HTML, CSS, Bootstrap Framework

What we will talk about:

Front-end vs Back-end coding at bswift

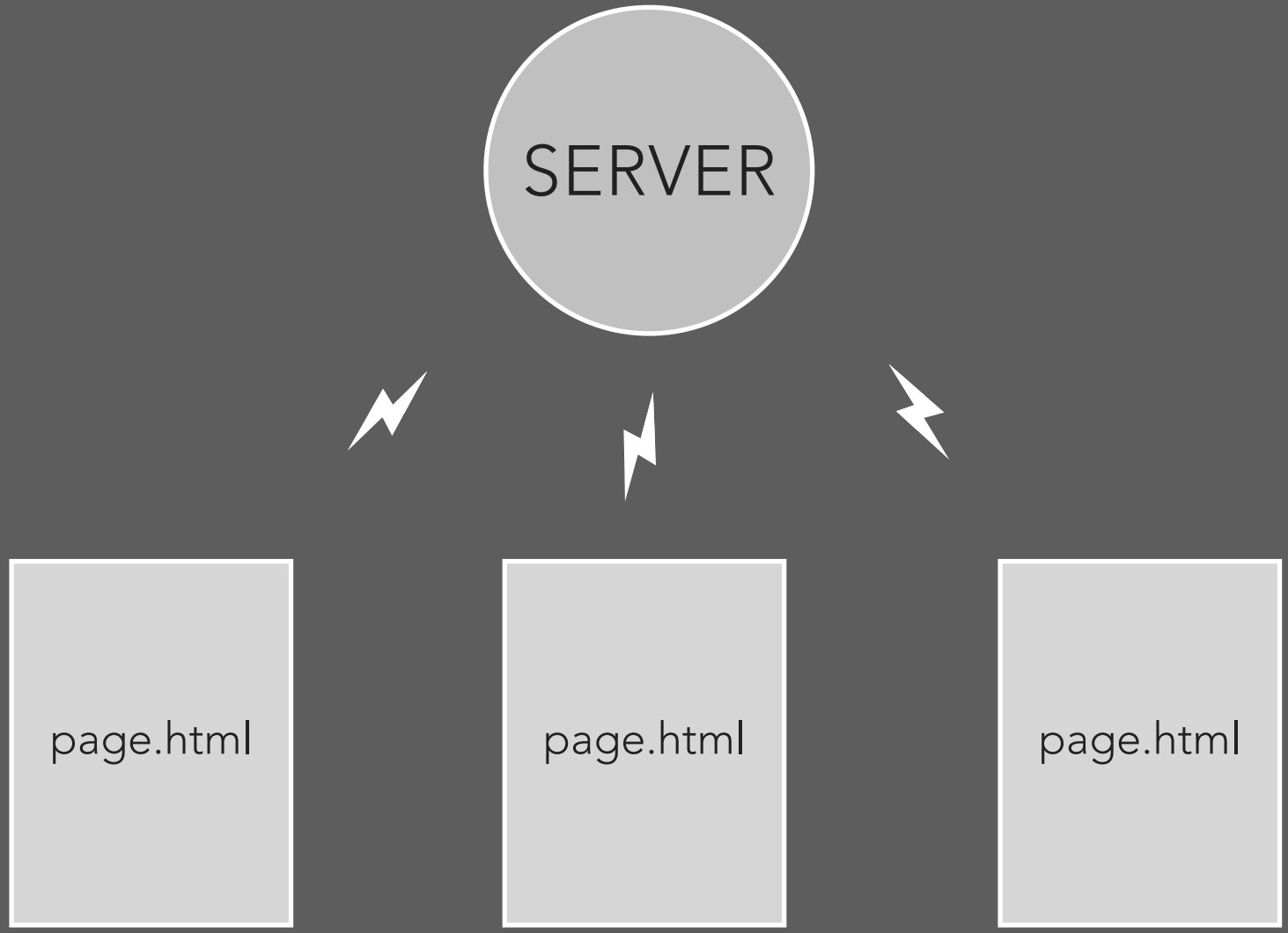
Defining HTML, CSS and Javascript

How the client stylesheets work

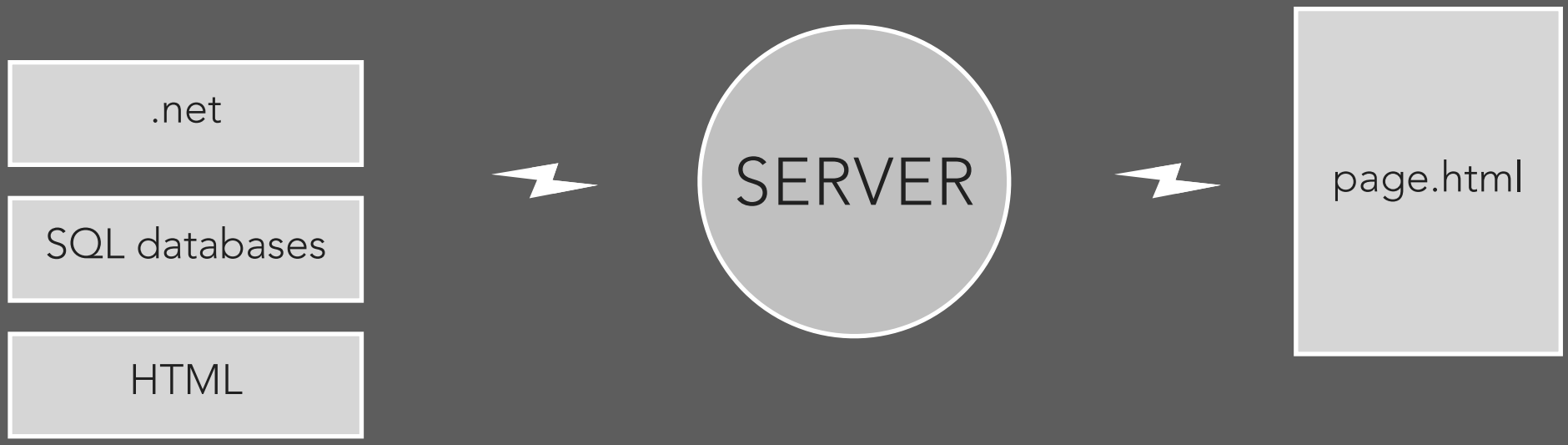
Examples

Tools

Static Pages / Dynamic Pages

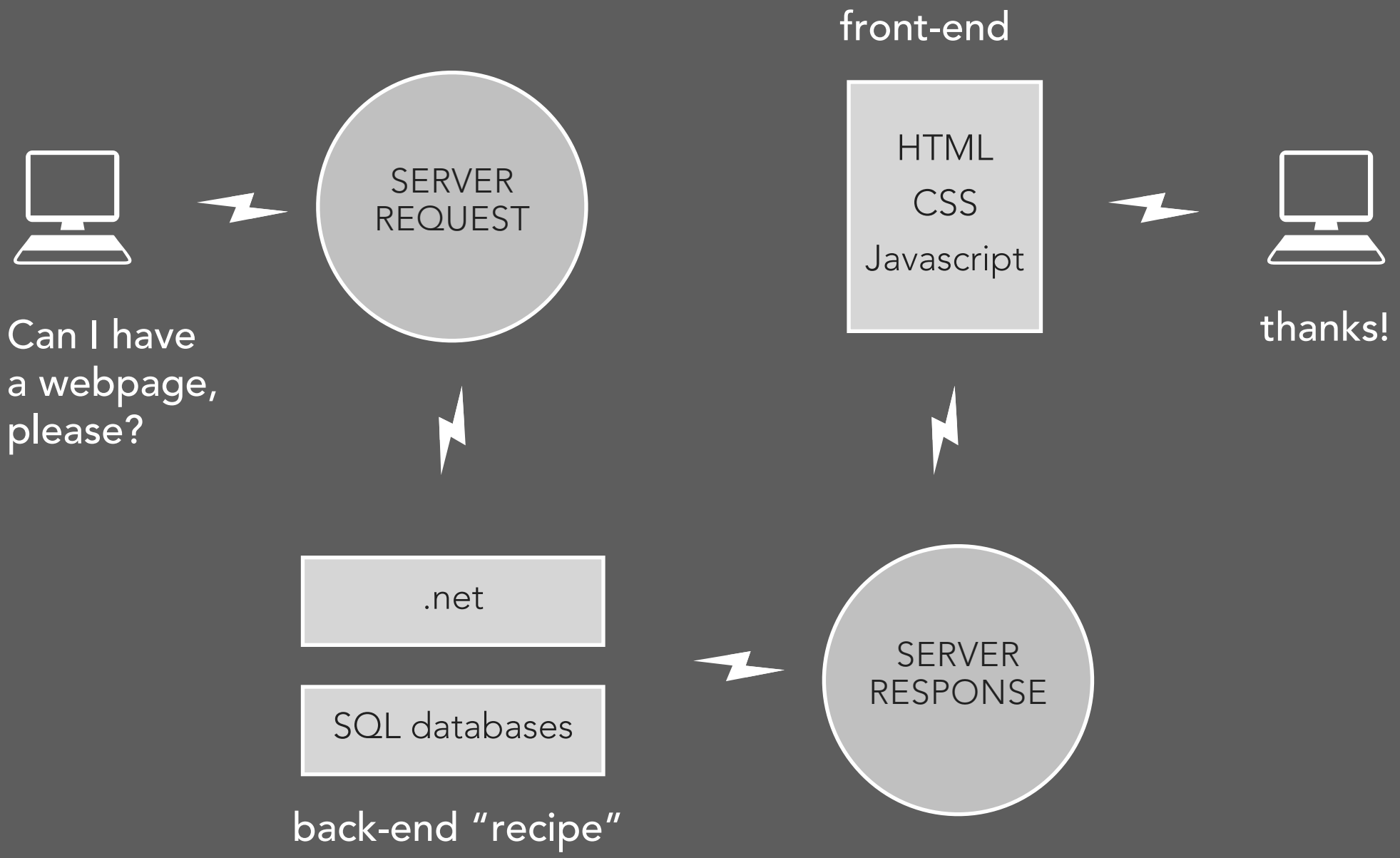


A static website is a group of self-contained, individual pages (or page), sent to the browser from the server one-page-at-a-time.



Dyamic 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).

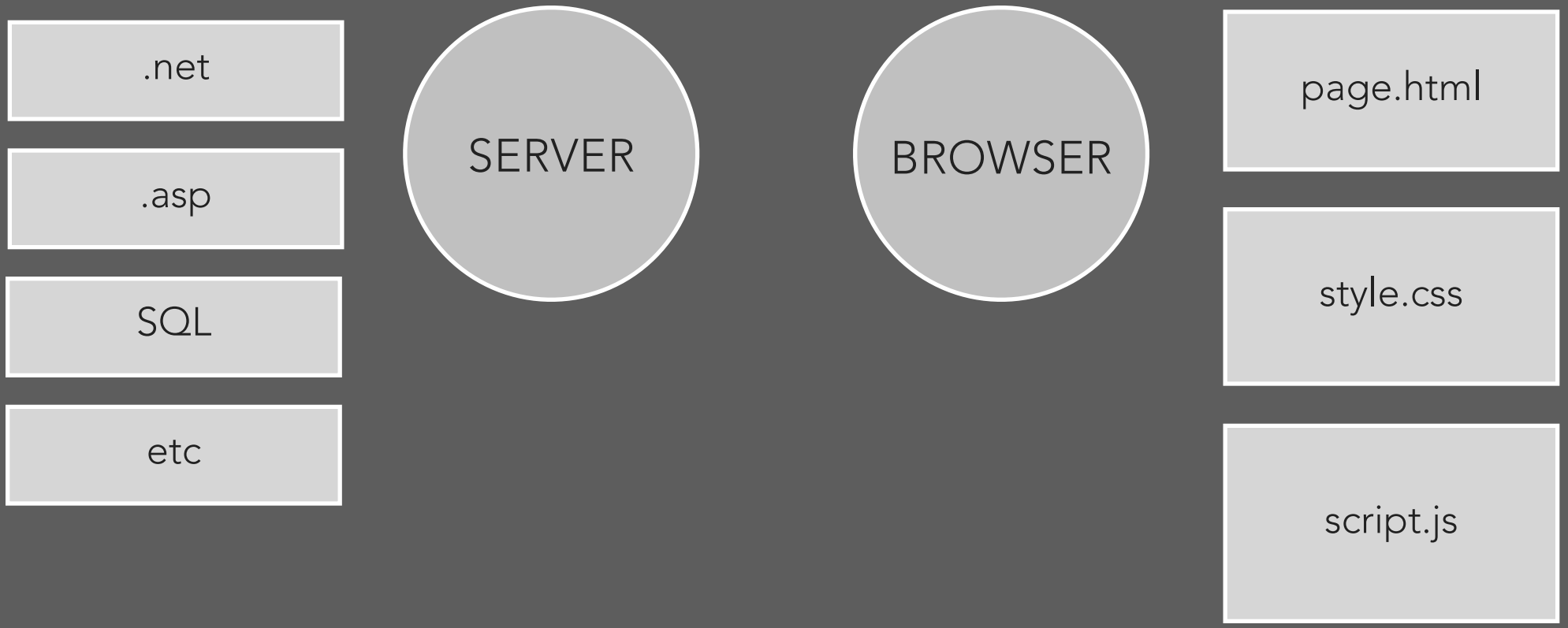
Most websites contain both static and dynamic elements.


front-end
SERVER
REQUEST
HTML
CSS
Javascript
Can I have
a webpage,
please?
thanks!
.net
SQL databases
SERVER
RESPONSE
back-end “recipe”

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.

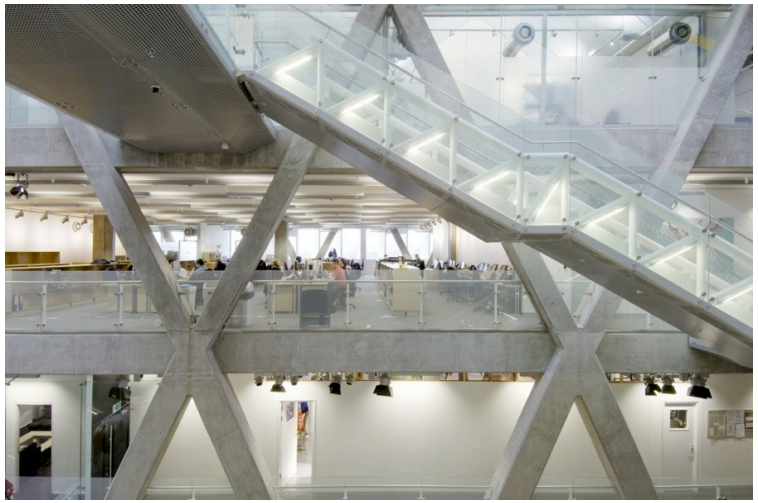
HTML, CSS, Javascript

Three layers of web design:

Structure, Style, Behavior

BEHAVIOR

Javascript



PRESENTATION

CSS

Imagery

STRUCTURE

HTML markup

Site planning

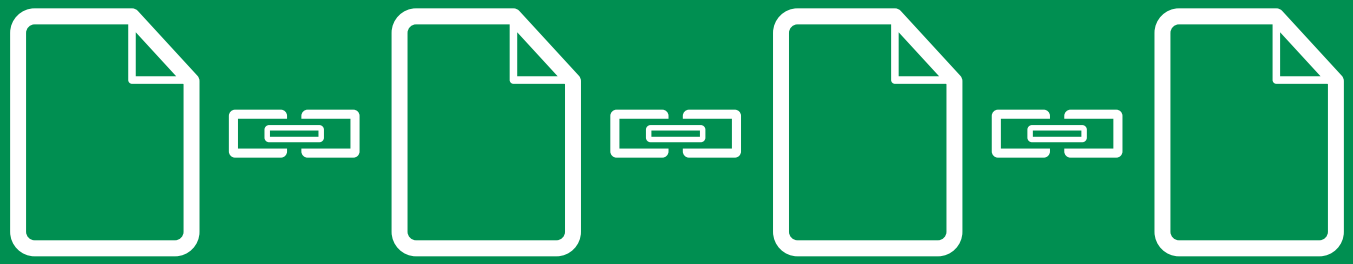
HTML

Hyper Text

+

Markup Language

Hyper Text



Markup Language

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).

```
<!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 f

  <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>6:25 Big K.R.I.T. (GREEN)</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>
```

Markup Language

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, 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 ASAP Rocky (RED)

Saturday, July 14:

Gates at 12 p.m.

doctype

html

head

body

EXCEPTION

<!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.

