>
  <head>
    <title>Learn HTML</title>
  </head>
  <body></body>
</html>
<!DOCTYPE HTML>
<html>
  <head>
    <title>Learn HTML</title>
  </head>
  <body>
    <h1>My first page</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. 
       ipsum dolor sit amet, consectetur adipiscing elit.</p>
  </body>
</html>
My first page

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ipsum dolor sit amet, consectetur adipiscing elit.
<h1>Header 1</h1>
<h2>Header 2</h2>
<h3>Header 3</h3>
<h4>Header 4</h4>
<h5>Header 5</h5>
<h6>Header 6</h6>
Header 1

Header 2

Header 3

Header 4

Header 5

Header 6
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ipsum dolor sit.
consectetur adipiscing elit.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ipsum dolor sit.
consectetur adipiscing elit.
<ul>
  <li>List item</li>
  <li>List item</li>
  <li>List item</li>
</ul>

<ol>
  <li>List item</li>
  <li>List item</li>
  <li>List item</li>
</ol>
List item
List item
List item

1. List item
2. List item
3. List item
<div>
  <h1>Division Header</h1>
  <p>Division Paragraph</p>
</div>

<div>
  <h1>Division 2 Header</h1>
  <p>Division 2 Paragraph</p>
</div>
Division Header

Division Paragraph

Division 2 Header

Division 2 Paragraph
Block elements

Block elements expand naturally

And naturally drop below other elements
<p>This paragraph contains <strong>inline element</strong></p>
This paragraph contains **inline element**
More Inline elements

<strong>Inline element</strong>

<em>Inline element</em>

<span>Inline element</span>
Inline element Inline element Inline element
Anatomy of an Element

Most elements can have attributes, which provides additional information about the element.
Anatomy of an Element

Attributes always follow the same format: `name="value"`. You can use either single or double quotes.
The essential attributes

