# HTML & CSS: LEVEL 1

Instructor: Beck Johnson

Week 5

# 📋 SESSION OVERVIEW

- Review classes, ids, and floats

- Positioning

- Tables

- Fonts

- Evaluations

# REVIEW!

# {} CLASSES AND IDS

You can add `class` and `id` to any HTML element to identify it for styling.

- You decide the `class` and `id` values – be descriptive!

```
<p class="important">Big text</p>

<p class="anyLettersOrNumbersOr_Or-">Still
totally valid</p>
```

# {} CLASSES

**Multiple** elements can have the same `class`

In CSS, target a class with a **period**

```
.kittens { color: gray; }

<p class="kittens">This will be gray.</p>
```

Only **one** element per page can use the same id

In CSS, target an id with a **hash**:

```
#kittenContainer { color: gray; }
```

```
<div id="kittenContainer">This will be
gray.</div>
```

If you think it's likely or possible that you'll want to apply the same style to multiple things, definitely use `class`

If your element is guaranteed to be the only one on the page, you can use `id` – or you can still use `class`
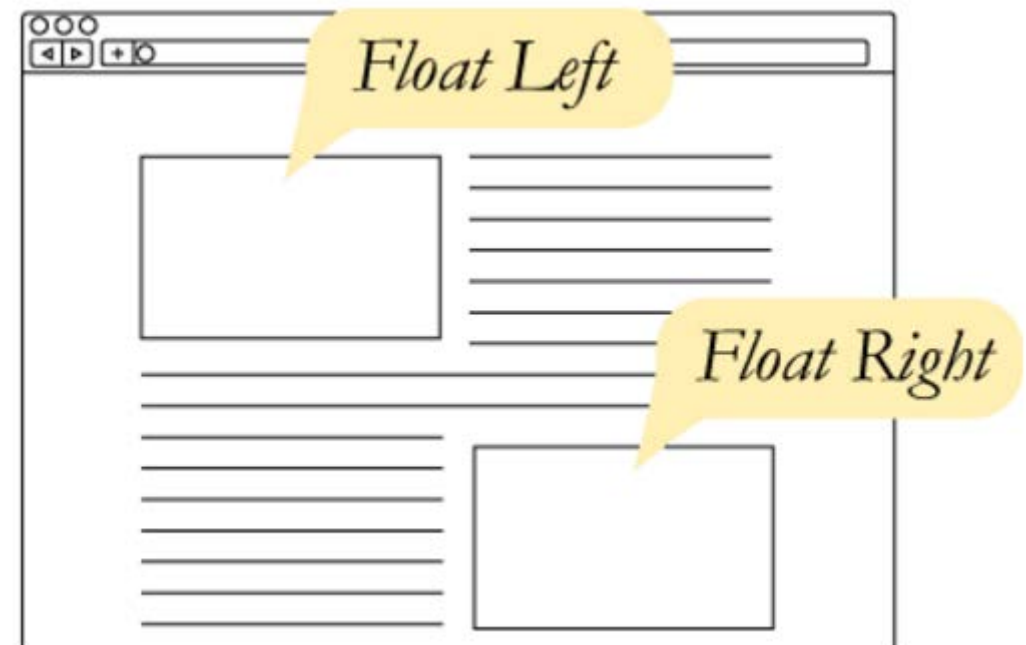
If your element needs to be linked to directly, use `id`

# CSS FLOATS

The `float` property takes an element out of the normal flow and "floats" it to the left or right side of its container.

- This allows other content to flow around it
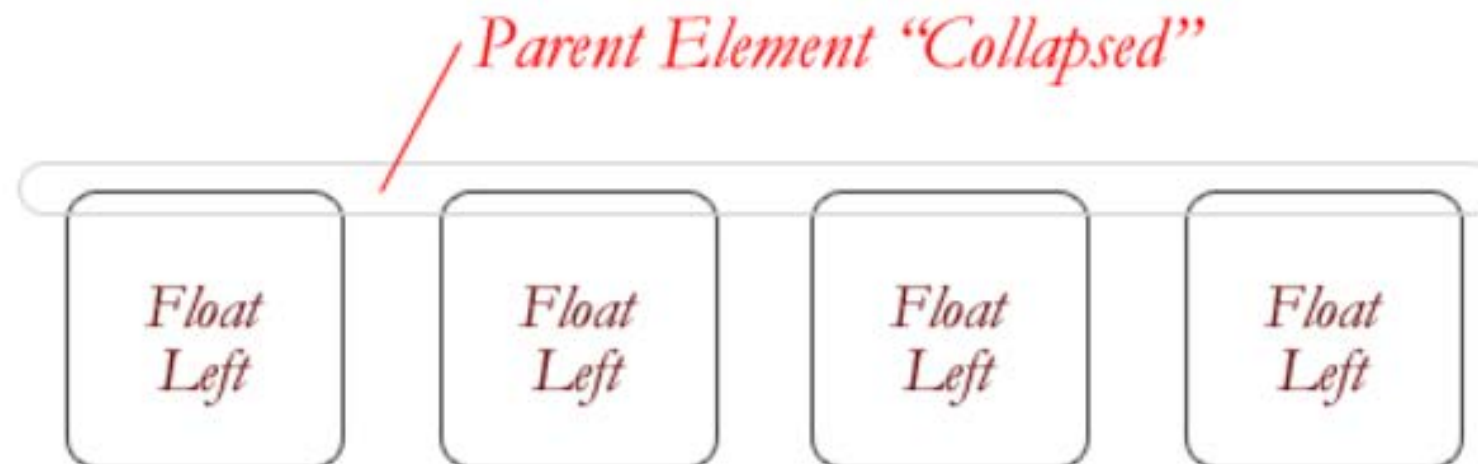
```
img { float: left; }
```