<html></html>

The <html> element defines the whole HTML document.

<head></head>

The <head> element contains special elements that instruct the browser where to find stylesheets, provide meta info, and more.

<body></body>

The <body> element contains the document content (what is shown inside the browser window).

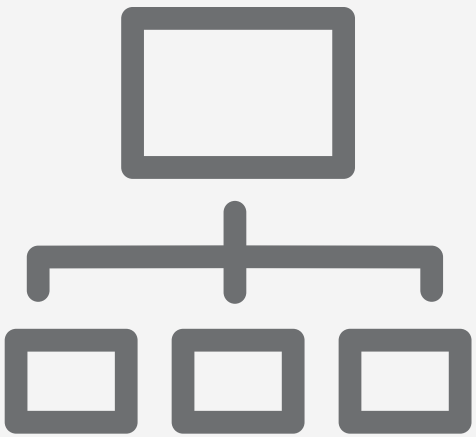
Nesting

The use of our first three tags (html, head and body), introduces and 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>
  <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).



<parent x>

<child and sibling y> </child and sibling y>

<child and sibling z> </child and sibling z>

</parent x>

The ‘address’ of an element

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

```
<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, v
        </td>
        <td>
          <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>
```

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

Primary Structure

html

head

body

Head Elements

title

meta

link

Structural Elements (block)

p

br

h1 – h6

ul

ol

a

img

(div)

Formatting Elements (inline)

em

i

strong

b

q

blockquote

(span)

Nested Tags

```
<div>
   <p>Content</p>
</div>
```

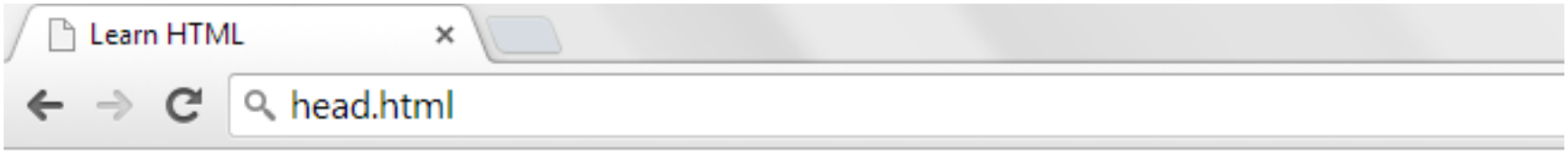
Basic Structure

```
<!DOCTYPE HTML>
<html>
   <head></head>
   <body></body>
</html>
```

Head

```
<!DOCTYPE HTML>
<html>
   <head>
      <title>Learn HTML</title>
   </head>
   <body></body>
</html>
```

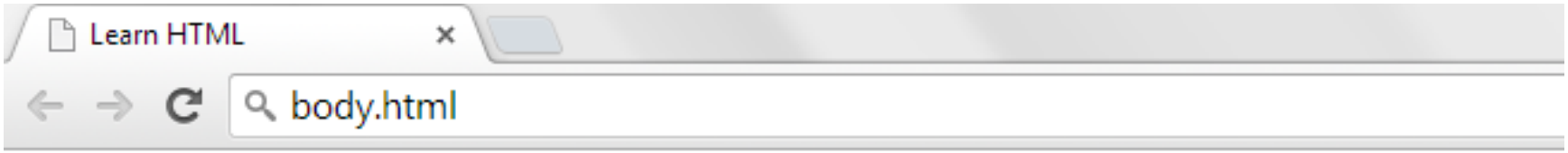
Preview



Body

```
<!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>
```

Preview



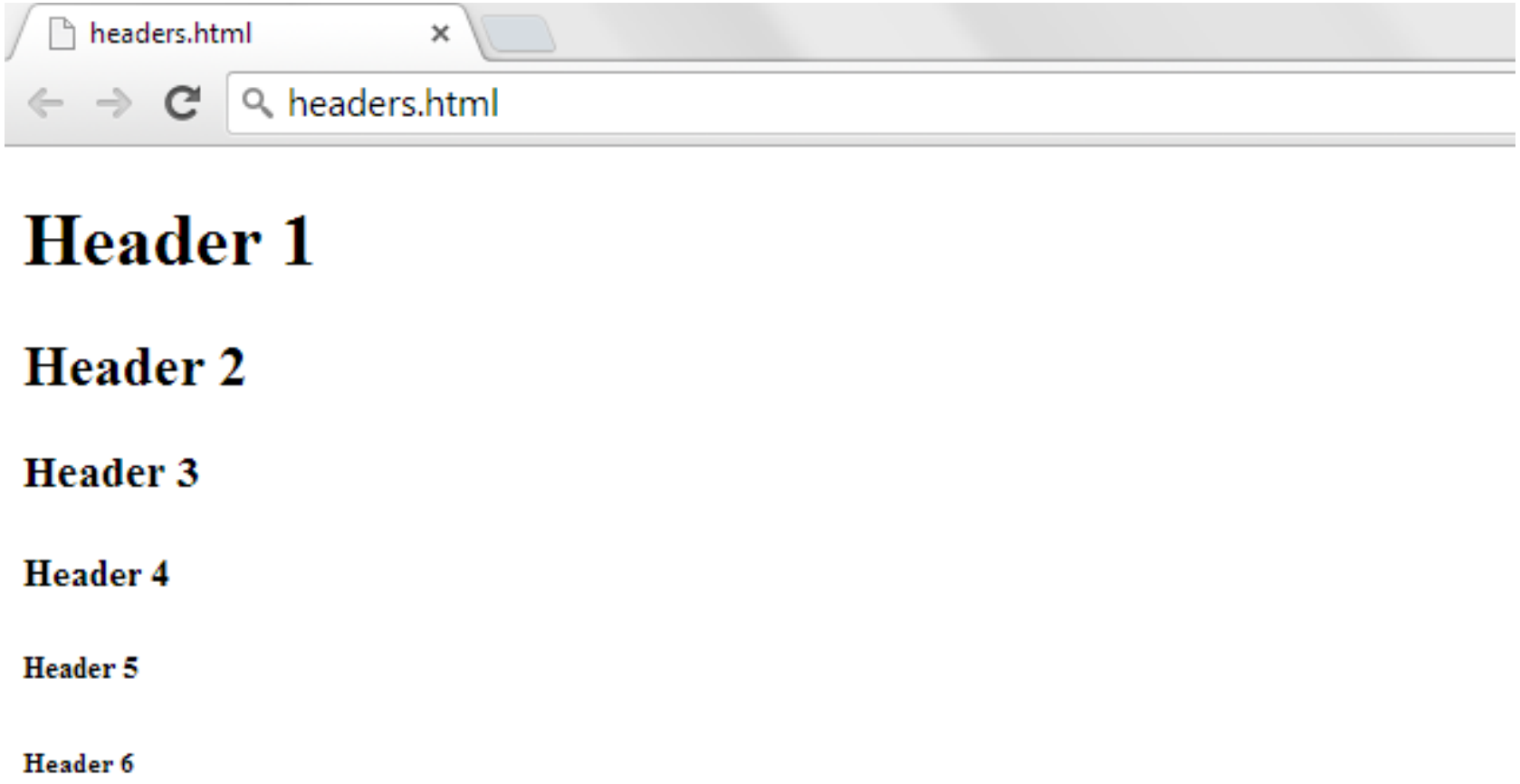
My first page

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

Headers

```
<h1>Header 1</h1>
<h2>Header 2</h2>
<h3>Header 3</h3>
<h4>Header 4</h4>
<h5>Header 5</h5>
<h6>Header 6</h6>
```

Preview

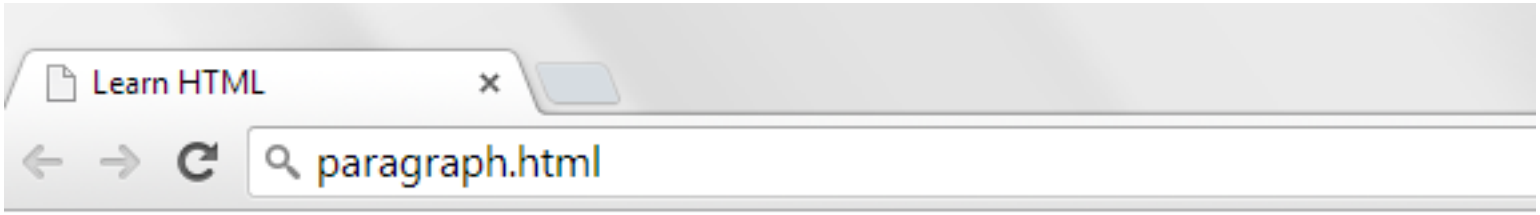

headers.html
headers.html
Header 1
Header 2
Header 3
Header 4
Header 5
Header 6

Paragraph

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

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

Preview



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

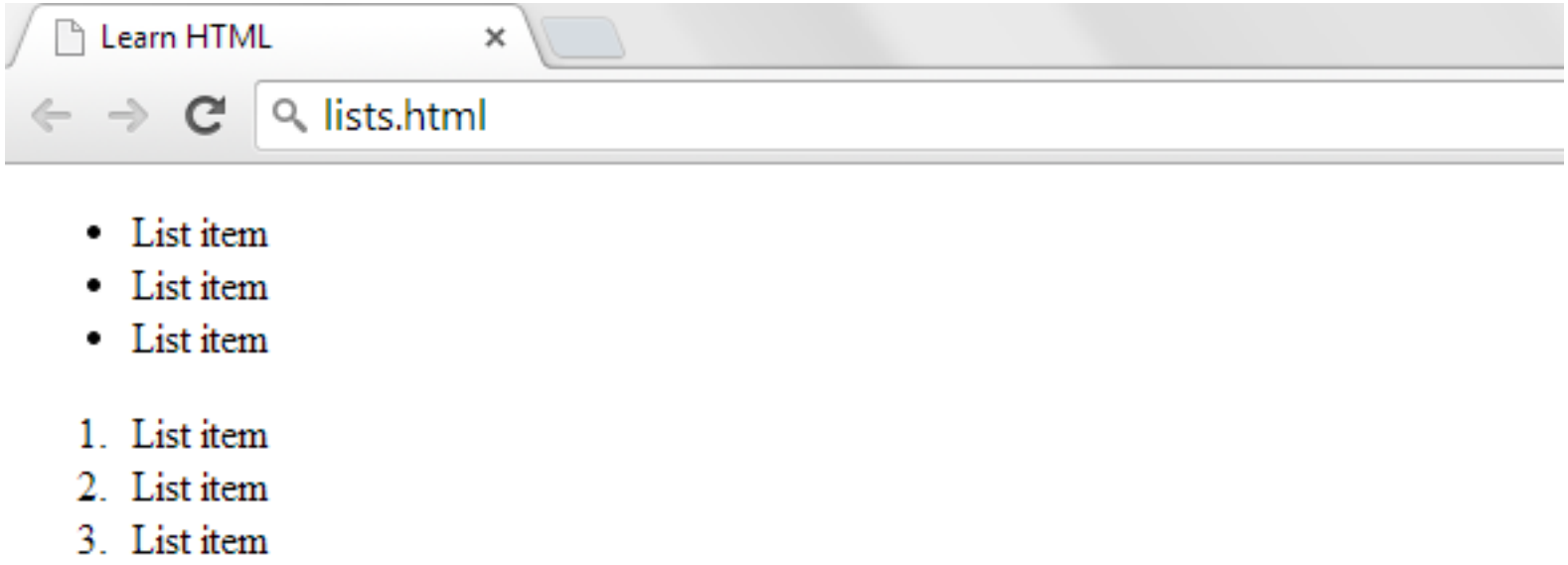
Lorem ipsum dolor sit amet, consectetur adipiscing elit. ipsum dolor sit consectetur adipiscing elit.