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

img <img src="images/image.jpg" alt="Sam">

a <a href="http://colum.edu">My school</a>
```
CSS
Cascading + Style Sheet
A stylesheet is a set of rules defining how an html element will be “presented” in the browser.

These rules are targeted to specific elements in the html document.
The “cascade” part of CSS is a set of rules for resolving conflicts with multiple CSS rules applied to the same elements.

For example, if there are two rules defining the color or your h1 elements, the rule that comes last in the cascade order will “trump” the other.
Most elements will inherit many style properties from their parent elements by default.

HTML
<body>
  <div>
    <ul>
      <li></li>
    </ul>
  </div>
</body>

relationship

- `<body>`: parent of site
- `<div>`: parent of `ul` and `li`, child of `body`
- `<ul>`: parent of `li`, child of `div` and `body`
- `<li>`: child of `ul`, `div`, and `body`
Inheritance

body
make the paragraph 16px, Verdana, red

p
make the paragraph blue

16px, Verdana, blue
Shortly after styling your first html elements, you will find yourself wanting more control over where your styles are applied.

This is where *specificity* comes in.

Specificity refers to how specific your selector is in naming an element.
Specificity

body
make the paragraph 16px, Verdana, red

\[ \text{p} \]
make the paragraph blue

\[ \text{p.pink} \]
make the paragraph pink

16px, Verdana, pink
HTML
<div id="plan-2323">
  <p>Here is some text.</p>
  <p>Hide this text.</p>
</div>

<div id="plan-2323">
  <p>Here is some text.</p>
  <p class="hideclass">Hide this text.</p>
</div>

css
#plan-2323.hideclass {display: none}
<h1>Health Savings Account</h1>
CSS Syntax

Syntax = the rules for how to write the language
Three terms for describing your styles:

CSS rule
CSS selector
CSS declaration
Every style is defined by a selector and a declaration. The declaration contains at least one property/value pair. Together they are called a CSS Rule.
CSS Selector

body {font-family: Arial, Helvetica}

p {color: #666666}

h1 {font-size: 24px}

a {color: blue}

The **selector** associates css rules with HTML elements.
The selector is typed in front of the declaration, with a space separating it and the opening curly-bracket (aka curly-brace).

Typically, extra spaces and returns are added as shown for the sake of readability.
You can apply styles to multiple selectors in the same rule by separating the selectors with commas.
The declaration is always defined in a property/value pair. The two are separated by a colon. How you define the properties will affect how HTML elements are displayed.
You can apply multiple **declarations** to a **selector(s)** by separating the declarations with semi-colons.
CSS Selectors
p  Type (element)
#
ID
.
Class
The simplest selector is the type selector, which targets an HTML element by name.
<h1>My first page</h1>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>

h1{color:red;}
p{font-size:20px;}

eSpace
My first page

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ipsum dolor consectetur adipiscing elit.
<table>
<thead>
<tr>
<th>Primary Structure</th>
<th>Body Elements</th>
<th>Formatting Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>html</td>
<td>p</td>
<td>em</td>
</tr>
<tr>
<td>body</td>
<td>br</td>
<td>i</td>
</tr>
<tr>
<td></td>
<td>h1 – h6</td>
<td>strong</td>
</tr>
<tr>
<td></td>
<td>ul</td>
<td>b</td>
</tr>
<tr>
<td></td>
<td>ol</td>
<td>q</td>
</tr>
<tr>
<td></td>
<td>a</td>
<td>blockquote</td>
</tr>
<tr>
<td></td>
<td>img</td>
<td>span</td>
</tr>
<tr>
<td></td>
<td>div</td>
<td></td>
</tr>
</tbody>
</table>
ID Selectors

CSS

#logo {declaration}

HTML

<img id="logo" src="" alt=""/>

An ID is an html attribute that is added to your html markup. You reference that ID in your css with a hash.
<h1>My first page</h1>

<p id="intro">Lorem ipsum dolor sit amet, adipiscing elit. adipiscing elit.</p>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. adipiscing elit.</p>

h1{color:red;}
p{font-size:20px;}
#intro{color:green; }
My first page

Lorem ipsum dolor sit amet, consectetur adipiscing elit. adipiscing elit.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. adipiscing elit.
A class is an html attribute that is added to your html markup. You reference that ID in your css with a period.
<h1>My first page</h1>

<p class="intro">Lorem ipsum dolor sit amet, adipiscing elit. adipiscing elit.</p>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. adipiscing elit.</p>

h1{color:red;}
p{font-size:20px;}
.intro{color:green;}

eSpace
My first page

Lorem ipsum dolor sit amet, consectetur adipiscing elit. adipiscing elit.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. adipiscing elit.
IDs vs Classes

The most important difference between IDs and classes is that there can be only one ID on a page, but multiple classes.

An ID is more specific than a class.

An element can have both an ID and multiple classes.
IDs vs Classes

ID: #344-34-4344
Class: Male
Class: Employee

ID: #123-54-9877
Class: Female
Class: Employee
My first page

Lorem ipsum dolor adipiscing elit. adipiscing elit.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. adipiscing elit.
My first page

Lorem ipsum dolor sit amet, consectetur adipiscing elit. adipiscing elit.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. adipiscing elit.
Descendant Selectors

CSS
#sidebar .author {declaration}

HTML
<div id="sidebar">
  <p class="author"></p>
</div>

A space between two selectors indicates a descendant selector. In the example above, the style is targeted to an element with the class "author" inside the id "sidebar".
<div class="content">
  <h1>My first page</h1>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. adipiscing elit.</p>
</div>

.content { background: #eee; }
.content h1 { color: red; }
.content p { color: green; }
My first page

Lorem ipsum dolor sit amet, consectetur adipiscing elit. adipiscing elit.
Elements can have multiple classes, giving you more control. The are written in the CSS in the exact order they appear in the html, with no spaces.
Box Model (Block)
Box Model (Block)

```html
<div class="box1">box1</div>
<div class="box2">box2</div>
```
Box Model (Block)