The three values for `float` are:

- `left`
- `right`
- `none`

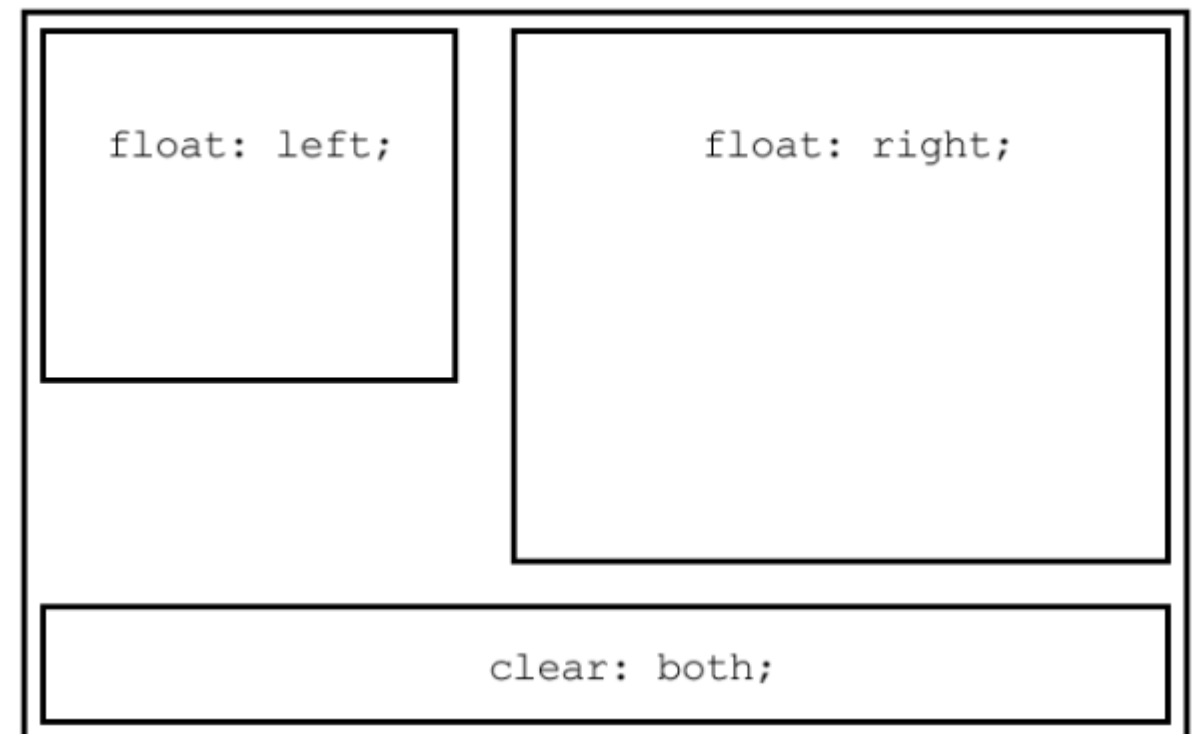If everything in a container is floated, then the container thinks it's empty.



*Parent Element "Collapsed"*

| | | | |
|---|---|---|---|
| *Float Left* | *Float Left* | *Float Left* | *Float Left* |

The `clear` property is the sister property to `float`

- It doesn't do much until there are floated elements on the page

- An element with `clear` applied to it will force itself **below** the floated element

- Everything after that will be back in the normal flow

- This "stretches" out the container and keeps it from collapsing

float: left;

float: right;

clear: both;

`overflow` is a CSS property that governs how content looks when it breaks out of its container.

By default, elements have `overflow: visible`, which means all content is fully visible.

`overflow: auto` adds scrollbars when the content is bigger than its container.

# QUESTIONS?

# WEB FONTS

Remember that `font-family` looks for a font installed on the user's local machine.

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

If the Tahoma font isn't found, the browser will default to a generic sans-serif font instead.

What if you want to use an interesting font that most people aren't likely to have installed?