Lists

```
<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>
```

Preview

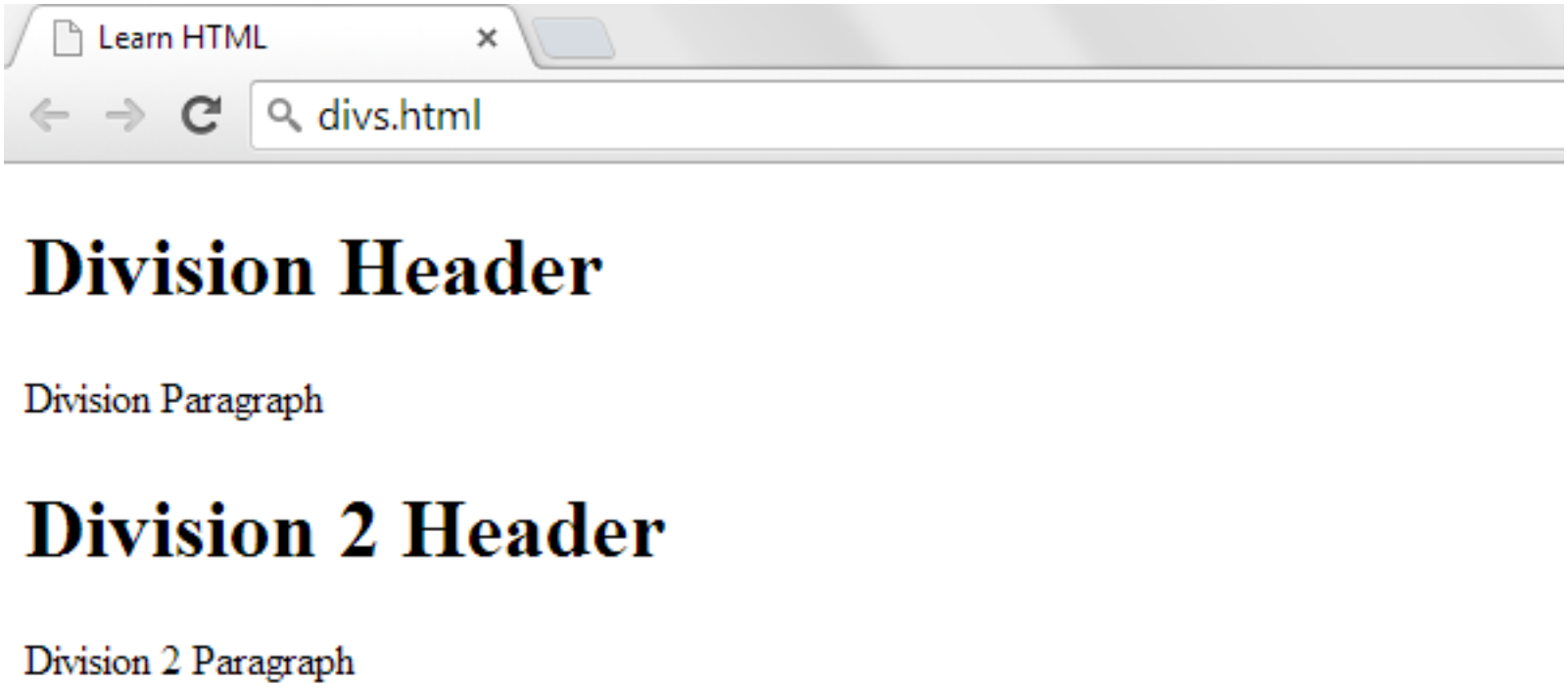

Learn HTML
lists.html
List item
List item
List item
1. List item
2. List item
3. List item

Div

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

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

Preview

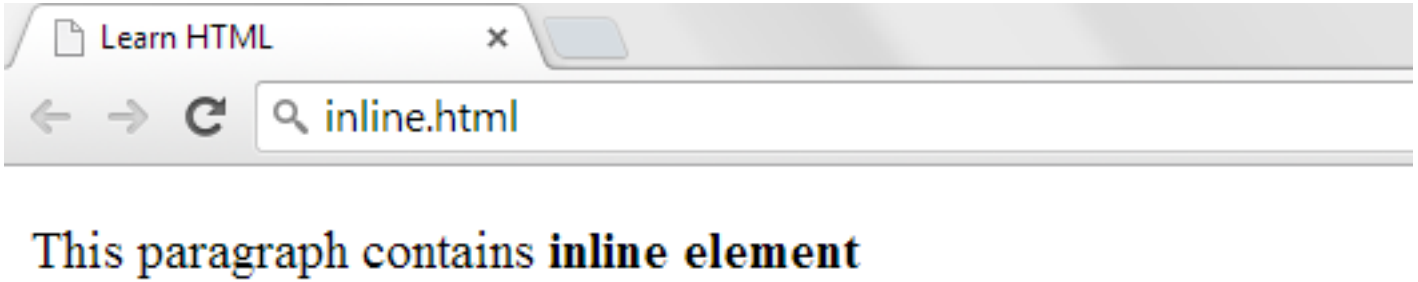

Learn HTML
divs.html
Division Header
Division Paragraph
Division 2 Header
Division 2 Paragraph

Block elements

Inline elements

```
<p>This paragraph contains <strong>inline elment</strong></p>
```

Preview


Learn HTML
inline.html
This paragraph contains inline element

More Inline elements

```
<strong>Inline element</strong>

<em>Inline element</em>

<span>Inline element</span>
```

Preview

Learn HTML
inline-2.html
Inline element Inline element Inline element

Anatomy of an Element

<html lang="en"></html>

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 My school

CSS

Cascading
+
Style Sheet

The Stylesheet

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

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.

low importance

Browser stylesheet

Linked (external) stylesheet

Embedded (internal) stylesheet

Inline (internal) Styles

high importance

```
  <link rel="stylesheet" href="../css/normalize.css" media="screen, print"
  <link rel="stylesheet" href="../css/global.css" media="screen, print" ty
  <link rel="stylesheet" href="../css/tables.css" media="screen, print" ty
  <link rel="stylesheet" href="../css/forms.css" media="screen, print" typ
  <link rel="stylesheet" href="../css/enrollment.css" media="screen, print
  <link rel="stylesheet" href="../css/portlets.css" media="screen, print"
  <link rel="stylesheet" href="../css/jquery-ui-themes/smoothness/jquery-u
  <link rel="stylesheet" href="../css/plugins.css" media="screen, print" t
  <link rel="stylesheet" href="../css/templates.css" media="screen, print"
  <link rel="stylesheet" href="../css/askemma.css" media="screen, print" t


  <link rel="stylesheet" href="client49384.css" media="screen, print" type



</head>
```

Inheritance

Most elements will inherit many style properties from their parent elements by default.

HTML	relationship
`<body>`	parent of site
`<div>`	parent of ul and li, child of body
`<ul>`	parent of li, child of div and body
`<li></li>`	child of ul, div, and body
`</ul>`	
`</div>`	
`</body>`	

Inheritance

body
make the paragraph 16px, Verdana, red

p
make the paragraph blue

16px, Verdana, blue

Specificity

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

p
make the paragraph blue

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}
```

```
<div id="benefittype-55" style="display:none;">
  <div class="plan-header"><h1>Health Savings Account</h1></div><div class="p
    <div id="plan-info-left">

    </div><div id="plan-info-right">
      <div class="plan-info-video">
        <div class="plan-info-video-thumb">
          <img src="/images/generic-video-still.jpg" border="0" alt="Video Th
        </div><div class="plan-info-video-title">
          <ul><li><a href="#this" onclick="try{playThisVideoNow('201108241817
        </div>
      </div><div class="plan-info-resources">

      </div>
    </div>
  </div><div class="plan-selection-container">
    <div class="plan-selection-step" id="planSelection-55">
      <div class="plan-selection-step-header" onclick="javascript:expandStep(
        <div class="plan-step-label"><ul><li class="plan-step-number"></li><l
      </div><div class="plan-selection-step-content" style="display:none;">
        <table class="plan-selection-subheader" id="planTypeInfo-55" border="
          <tr>
            <td class="plan-step-cart"><img src="/images/icon-cart.gif"/></td
          </tr>
        </table><table class="plan-selection-cost-summary" id="planTypeCostSu
          <tr>
            <th colspan="2">Cost Summary (per month)</th>
          </tr><tr>
            <td>Total Premium</td><td id="tp_55">$312.50</td>
```

CSS Syntax

Syntax = the rules for how to write the language

Three terms for describing your styles:

CSS rule

CSS selector

CSS declaration

CSS Rule

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

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.

CSS Selector

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

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.

CSS Selector

```
h1,h2,h3,h4 {
   font-weight: bold
}
```

You can apply styles to multiple **selectors** in the same rule by separating the selectors with commas.

CSS Declaration

```
p {
  property: value
}
```

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.

CSS Declaration

```
p {
  font-family: Arial, sans-serif;
  font-size: 14px;
  color: #666666;
}
```

You can apply multiple **declarations** to a **selector**(s) by separating the delcarations with semi-colons.

CSS Selectors

p	Type (element)
#	ID
.	Class

Type (element) Selectors

```
body {declaration}
p {declaration}
h1 {declaration}
ul {declaration}
```

The simplest selector is the type selector, which targets an html element by name.

Element Selector

```
<h1>My first page</h1>
<p>Lorem ipsum dolor sit amet,
  consectetur adipiscing elit.
  ipsum dolor sit amet, consectetur
  adipiscing elit.</p>
```

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

Preview

My first page

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

The essential selector types (elements)

Primary Structure

html

body

Body Elements

p

br

h1 – h6

ul

ol

a

img

div

Formatting Elements

em

i

strong

b

q

blockquote

span

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.

ID Selector

```
<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;}
```

Preview

My first page

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

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

Class Selectors

CSS

```
.ingredients {declaration}
```

HTML

```
<ul class="ingredients">
```

A class is an html attribute that is added to your html markup. You reference that ID in your css with a period.

Class Selector

```
<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;}
```

Preview

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

2 Rules

```
<h1>My first page</h1>

<p id="intro" class="intro">Lorem ipsum dolor
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;}
.intro{color:blue;}
```

Preview

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>
```

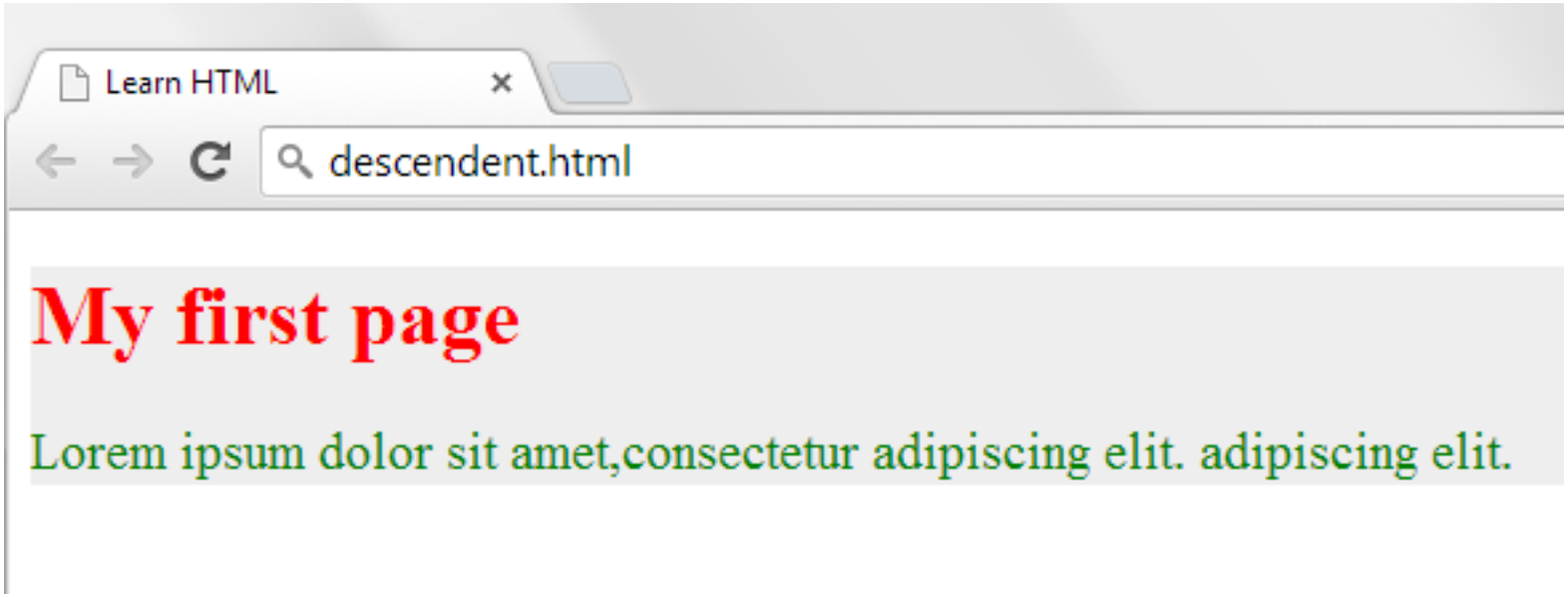
A space between two selectors indicates a descendant selector. In the example above, the style is targeted to an element with the class "author" inside the id "sidebar".

Descendent Selector

```
<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;}
```

Preview


Learn HTML
descendent.html
My first page
Lorem ipsum dolor sit amet,consectetur adipiscing elit. adipiscing elit.

Multiple classes

CSS

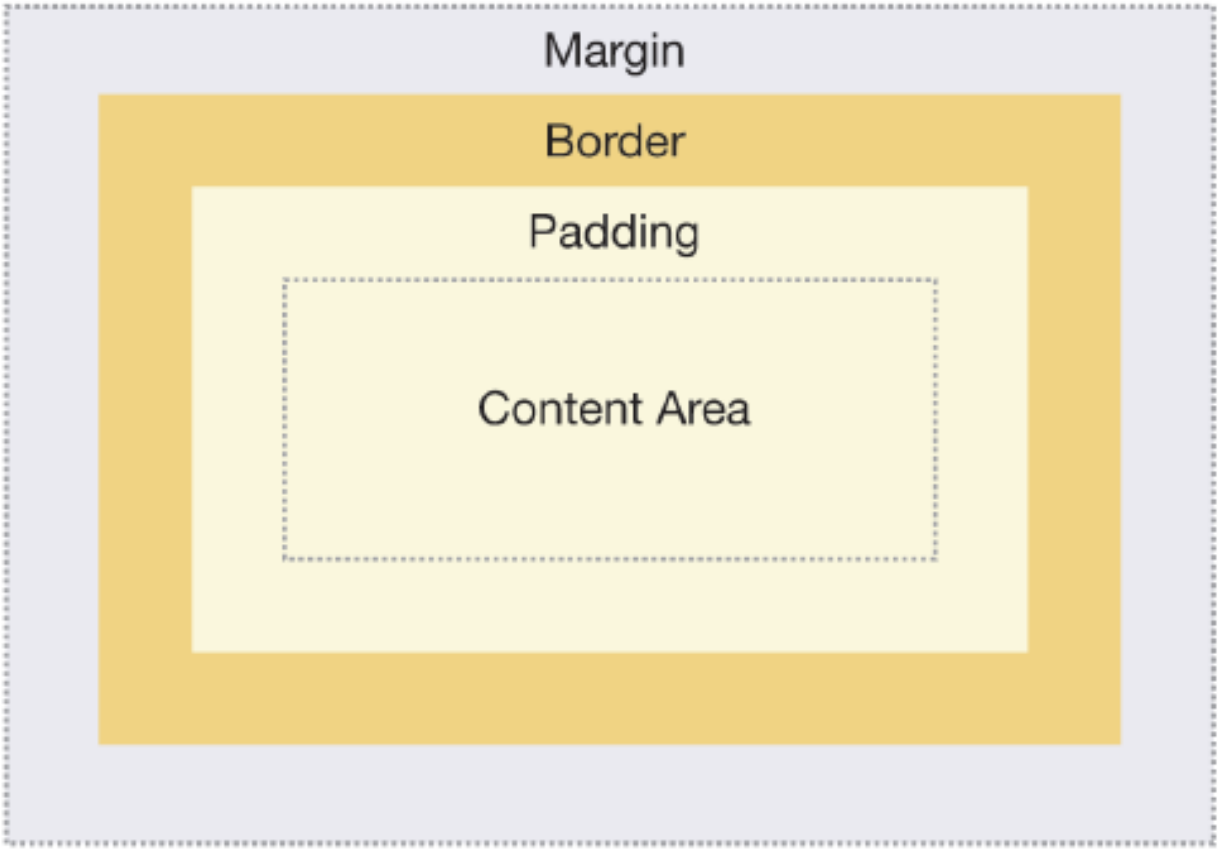
```
.ingredients.time {declaration}
```

HTML

```
<div class="ingredients time">
  <h1></h1>
</div>
```

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)