```
.box1{
  background: red;
  padding: 20px;
  margin: 20px;
  border: 5px solid black;
}
.box2{
  background: green;
  padding: 20px;
  margin: 20px;
  border: 5px solid black;
}
```
Margin collapse

Before

Content Area
margin-bottom: 30px
margin-top: 20px

Content Area

After

Content Area
margin-bottom: 30px

Content Area

Margins collapse to form a single margin.
Margin collapse

Without Margin Collapsing

With Margin Collapsing

The space between paragraphs is double the space at the top.
The space between paragraphs is the same as the space at the top.
Box Model (Inline)
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Eget felis mollis, venenatis nunc non ornare nibh. Venenatis nunc non, ornare nibh orcem ipsum dolor sit amet
Relative Position

- Box 1: left: 20px
- Box 2: position: relative
- Box 3: top: 20px

Containing element
Absolute Position

box 1

left: 20px

top: 20px

position: absolute

box 2

box 3

relatively positioned ancestor
Fixed Position
Float

No Boxes Floated

box 1
box 2
box 3

Box 1 Floated Right

box 1

box 2
box 3
Float

Container does not enclose floats.

Container now encloses floats.

Floats take up no space.

Empty clearing div.
Bootstrap is Front-end Framework

HTML, CSS, and JS framework for developing responsive, mobile first projects on the web.

www.getbootstrap.com
Bootstrap is **Ready-to-use Web Elements**

HTML / CSS elements for button, form, table, image, navbar, label, progress bar, alert etc.
**EXAMPLES of Bootstrap Elements**

<table>
<thead>
<tr>
<th>#</th>
<th>Column heading</th>
<th>Column heading</th>
<th>Column heading</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Column content</td>
<td>Column content</td>
<td>Column content</td>
</tr>
<tr>
<td>2</td>
<td>Column content</td>
<td>Column content</td>
<td>Column content</td>
</tr>
<tr>
<td>3</td>
<td>Column content</td>
<td>Column content</td>
<td>Column content</td>
</tr>
<tr>
<td>4</td>
<td>Column content</td>
<td>Column content</td>
<td>Column content</td>
</tr>
<tr>
<td>5</td>
<td>Column content</td>
<td>Column content</td>
<td>Column content</td>
</tr>
<tr>
<td>6</td>
<td>Column content</td>
<td>Column content</td>
<td>Column content</td>
</tr>
<tr>
<td>7</td>
<td>Column content</td>
<td>Column content</td>
<td>Column content</td>
</tr>
<tr>
<td>8</td>
<td>Column content</td>
<td>Column content</td>
<td>Column content</td>
</tr>
<tr>
<td>9</td>
<td>Column content</td>
<td>Column content</td>
<td>Column content</td>
</tr>
</tbody>
</table>

![Bootstrap Element Example](image)

**Oh snap! You got an error!**

Change this and that and try again. Duis mollis, est non commodo luctus, nisi erat porttitor ligula, eget lacinia odio sem nec elit. Cras mattis consectetur purus sit amet fermentum.

- Take this action
- Or do this
more EXAMPLES of Bootstrap Elements
Websites created by Bootstrap

http://unroll.me
Websites created by Bootstrap

www.fliplingo.com


Fliplingo is the simplest way to translate your tweets, with real human translators.

LEARN MORE

GET STARTED — IT'S FREE!
Why Bootstrap?

- Save 100+ hours of coding
- Easy to use web elements
- Quick responsive prototype / website
- Great documentation
Bootstrap Package

- **CSS** - bootstrap.css
- **JS** - bootstrap.js
- **Icon Fonts** - glyphicons-halflings-regular.ttf
Bootstrap CDN

www.bootstrapcdn.com

Quickstart

Complete CSS

//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css

Complete JavaScript

//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.min.js

Mad Love

Lewis Lehe
@LewisLehe

this rules! #BootstrapCDN:
@twBootstrap hosted by @MaxCDN bootstrapcdn.com
#webperf via @jdorfman
11:48 PM - 8 Nov 2013
2 RETWEETS

Tim Watson
@visualinventor

Twitter Bootstrap gets a speed
bump via CDN serving. bootstrapcdn.com
Bootstrap Playground

Bootstrap Components

Start Workshop
What is Grid in web design?
What is Grid in web design?
What is Grid in web design?

4 Grids x 3 Columns = 12 Grids
## Bootstrap Grid

<table>
<thead>
<tr>
<th>E-Commerce</th>
<th>Responsive Design</th>
<th>Web Security</th>
</tr>
</thead>
</table>

### 12 Responsive Grid
Grid Overlay for Bootstrap & Foundation


Grid displayer for Bootstrap and Foundation

In-browser web design made easier with this bookmarklet which displays the grid of your favourite front-end framework.

1 Install it
For Firefox and WebKit browsers only.

2 Try it
Try the bookmarklet on this page or on the following sites:
## 4 Sizes of Bootstrap Grid

<table>
<thead>
<tr>
<th>Size Name</th>
<th>Screen Size</th>
<th>CSS Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extra Small Devices (Phone)</td>
<td>0 - 767 px</td>
<td>.col-xs-1 ~ .col-xs-12</td>
</tr>
<tr>
<td>Small Devices (Tablet)</td>
<td>768 - 991 px</td>
<td>.col-sm-1 ~ .col-sm-12</td>
</tr>
<tr>
<td>Medium Devices (Desktop)</td>
<td>992 - 1219 px</td>
<td>.col-md-1 ~ .col-md-12</td>
</tr>
<tr>
<td>Large Devices (Large screen desktop)</td>
<td>1200px +</td>
<td>.col-lg-1 ~ .col-lg-12</td>
</tr>
</tbody>
</table>
# 4 Sizes of Bootstrap Grid

<table>
<thead>
<tr>
<th>Size Name</th>
<th>Screen Size</th>
<th>CSS Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extra Small Devices (Phone)</td>
<td>0 - 767 px</td>
<td>.col-xs-1 ~ .col-xs-12</td>
</tr>
<tr>
<td>Small Devices (Tablet)</td>
<td>768 - 991 px</td>
<td>.col-sm-1 ~ .col-sm-12</td>
</tr>
<tr>
<td>Medium Devices (Desktop)</td>
<td>992 - 1219 px</td>
<td>.col-md-1 ~ .col-md-12</td>
</tr>
<tr>
<td>Large Devices (Large screen desktop)</td>
<td>1200px +</td>
<td>.col-lg-1 ~ .col-lg-12</td>
</tr>
</tbody>
</table>
Bootstrap Grid Example


How many grids in each box?
Bootstrap Grid Example

4 grids x 3 Columns
Bootstrap Grid Example