# WEB FONTS

The absolutely easiest way to get custom fonts is to link to a CDN font stylesheet in the head of your page:

```html
<link href="https://fonts.googleapis.com/css?family=Roboto" rel="stylesheet">
```

CDN means "Content Delivery Network"

• A fancy way of saying a big, reliable company hosts the file for you

• Google hosts 100s of free fonts

# WEB FONTS

```html
<link href="https://fonts.googleapis.com/css?family=Roboto"
rel="stylesheet">
```

Once you include a font stylesheet from a CDN, you can refer to the font in a `font-family` rule, just like you would a web-safe font:

```css
p { font-family: Roboto, sans-serif; }
```

# WEB FONTS



When you choose a font Google will tell you what name to use.

# WEB FONTS

**Roboto**
Christian Robertson (12 styles)

All their equipment and instruments are alive.

**Baloo Bhaina**
Ek Type (1 style)

**A red flair silhouetted the jagged edge of a wing.**

Try typing directly into the text fields. GOT IT

**Diplomata SC**
Eduardo Tunni (1 style)

I WATCHED THE STORM, SO BEAUTIFUL YET TERRIFIC.

**Open Sans**
Steve Matteson (10 styles)

Almost before we knew it, we had left the ground.

**Lato**
Łukasz Dziedzic (10 styles)

A shining crescent far beneath the flying vessel.

**Slabo 27px**
John Hudson (1 style)

It was going to be a lonely trip back.

**Ranga**
TipTopTyp (2 styles)

Mist enveloped the ship three hours out from port.

**Oswald**
Vernon Adams, Kalapi Gajjar, Cyreal (6 styles)

My two natures had memory in common.

**Roboto Condensed**
Christian Robertson (6 styles)

Silver mist suffused the deck of the ship.

**Source Sans Pro**
Paul D. Hunt (12 styles)

The face of the moon was in shadow.

**Montserrat**
Julieta Ulanovsky (18 styles)

She stared through the window at the stars.

**Griffy**
Neapolitan (1 style)

The recorded voice scratched in the speaker.

[https://fonts.google.com/](https://fonts.google.com/)
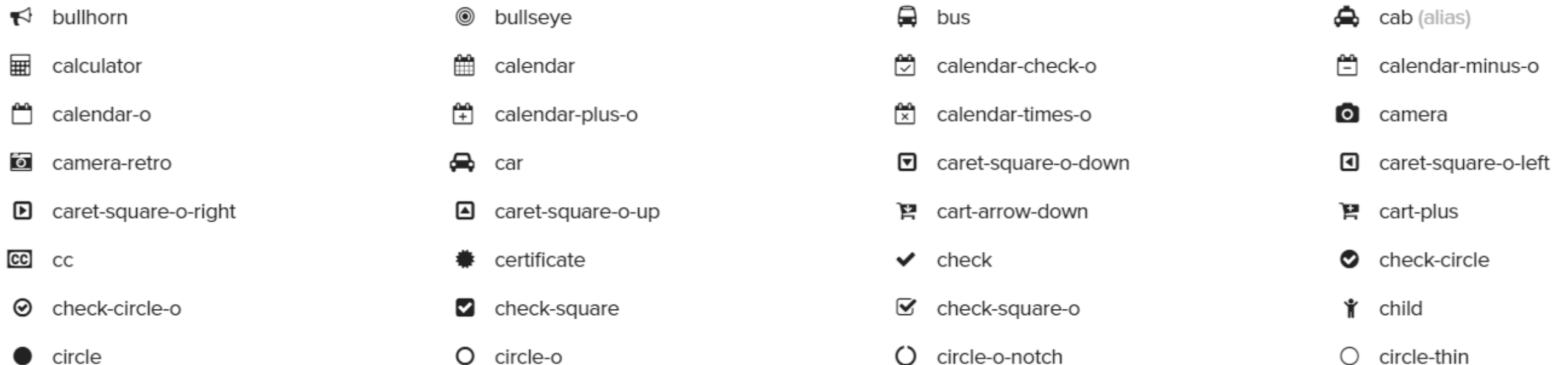
# WEB FONTS

## CONS:

- Relies on a 3rd party to provide assets

- If the request times out, the font won't download (so always provide fallbacks!)

## PROS:

- Extremely easy to get working

- Possibility that user already has the font in their web cache due to visiting other sites that use the same font

# ICON FONTS

[Font Awesome](#) is a free icon font that is used in many real-world projects.

| | | | |
|---|---|---|---|
| 📢 bullhorn | ◎ bullseye | 🚌 bus | 🚕 cab (alias) |
| 🖩 calculator | 📅 calendar | 📅 calendar-check-o | 📅 calendar-minus-o |
| 📅 calendar-o | 📅 calendar-plus-o | 📅 calendar-times-o | 📷 camera |
| 📷 camera-retro | 🚗 car | ▾ caret-square-o-down | ◂ caret-square-o-left |
| ▸ caret-square-o-right | ▴ caret-square-o-up | 🛒 cart-arrow-down | 🛒 cart-plus |
| cc cc | ✹ certificate | ✔ check | ✔ check-circle |
| ✔ check-circle-o | ☑ check-square | ☑ check-square-o | ⅄ child |
| ● circle | ○ circle-o | ○ circle-o-notch | ○ circle-thin |

An icon font means letters have been replaced with vector images

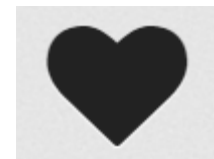- So to color or re-size icons on your site, just use the CSS `font` properties we already learned.