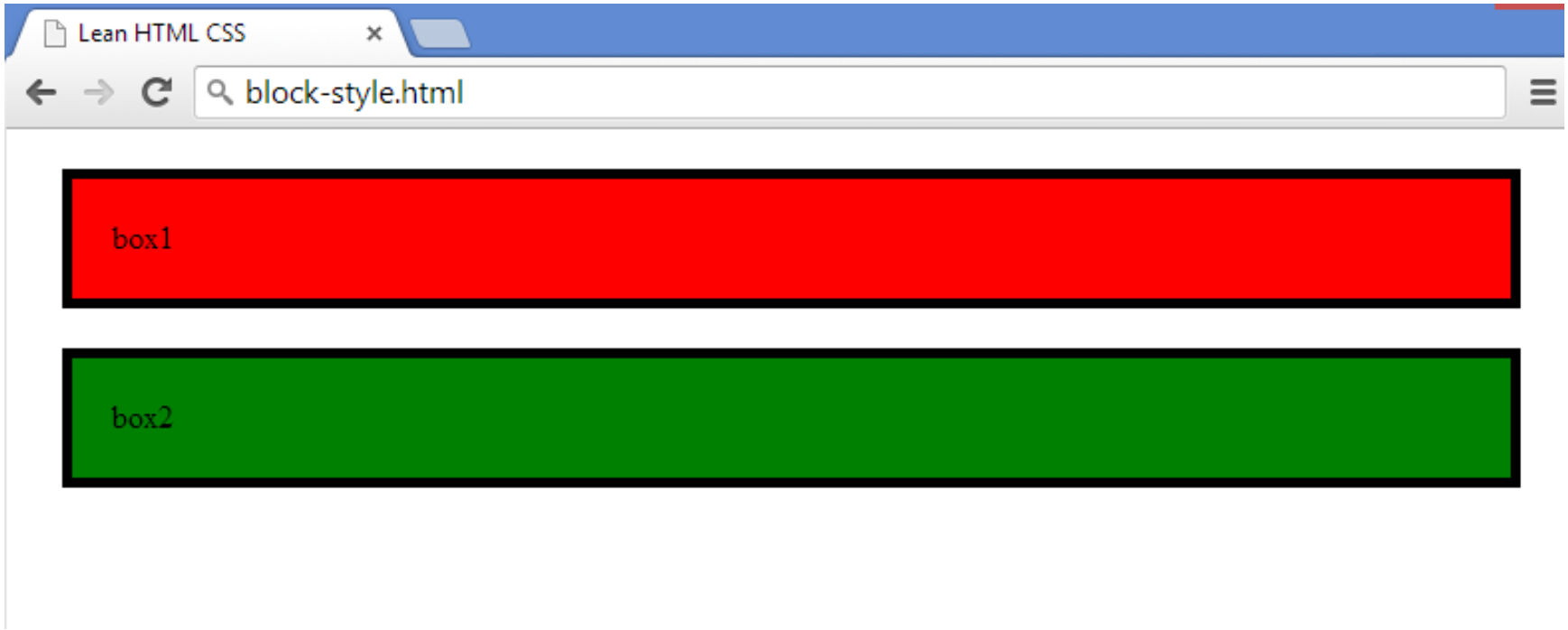
```
<div class="box1">box1</div>

<div class="box2">box2</div>
```

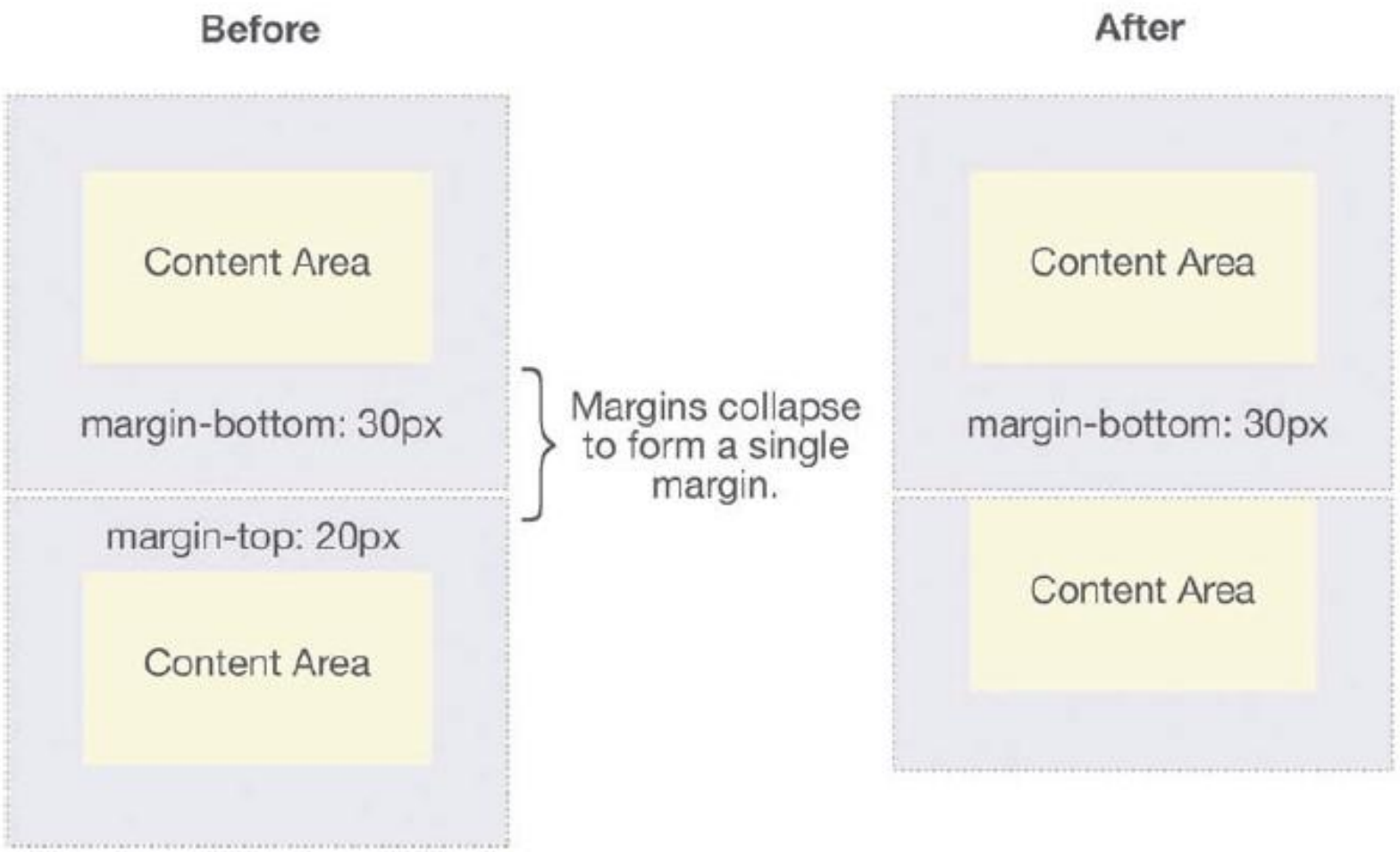
Box Model (Block)

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

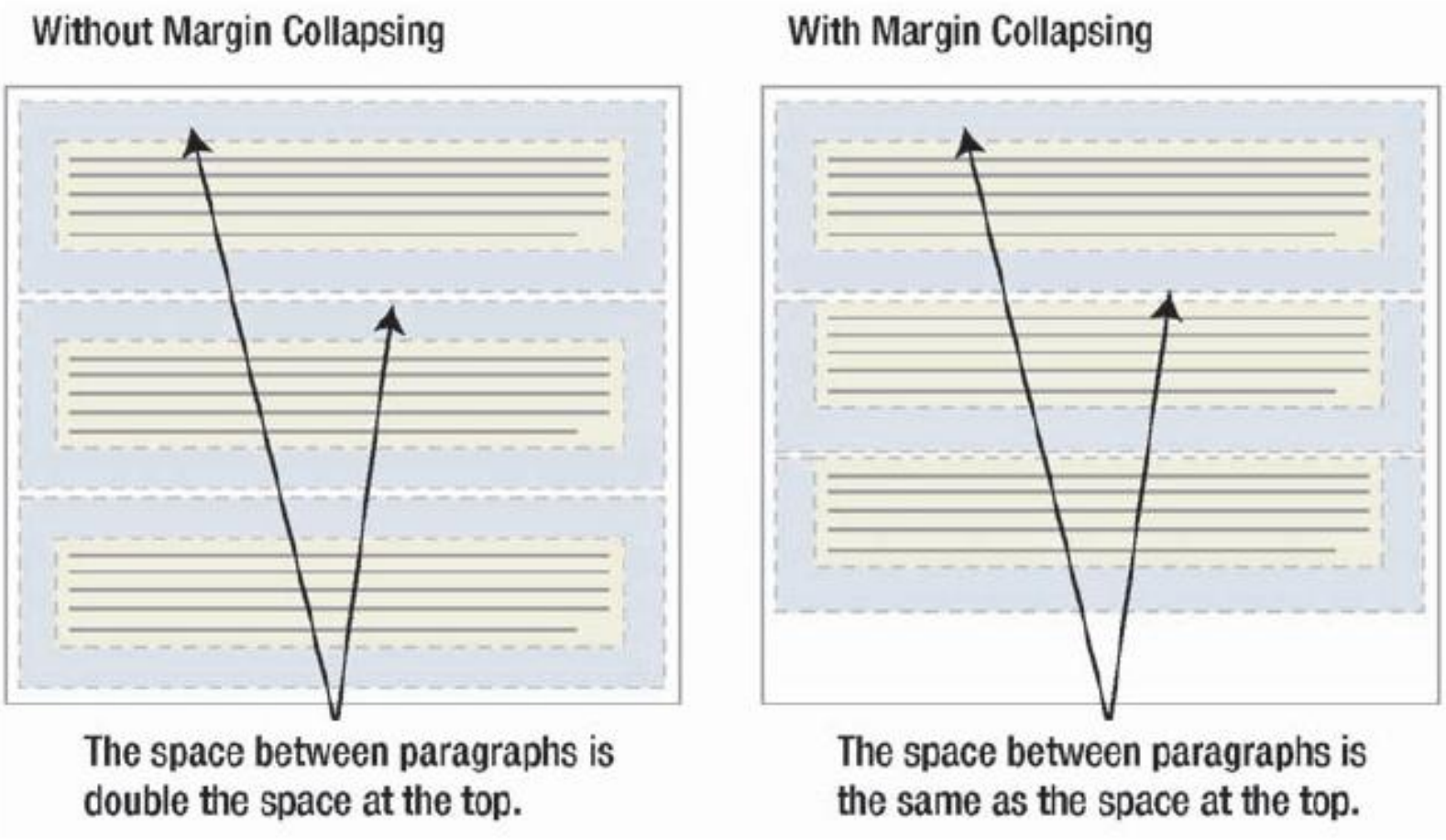
Preview



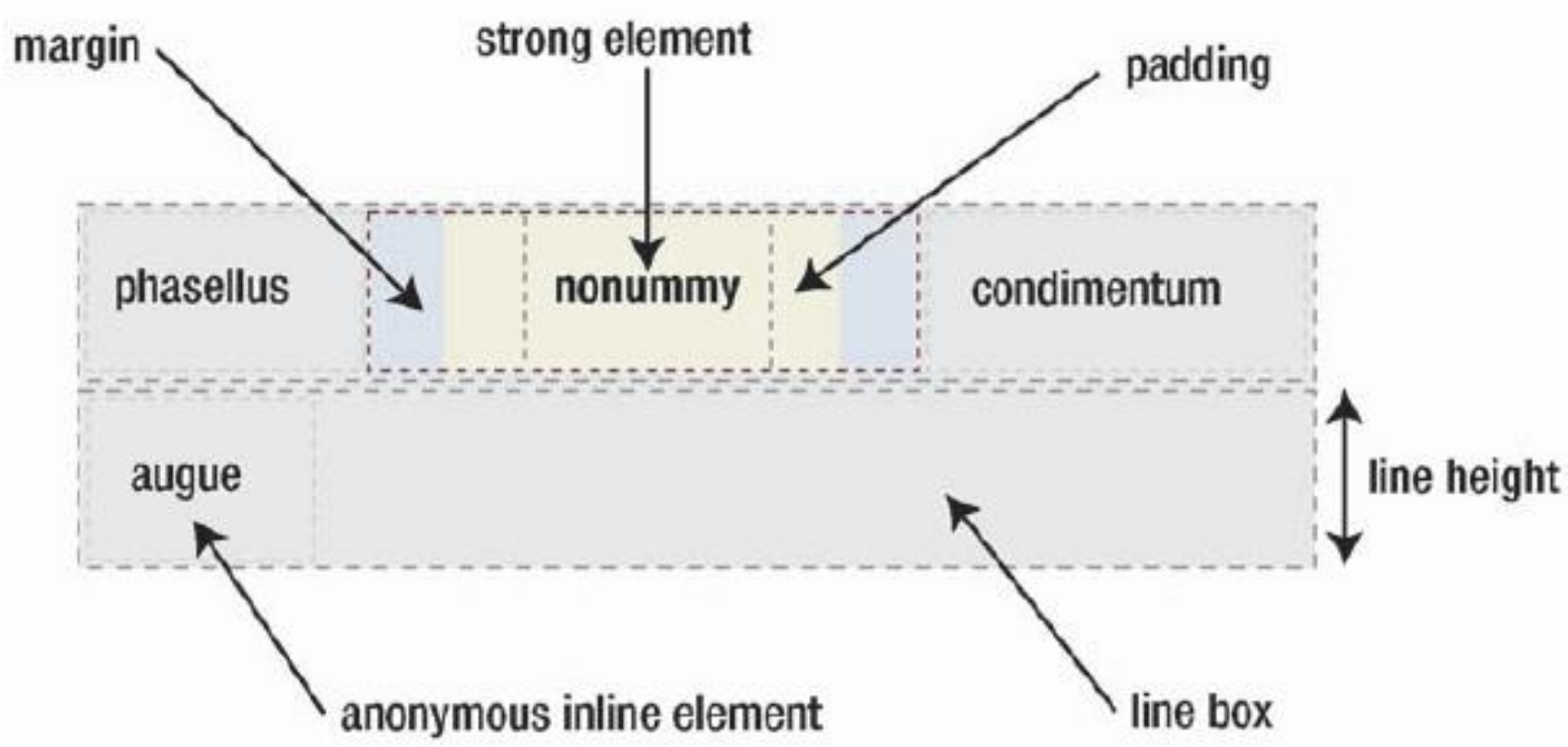
Margin collapse



Margin collapse



Box Model (Inline)