```html
<div class="container">
  <div class="row">
    <div class="col-md-4">
      ...
    </div>
  </div>
  <div class="row">
    <div class="col-md-4">
      ...
    </div>
    <div class="col-md-4">
      ...
    </div>
  </div>
</div>
```
Bootstrap Grid Example

```html
<div class="container">
  <div class="row">
    <div class="col-md-4">
      ... 
    </div>
    <div class="col-md-4">
      ... 
    </div>
    <div class="col-md-4">
      ... 
    
  </div> <!-- row -->
</div> <!-- container -->
```
Bootstrap Grid Example 2

How many grids in each box?
Bootstrap Grid Example 2

6 grids x 2 Columns
Bootstrap Grid Example 2

```html
<div class="container">
  <div class="row">
    <div class="col-md-6">
      ...
    </div>
    <div class="col-md-6">
      ...
    </div>
  </div>
</div>
```
Bootstrap Grid Example 2

```html
<div class="container">
  <div class="row">
    <div class="col-md-6">
      ...
    </div>
    <div class="col-md-6">
      ...
    </div>
  </div>
</div>
```

![HTML Example](image-url)
Bootstrap Row

1 Row = 12 Grids
Bootstrap Row Example

HTML

```html
<div class="container">
    <div class="row">
        <div class="col-md-12">
            ...
        </div>
    </div>
</div>

<div class="row">
    <div class="col-md-4">
        ...
    </div>
    <div class="col-md-4">
        ...
    </div>
    <div class="col-md-4">
        ...
    </div>
</div>

<div class="row">
    <div class="col-md-4">
        ...
    </div>
    <div class="col-md-4">
        ...
    </div>
    <div class="col-md-4">
        ...
    </div>
</div>

</div>
```
Bootstrap Row Example

```html
<div class="container">
  <div class="row">
    <div class="col-md-12">
      ... </div>
  </div><!-- row -->
  
  <div class="row">
    <div class="col-md-4">
      ... </div>
    <div class="col-md-4">
      ... </div>
    <div class="col-md-4">
      ... </div>
  </div><!-- row -->
  
  <div class="row">
    <div class="col-md-4">
      ... </div>
    <div class="col-md-4">
      ... </div>
    <div class="col-md-4">
      ... </div>
  </div><!-- row -->
</div><!-- container -->
```
Bootstrap Responsive Grid

Columns will stack when responsive
Bootstrap Responsive Grid

Desktop

Mobile

Columns stack on mobile
Bootstrap Grid Workshop

3 Easy Steps:
1. Add container
2. Add row
3. Add columns
Bootstrap Grid Workshop