# ICON FONTS

To quickly start using Font Awesome, add this stylesheet to the head of your page:

```
<link href="http://maxcdn.bootstrapcdn.com/font-awesome/latest/css/font-awesome.min.css" rel="stylesheet">
```

Click an icon from the **Font Awesome** website, and copy the markup they provide, like:

```
<span class="fa fa-heart"></span>
```

# FONT AWESOME

You can put a Font Awesome class on any element:

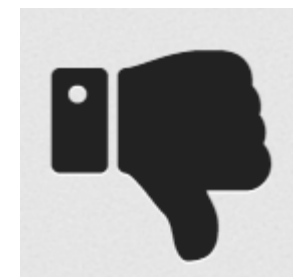```
<span class="fa fa-paw"></span>
```

```
<i class="fa fa-cog"></i>
```

```
<h1 class="fa fa-thumbs-down"></h1>
```

# WEB FONTS

The other way to ensure people see the correct font is to download the font onto their computer when they load your page.

- You do this by first downloading a font into a folder on your website

- There are free fonts available for download at websites like Font Squirrel or Font Spring

- After you have a copy of the font you can share it just like any other media file on your website

Some considerations with hosting fonts:

- Have to consider licensing fees – many fonts cost money

- Some fonts prohibit commercial use, or limit the number of pageviews

# @FONT-FACE

If you download a font that you want to use on your site, add `@font-face` before any other styles:

```css
@font-face {
  font-family: MyWebFont;
  src: url('webfont.eot');
       url('webfont.eot?#iefix') format('embedded-opentype'),
       url('webfont.woff') format('woff'),
       url('webfont.ttf') format('truetype'),
       url('webfont.svg#svgFontName') format('svg');
}
```

(The good news is, font sites will provide this chunk of CSS for you to copy into your stylesheet!)

# @FONT-FACE