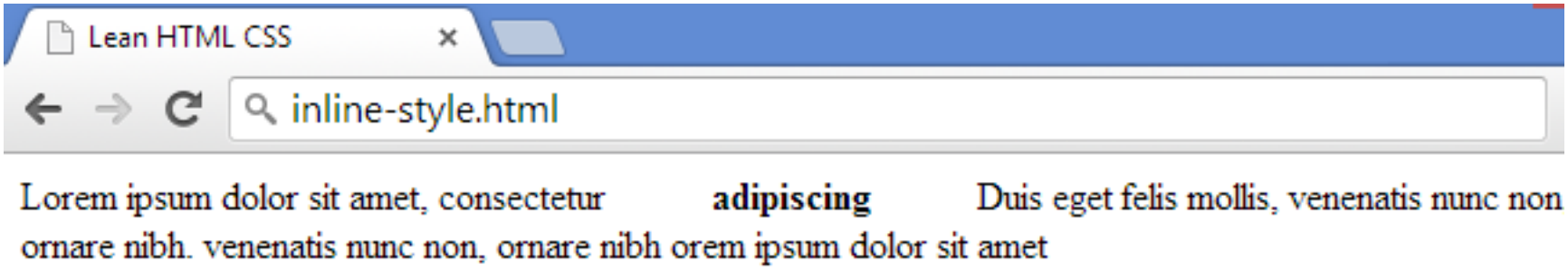


Box Model (Inline)

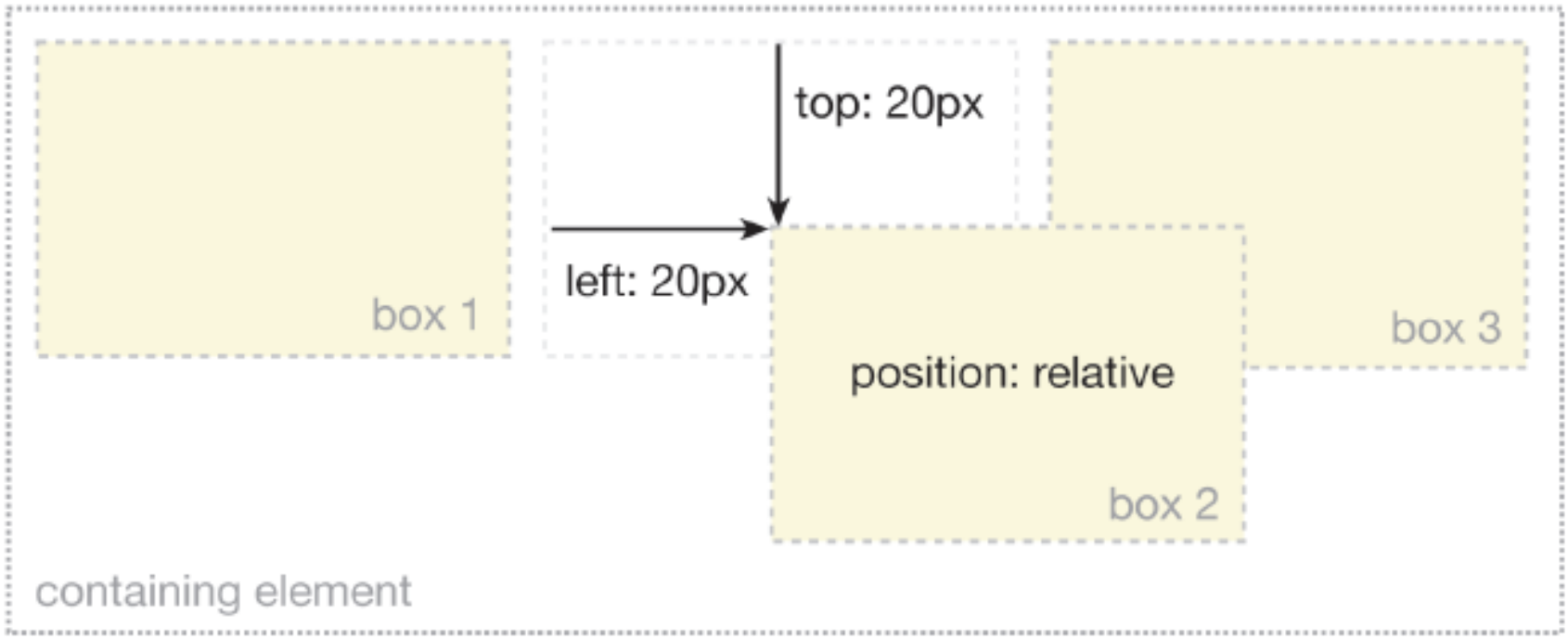
```
<p>Lorem ipsum dolor sit amet,
consectetur <strong>adipiscing</strong>
Duis eget felis mollis, venenatis
nunc non, ornare nibh. venenatis
nunc non, ornare nibh orem ipsum dolor sit
amet </p>
```

```
strong{
  padding:20px;
  margin:20px;
}
```

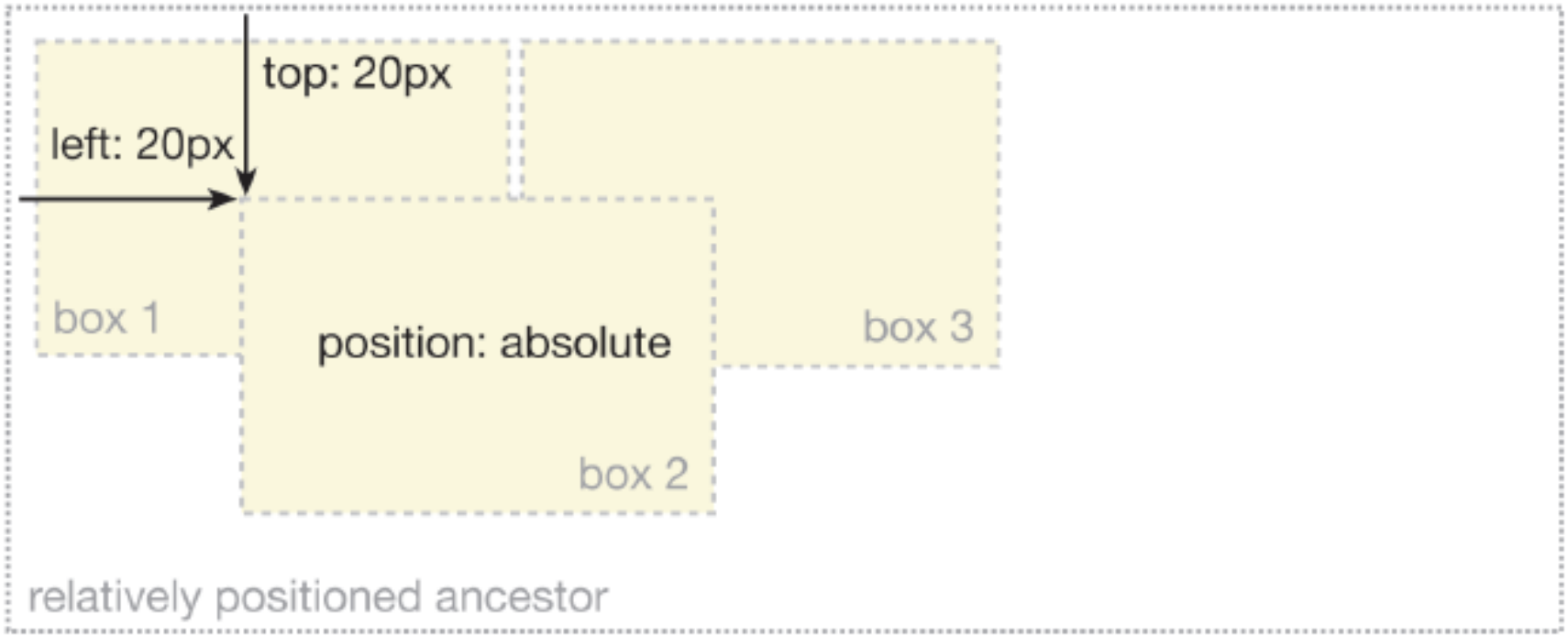
Preview


Lean HTML CSS
inline-style.html
Lorem ipsum dolor sit amet, consectetur adipiscing Duis eget felis mollis, venenatis nunc non ornare nibh. venenatis nunc non, ornare nibh orem ipsum dolor sit amet

Relative Position



Absolute Position

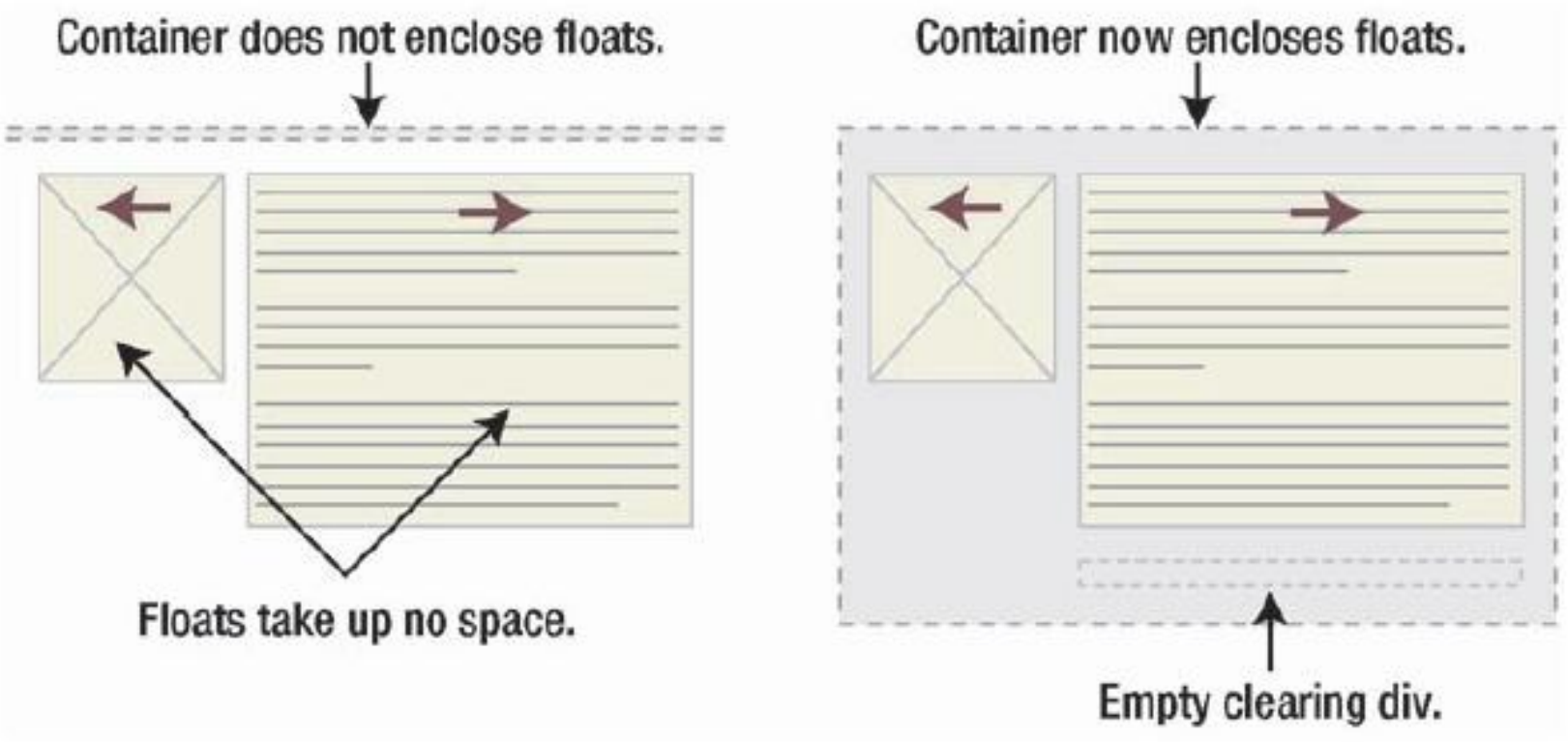


Fixed Position

Float

No Boxes Floated
box 1
box 2
box 3
Box 1 Floated Right
box 2
box 3
box 1

Float


Container does not enclose floats.
Floats take up no space.
Container now encloses floats.
Empty clearing div.

Bootstrap is Front-end Framework

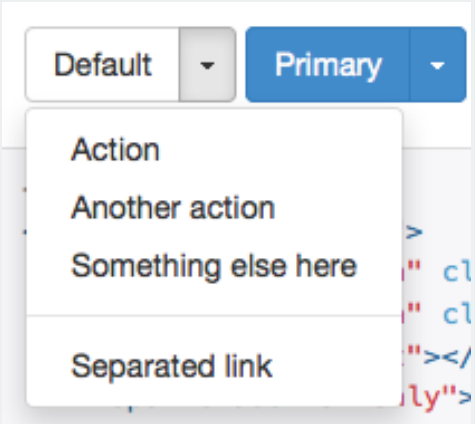
HTML, CSS, and JS framework for
developing responsive, mobile first projects on the web.

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

#	Column heading	Column heading	Column heading
1	Column content	Column content	Column content
2	Column content	Column content	Column content
3	Column content	Column content	Column content
4	Column content	Column content	Column content
5	Column content	Column content	Column content
6	Column content	Column content	Column content
7	Column content	Column content	Column content
8	Column content	Column content	Column content
9	Column content	Column content	Column content

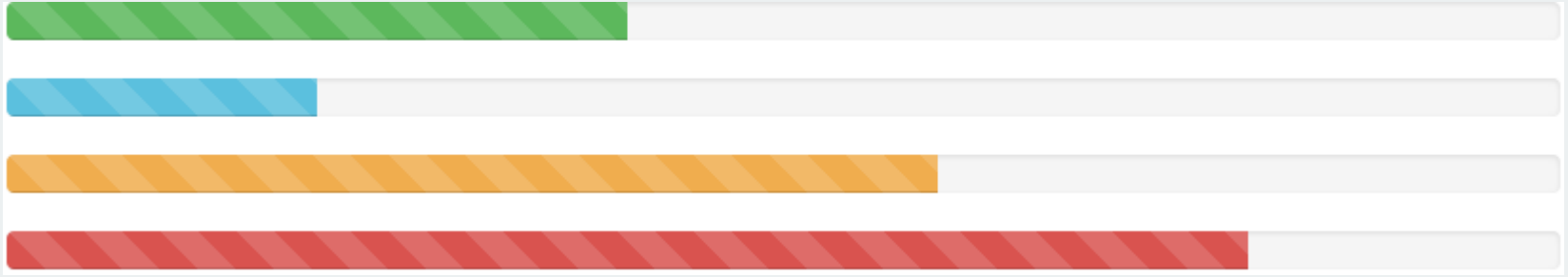


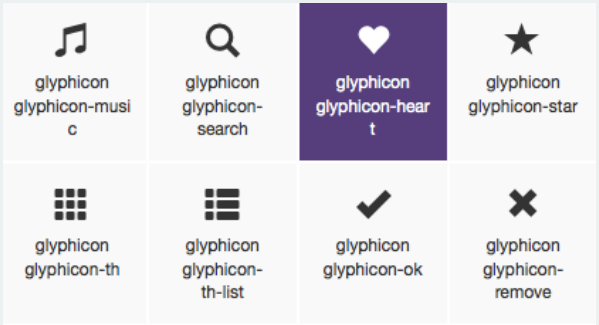
Oh snap! You got an error!