```css
@font-face {
  font-family: MyWebFont;
  src: url('webfont.eot');
       url('webfont.eot?#iefix') format('embedded-opentype'),
       url('webfont.woff') format('woff'),
       url('webfont.ttf') format('truetype'),
       url('webfont.svg#svgFontName') format('svg');
}
```

Different browsers support different font filetypes

- Modern browsers use woff
- IE needs eot
- Mobile devices need ttf or svg

PRACTICE TIME!

# ASSIGNMENT

Find a free font from **Google fonts** and use it on your site by including the font stylesheet.

- Apply the font to some (or all) elements on the page

Include the Font Awesome stylesheet.

- Display at least two different icons

- Make them different sizes and/or colors

- Bonus points: what other CSS can you apply to the icons?

CSS POSITIONING

# 3 WEB LAYOUT PROPERTIES

- `display:` dictates how elements behave within the box model

- `float:` moves elements around within the page flow

- `position:` takes elements entirely out of the page flow

# CSS POSITIONING

The `position` property specifies how an element is positioned on the page. Possible values are:

- `static`
- `fixed`
- `absolute`
- `relative`

The default `position` is `static`, which just means that the element obeys whatever its box model rules tell it to do.

# CSS POSITIONING

There are 4 directional properties that affect positioning:
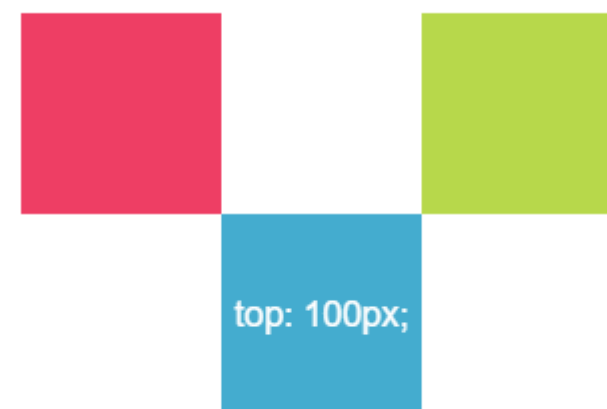
- `left`
- `right`
- `top`
- `bottom`

The value is a number (positive or negative) followed by a unit.

They define how far an element is offset that direction.

# CSS POSITIONING

`top` defines how far an element is offset from its original top edge.

- Positive `top` values push an element **down**
- Negative `top` values push an element **up**

top: -100px;

top: 100px;

# CSS POSITIONING

Similarly, `left` defines how far it's offset from its original left edge.

- Positive `left` values push an element **right**
- Negative `left` values push an element **left**


left: 100px;


left: -100px;

# POSITION: FIXED

`position: fixed` makes content "stick" to the browser window, regardless of where the user scrolls.
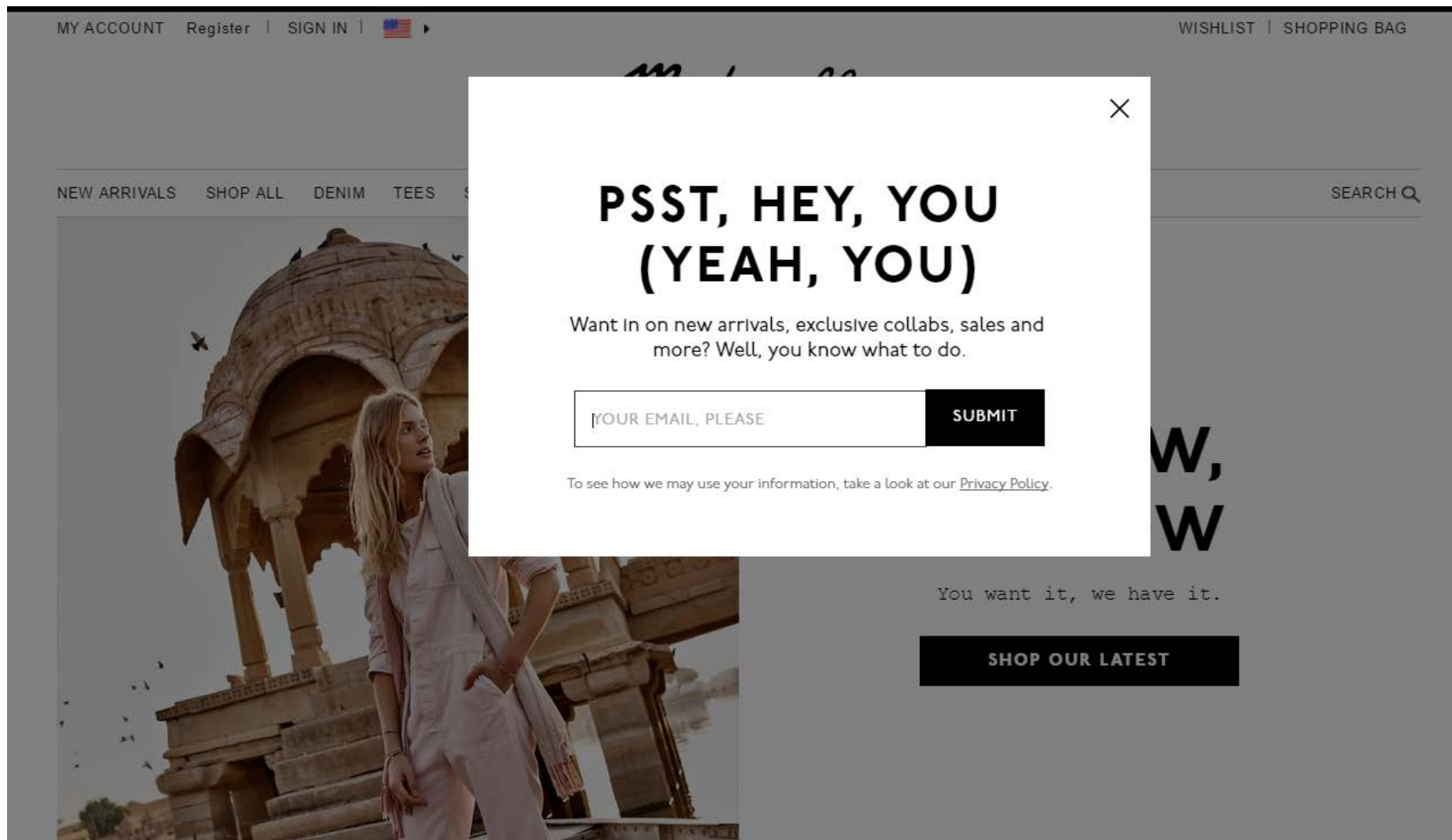
Commonly used to make headers, navigation menus, or sidebars that follow the page as it scrolls.