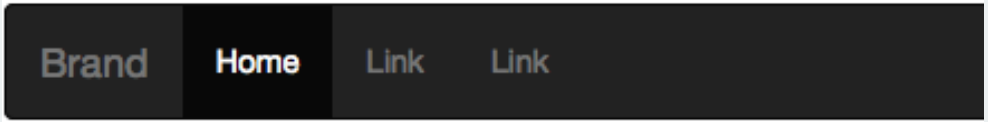
Change this and that and try again. Duis mollis, est non commodo luctus, nisi erat porttitor ligula, eget lacinia odio sem nec elit. Cras mattis consectetur purus sit amet fermentum.

Take this action | Or do this

more EXAMPLES
of Bootstrap Elements

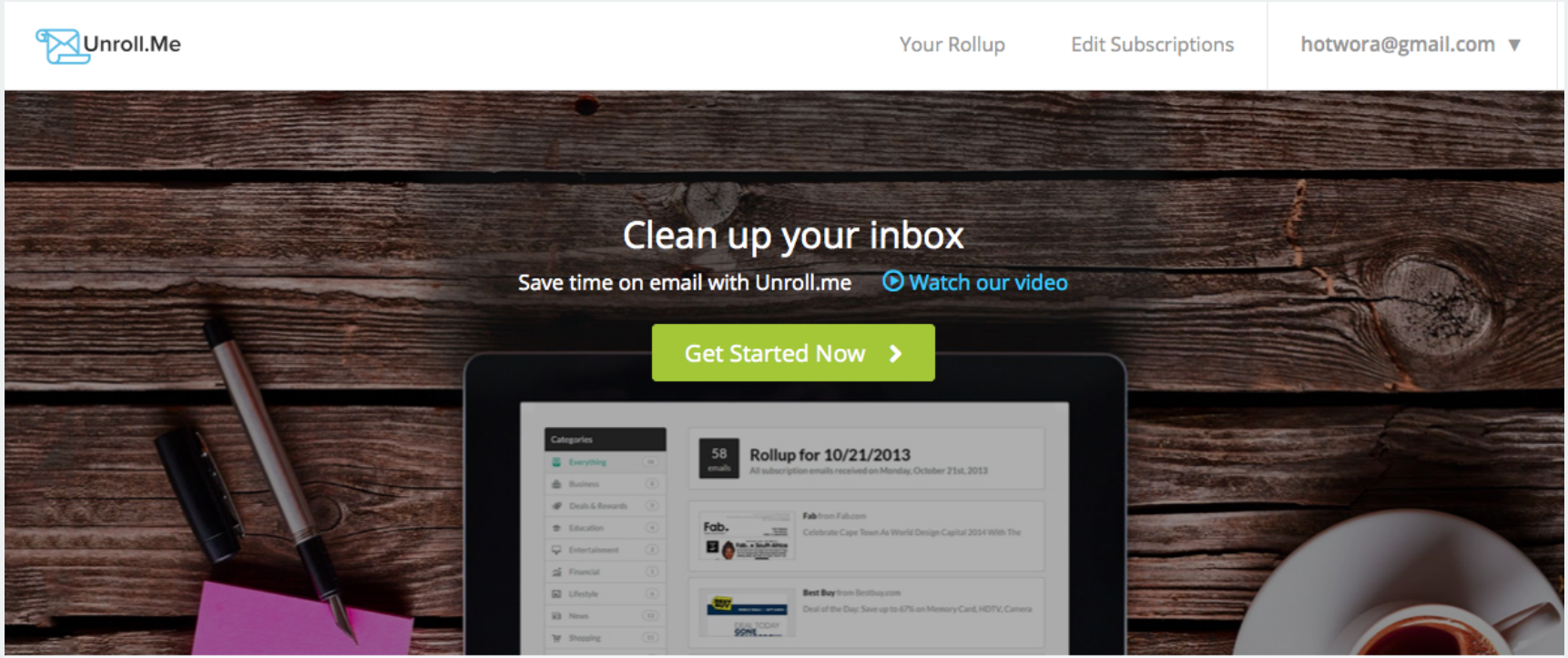






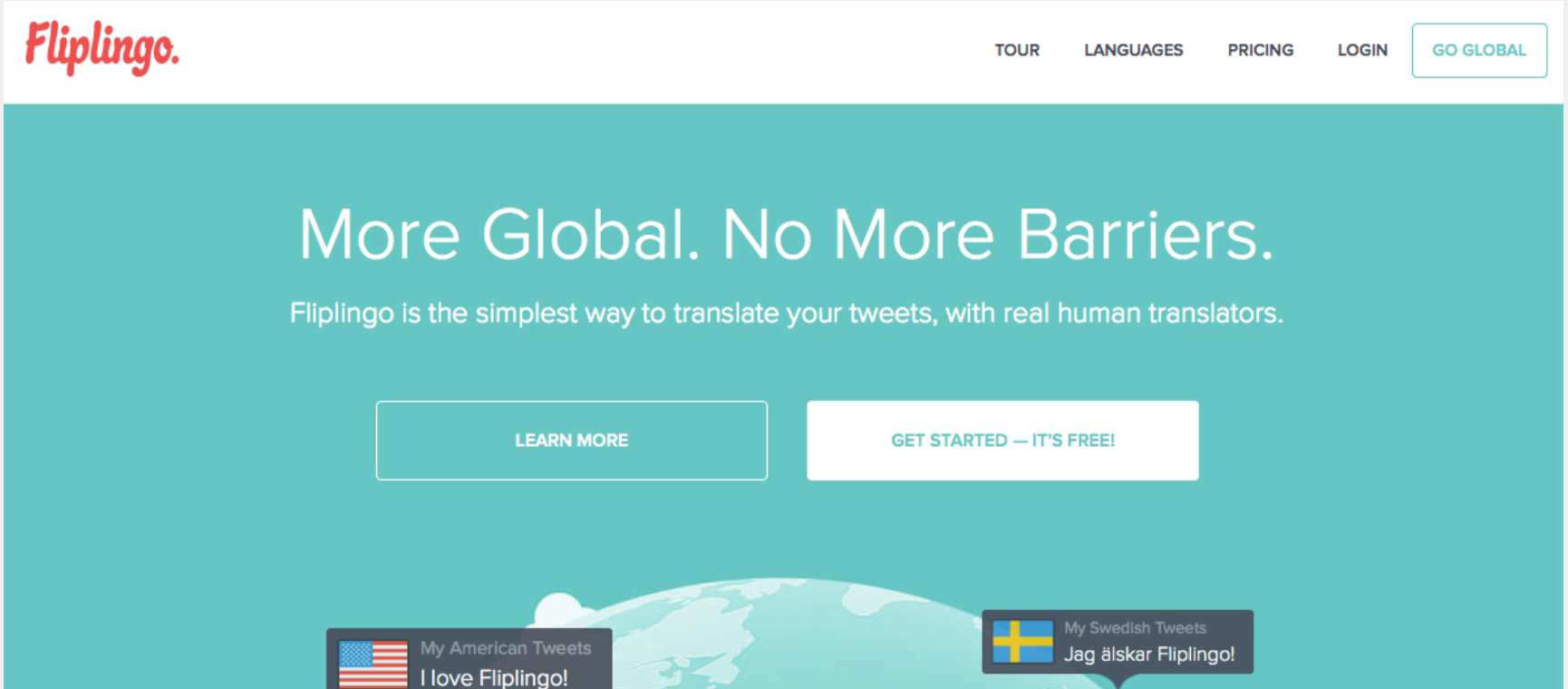
Websites created by Bootstrap

http://unroll.me



Websites created by Bootstrap

www.fliplingo.com



Why Bootstrap?

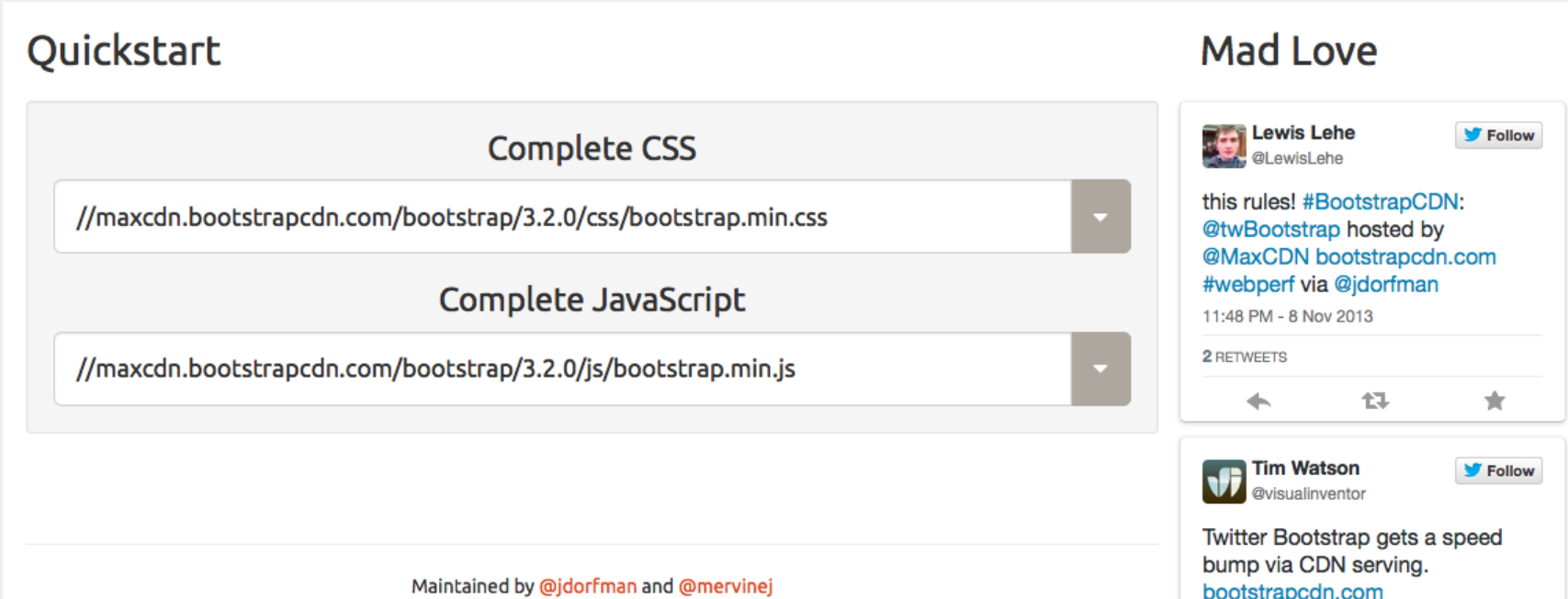
- Save 100+ hours of coding
- Easy to use web elements
- Quick responsive prototype / website
- Great documentation

Bootstrap Package

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

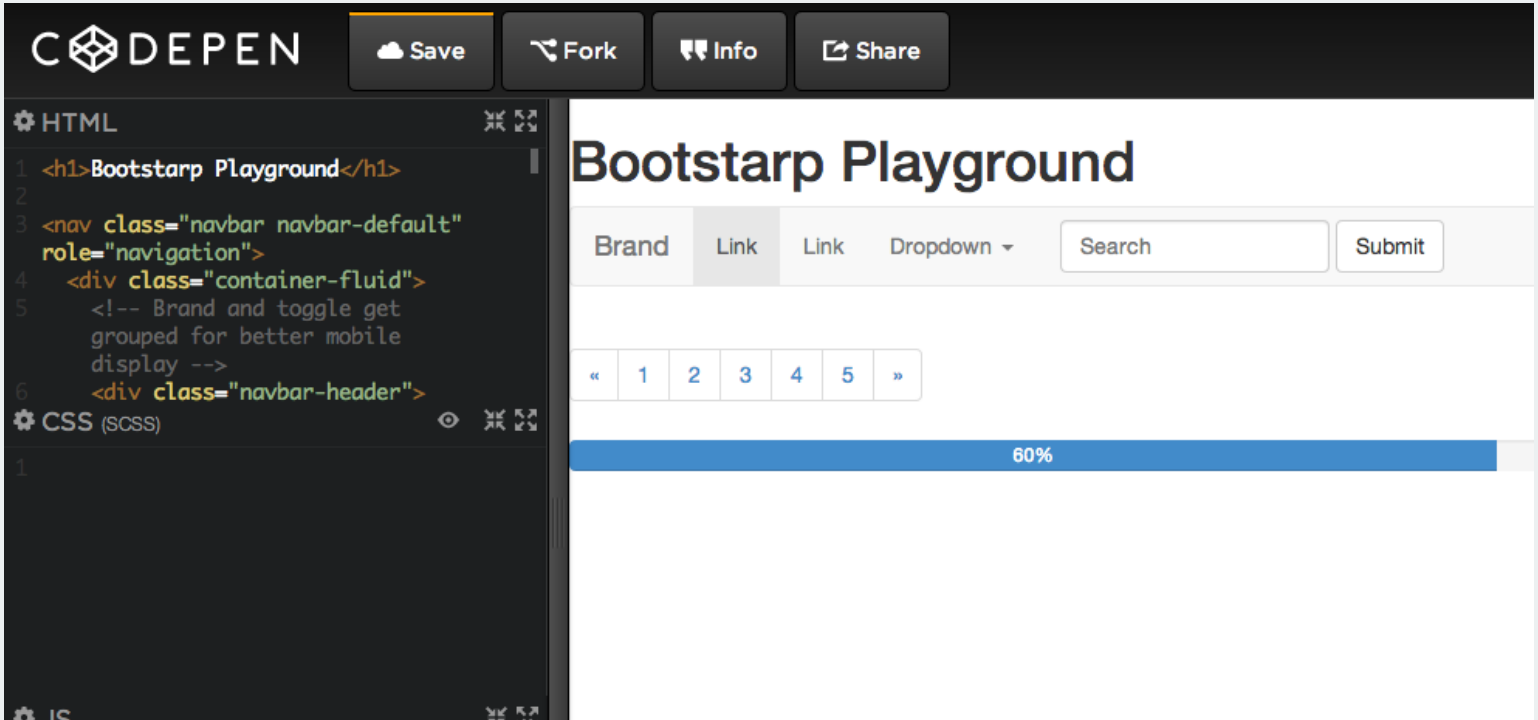
Bootstrap CDN

www.bootstrapcdn.com



Bootstrap Playground

http://bit.ly/bs-play



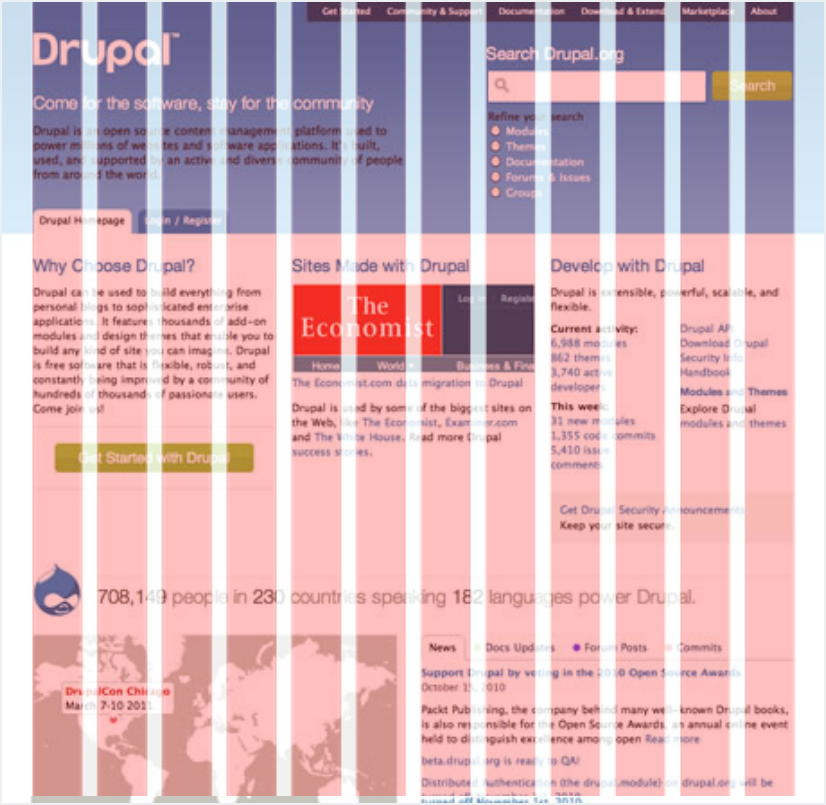
Bootstrap Components

Start Workshop

What is Grid in web design?

What is Grid in web design?

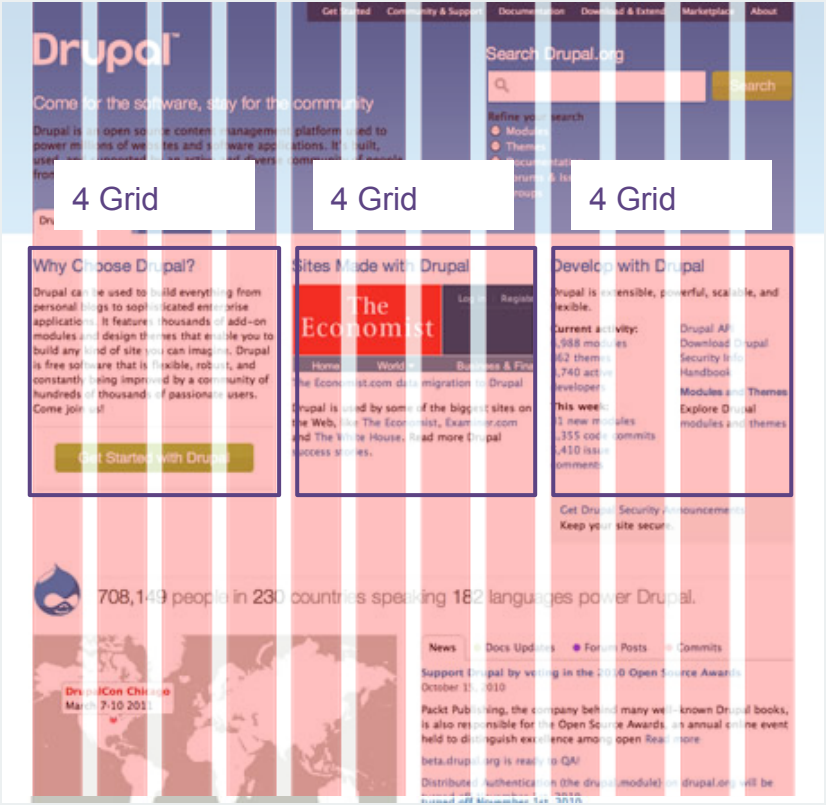
1 2 3 4 5 6 7 8 9 10 11 12



12 Grid

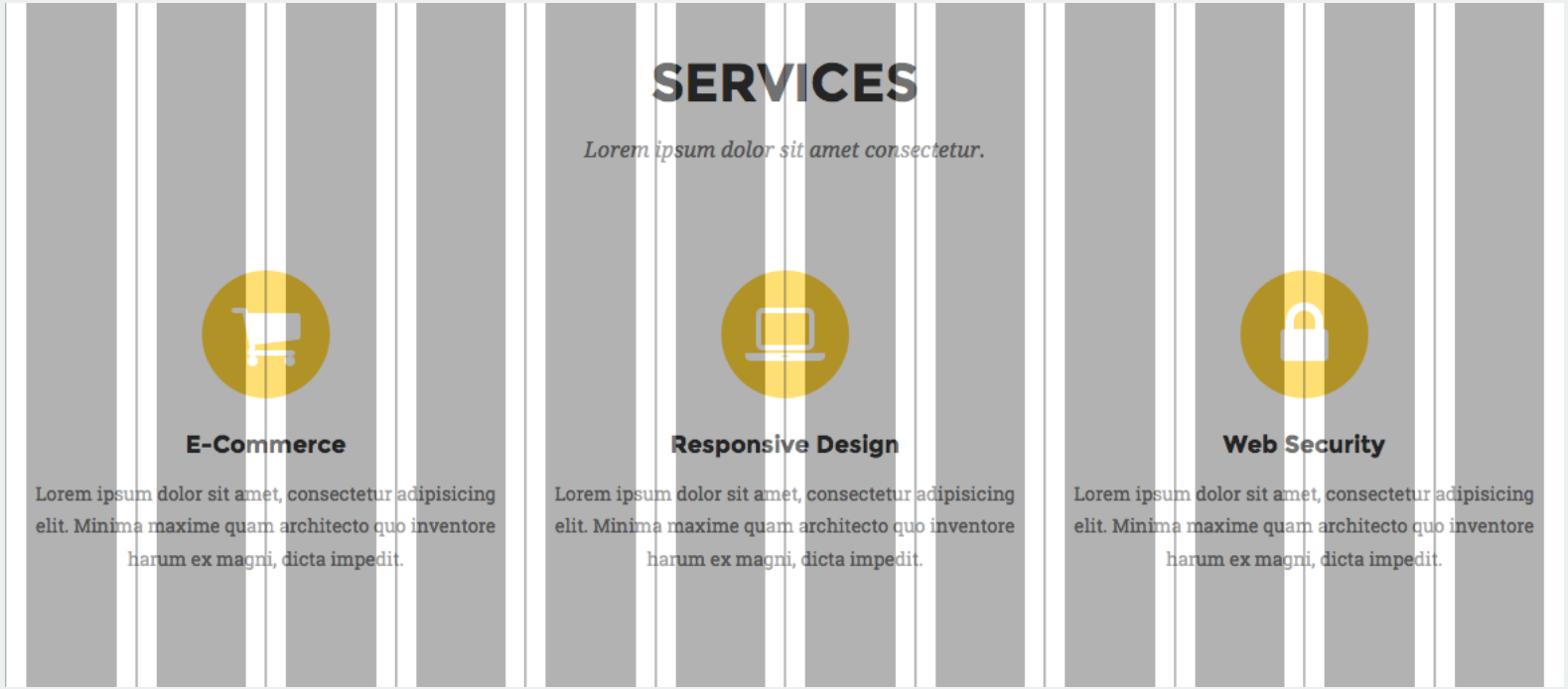
What is Grid in web design?

1 2 3 4 5 6 7 8 9 10 11 12



4 Grids x 3 Columns
= 12 Grids

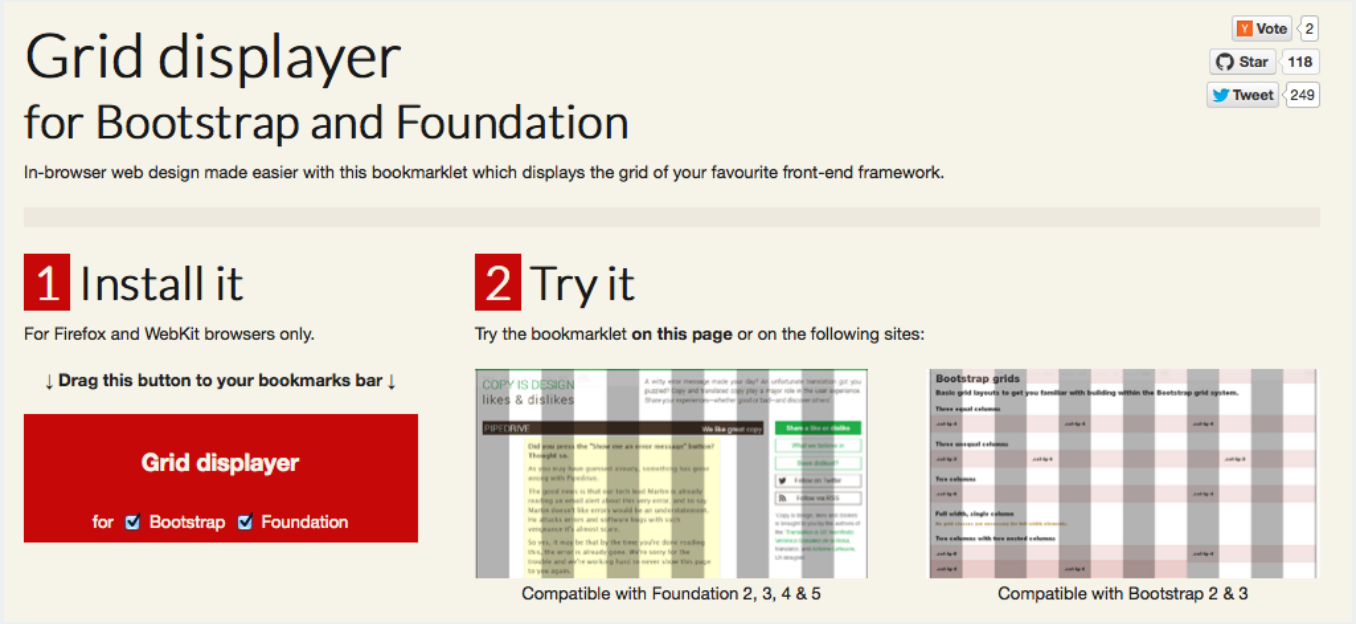
Bootstrap Grid



12 Responsive Grid

Grid Overlay
for Bootstrap & Foundation

http://bit.ly/grid-overlay



4 Sizes of Bootstrap Grid

Size Name	Screen Size	CSS Class
Extra Small Devices (Phone)	0 - 767 px	.col-xs-1 ~ .col-xs-12
Small Devices (Tablet)	768 - 991 px	.col-sm-1 ~ .col-sm-12
Medium Devices (Desktop)	992 - 1219 px	.col-md-1 ~ .col-md-12
Large Devices (Large screen desktop)	1200px +	.col-lg-1 ~ .col-lg-12

4 Sizes of Bootstrap Grid

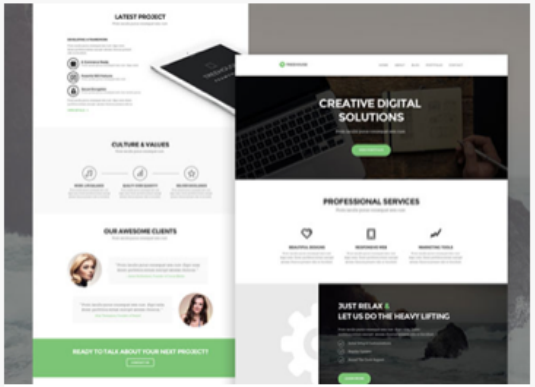
Size Name	Screen Size	CSS Class
Extra Small Devices (Phone)	0 - 767 px	.col-xs-1 ~ .col-xs-12
Small Devices (Tablet)	768 - 991 px	.col-sm-1 ~ .col-sm-12
Medium Devices (Desktop)	992 - 1219 px	.col-md-1 ~ .col-md-12
Large Devices (Large screen desktop)	1200px +	.col-lg-1 ~ .col-lg-12

Bootstrap Grid Example

http://bit.ly/bs-agency

Round Icons
Graphic Design

Startup Framework
Website Design



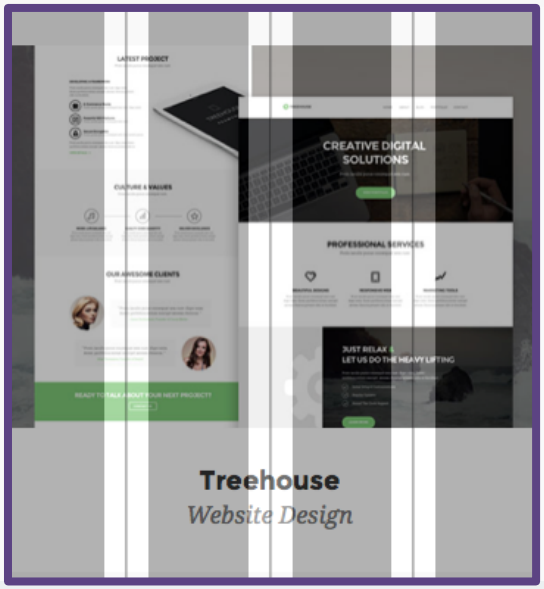
Treehouse
Website Design

How many grids in each box?

Bootstrap Grid Example

Round Icons
Graphic Design

Startup Framework
Website Design



Treehouse
Website Design

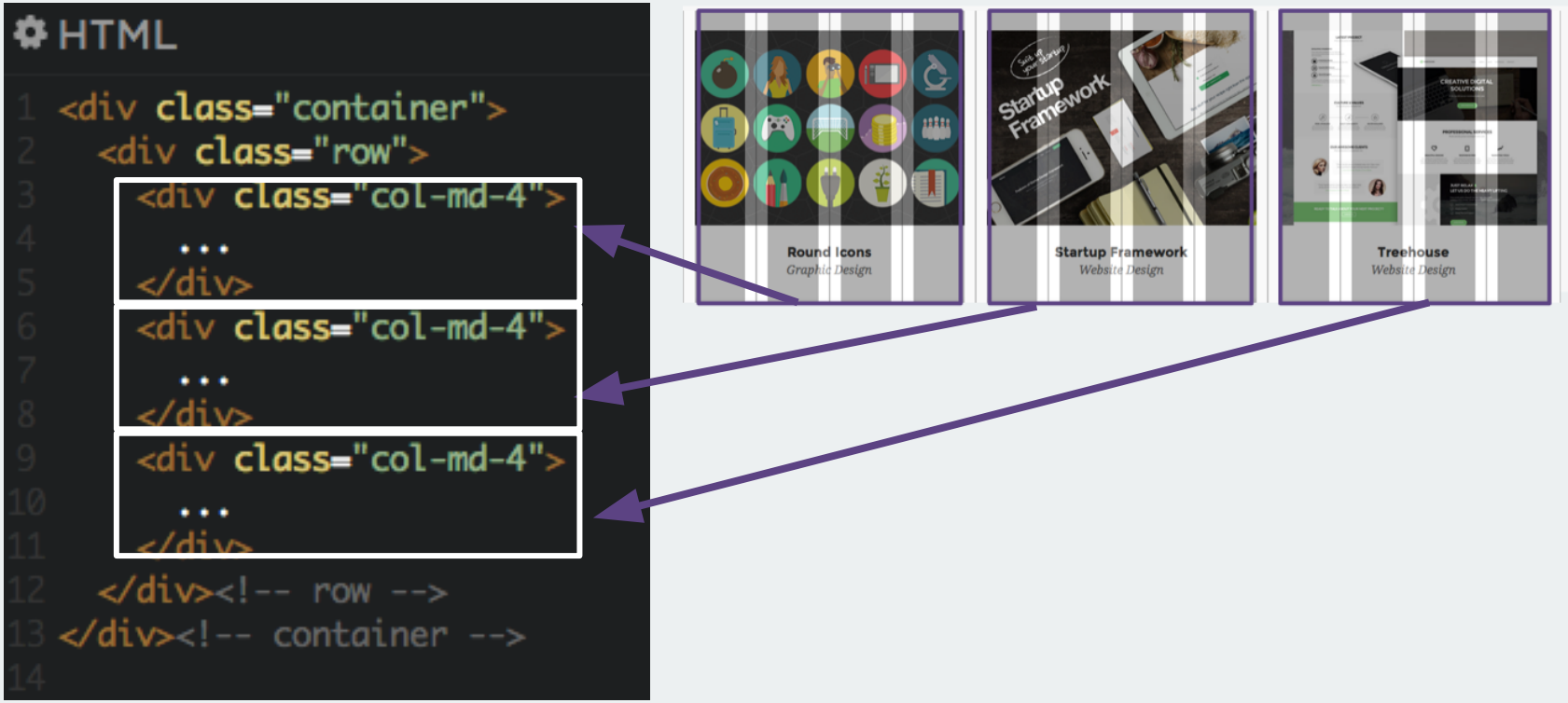
4 grids x 3 Columns

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>
  </div><!-- row -->
</div><!-- container -->
```

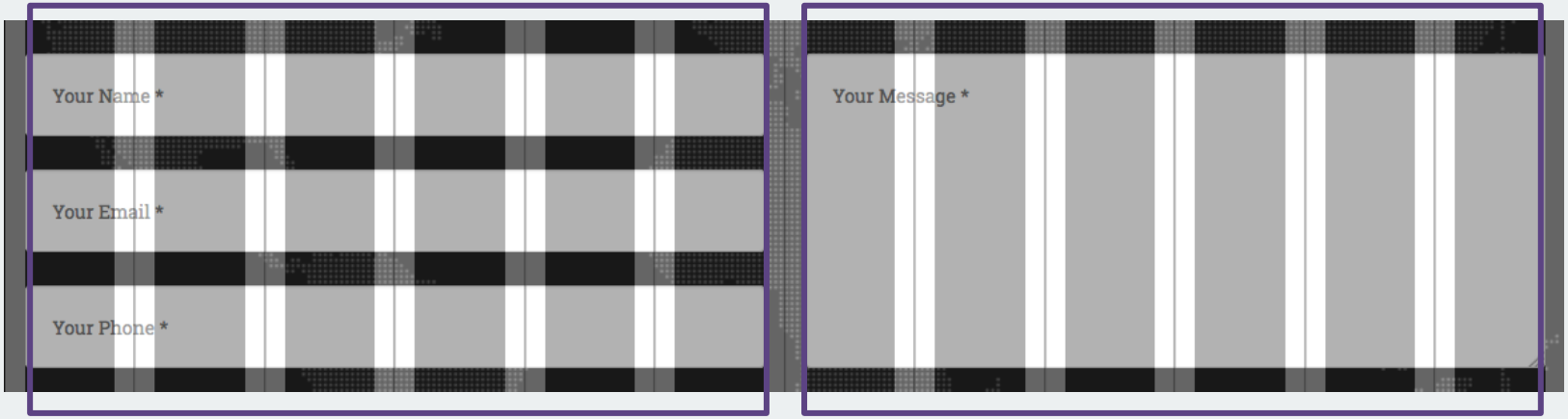
Bootstrap Grid Example



Bootstrap Grid Example 2

How many grids in each box?

Bootstrap Grid Example 2



6 grids x 2 Columns

Bootstrap Grid Example 2

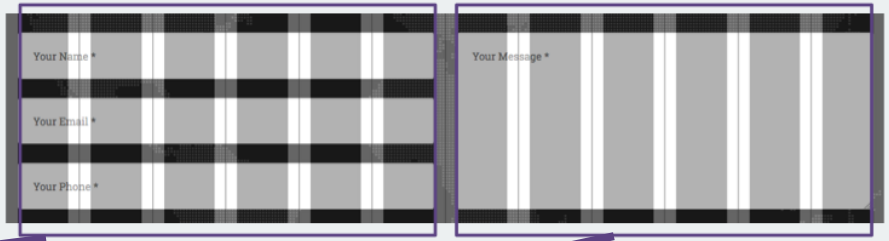
HTML

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

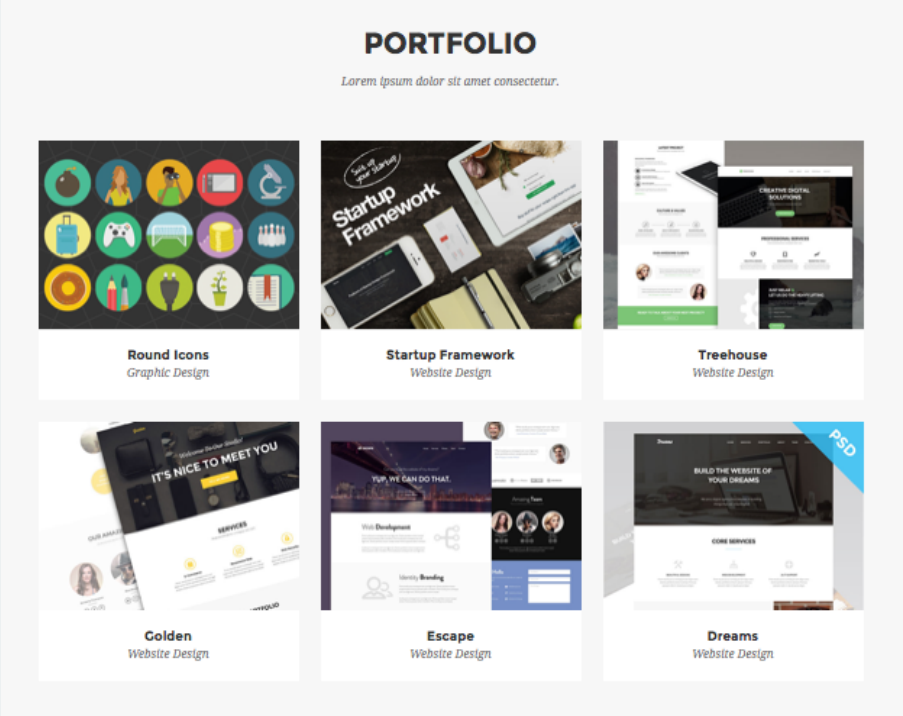
Bootstrap Grid Example 2

HTML

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

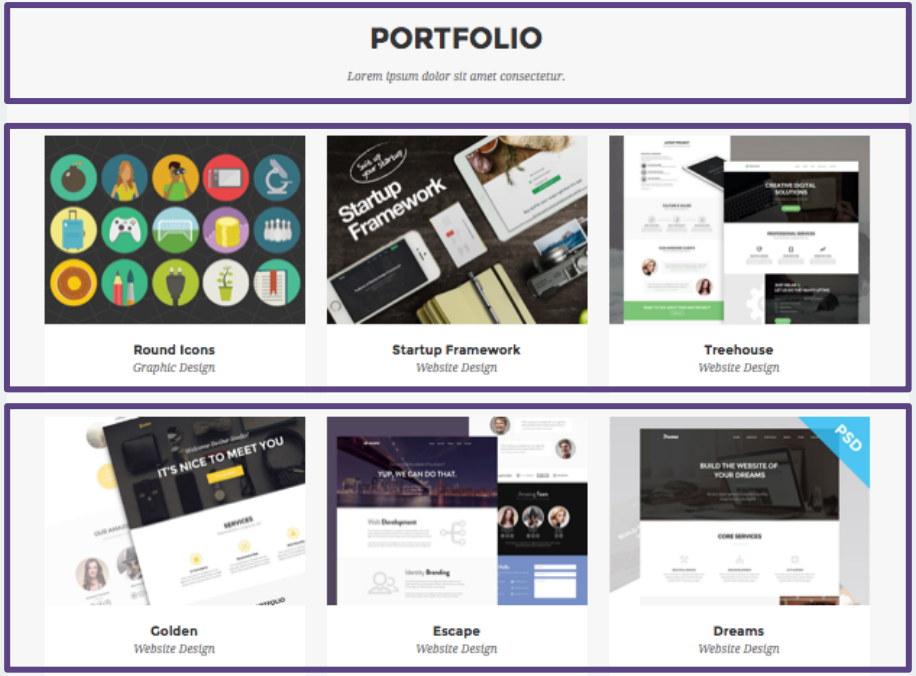


Bootstrap Row



1 Row = 12 Grids

Bootstrap Row



3 Rows

Bootstrap Row Example

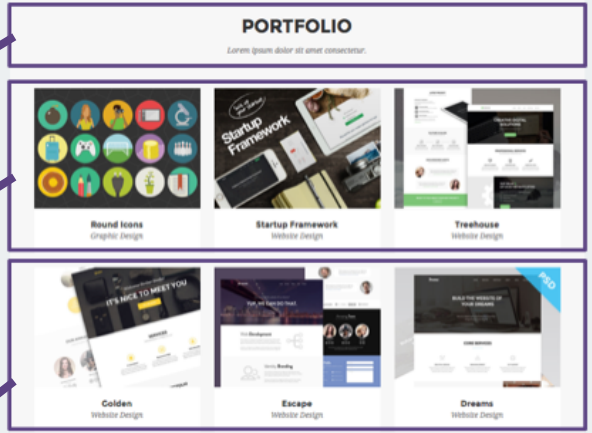
HTML

```
<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 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

Round Icons
Graphic Design

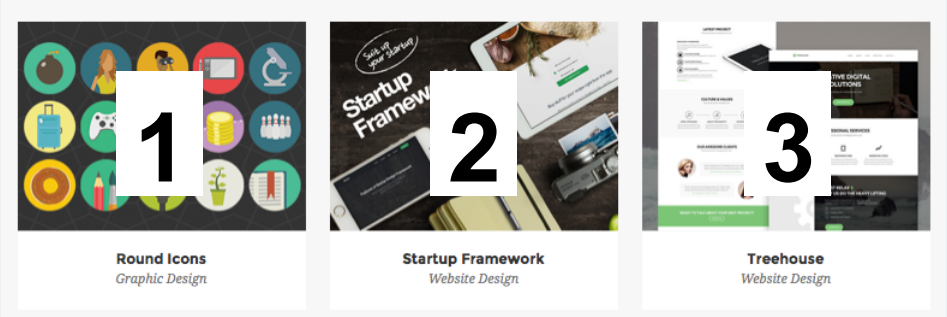
Startup Framework
Website Design

Treehouse
Website Design

Columns will stack when responsive

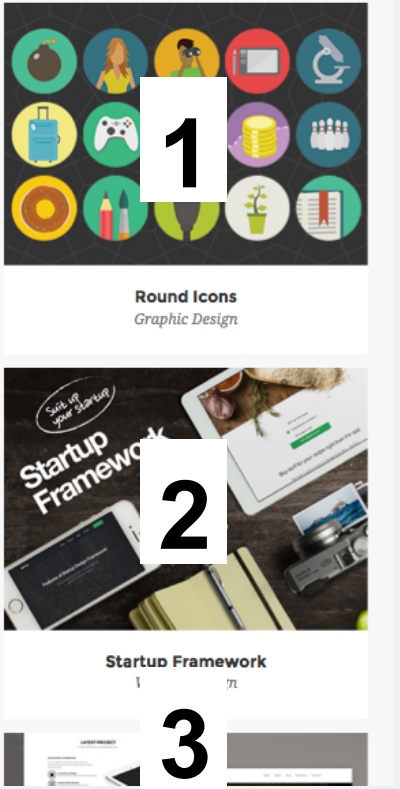
Bootstrap Responsive Grid

Desktop



Columns stack on mobile

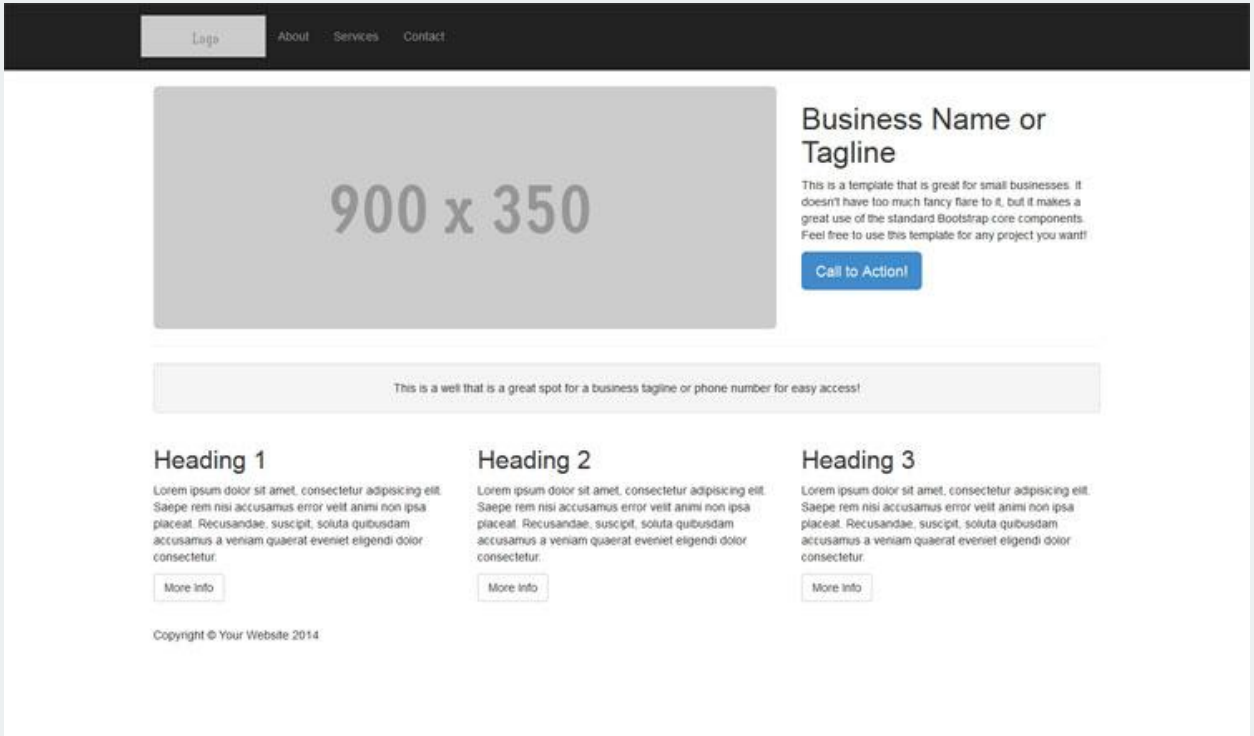
Mobile



Bootstrap Grid Workshop

3 Easy Steps:

1. Add container
2. Add row
3. Add columns

Bootstrap Grid Workshop

http://bit.ly/bs-business