```css
nav {
    position: fixed;
    left: 0;
    top: 0;
}
```

Hard to describe, see a [live demo](#)

# FIXED

This popup background uses `position: fixed` to grey-out the entire page, even if the user scrolls.

`position: absolute` is a powerful tool that allows you to place any page element exactly where you want it, down to the pixel.
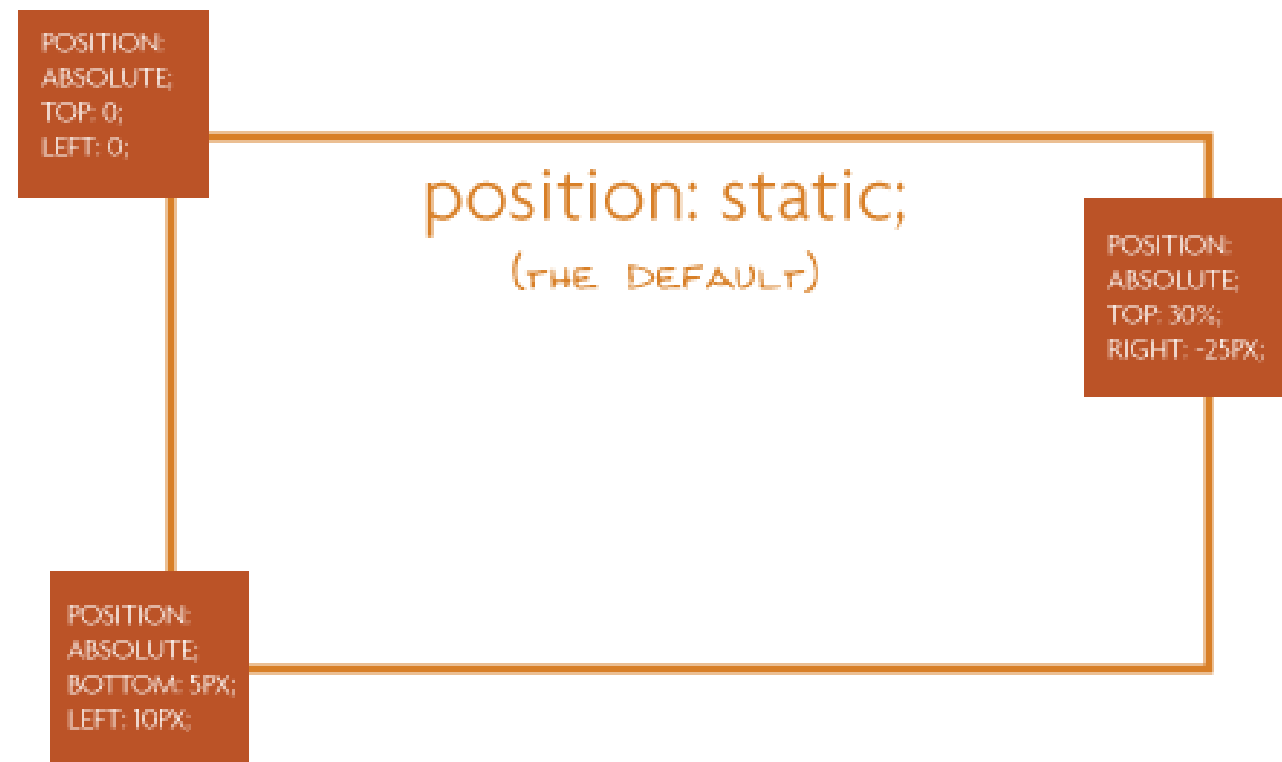
When an element has `position: absolute`, it is entirely removed from the normal flow of the page.

- That means its padding, margins, and borders no longer affect the elements around it

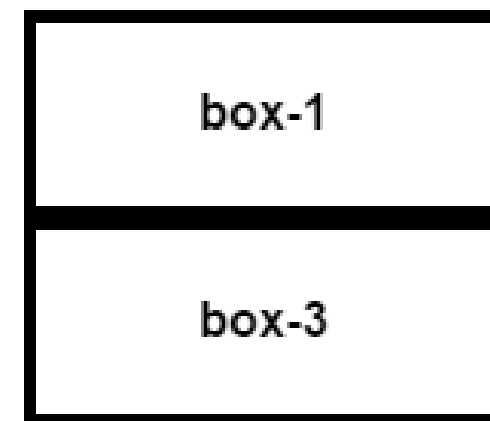An element with `position: absolute` is absolutely positioned:

- To its closest parent with positioning

- Or (if it has no parents with positioning other than `static`), to the `body` of the page

# ABSOLUTE

box-2 has the following style, which moves it to the bottom right of the page:

```
#box-2 {
    position: absolute;
    right: 10px;
    bottom: 10px;
}
```
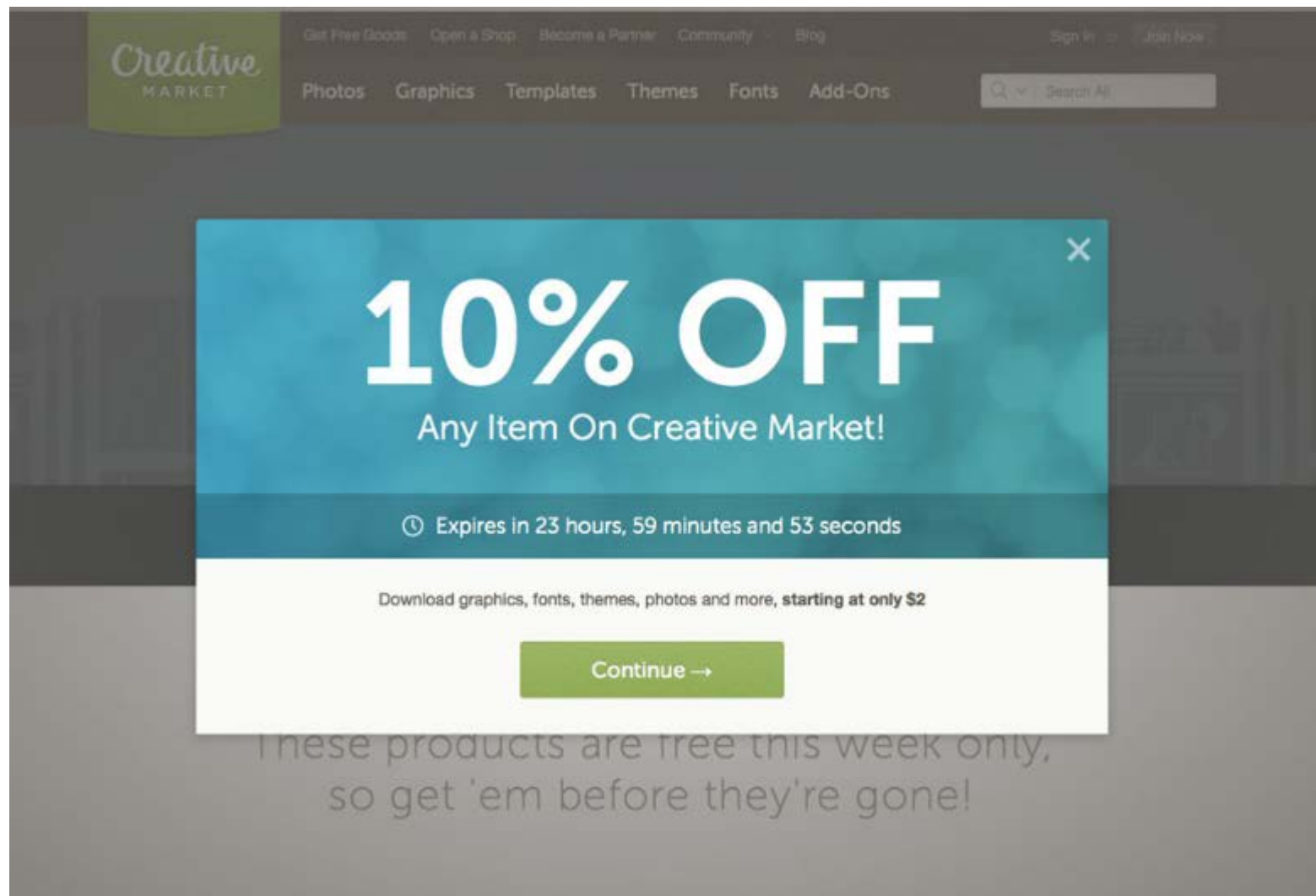
box-3 moves up to occupy the space vacated by box-2

# ABSOLUTE POWER

`position: absolute` is commonly used when creating page modals that pop up over other content
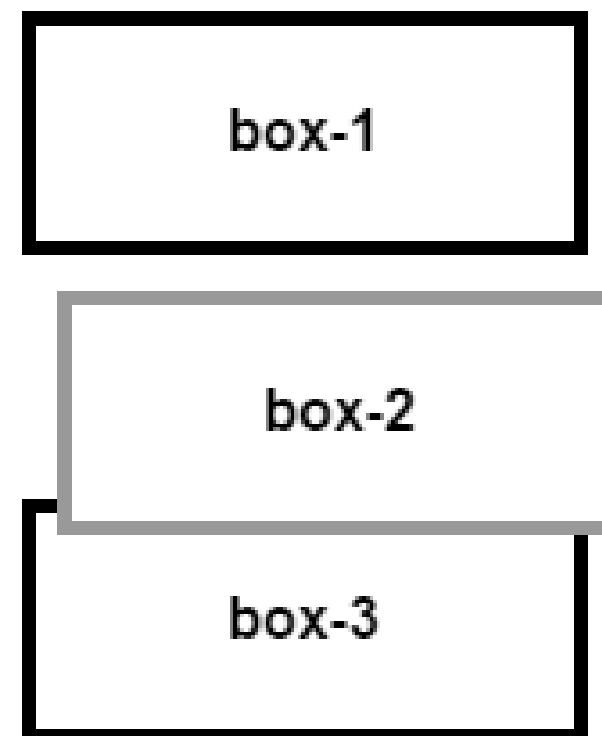
`position: relative` allows you to move an element using directional attributes (`top`, `bottom`, etc).

- If you set `position: relative` but no directional attributes, the element won't change at all

- If you set `top: 10px;` then it will be shifted 10 pixels down from where it would normally be

# RELATIVE

box-2 has the following style, which moves it down and right:

```
#box-2 {
    position: relative;
    left: 10px;
    top: 10px;
}
```



The other boxes behave like box-2 was in its original position (unlike using negative margins).

More commonly, `position: relative` is used to position other absolutely-positioned elements inside the container

PRACTICE TIME!

# ASSIGNMENT

Create a new page using this page as a template:

https://alistapart.com/d/css-positioning-101/example_a.html

- Give at least one of the boxes relative positioning and apply `left`, `top`, etc.

- Give at least one of the boxes absolute positioning.

  - Try putting a `section` around the boxes and give that `relative` positioning – what happens as you resize the browser?

# TABLES

# WHY TABLES?

Use tables to present data in a tabular format.

- Listings of people, addresses, etc

- Financial data

- Sports stats

- Product features

| Month | Sales |
|-------|-------|
| January | $ 50,000.00 |
| February | $ 10,000.00 |
| March | $ 85,000.00 |
| April | $ 56,000.00 |
| May | $ 98,000.00 |

`<table>` wraps the whole table

`<thead>` wraps the table header

`<tbody>` wraps the main data

`<tr>` wraps a **t**able **r**ow

`<th>` wraps a **t**able **h**eader cell

`<td>` wraps a regular **t**able **d**ata cell

# A BASIC TABLE

```html
<table>
    <thead>
        <tr>
            <th>Header 1</th>
            <th>Header 2</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>Data 1a</td>
            <td>Data 2a</td>
        </tr>
        <tr>
            <td>Data 1b</td>
            <td>Data 2b</td>
        </tr>
    </tbody>
</table>
```
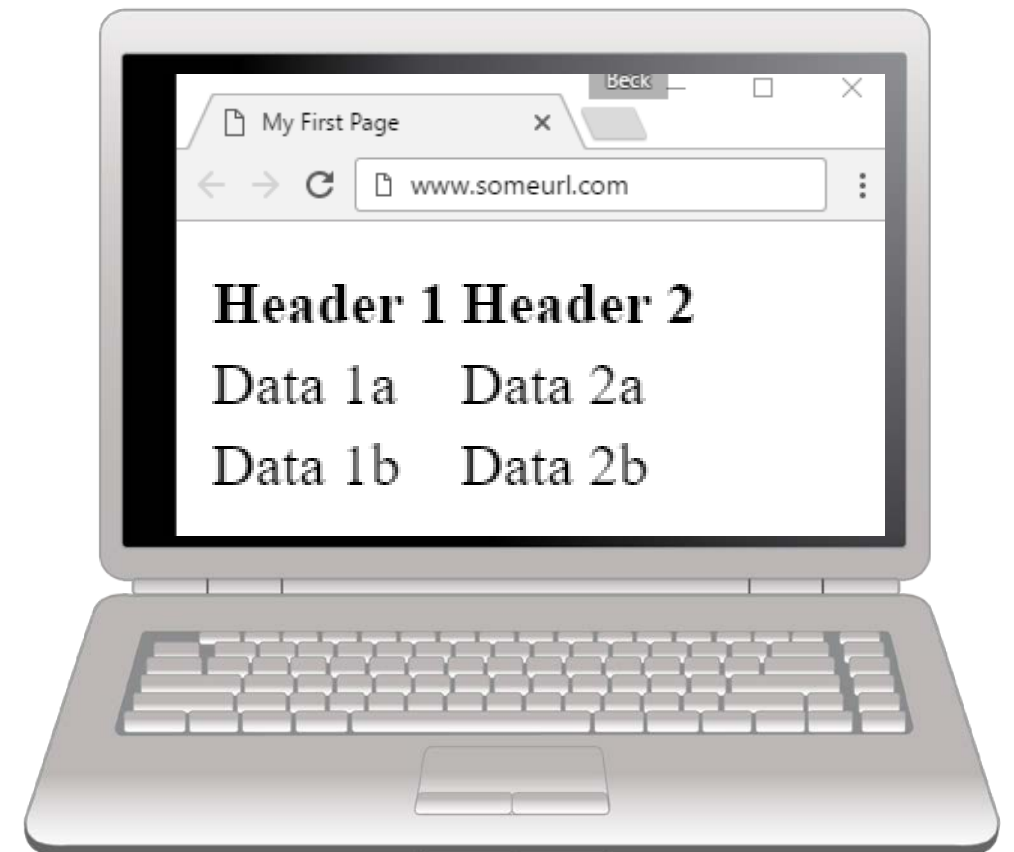
My First Page     www.someurl.com

**Header 1 Header 2**

Data 1a    Data 2a

Data 1b    Data 2b

`<tr></tr>` creates a table row.

- Every row is a new line

- Header rows and body rows both use `<tr>`

- Make sure the same number of cells appear in every row (or things will look broken)

`<th></th>` creates a **heading** table cell.

- Put inside a row (`<tr></tr>`) inside a `<thead>` element

- Displays as bold by default

- Screen readers will read the heading of a column when reading the corresponding item in a row

`<td></td>` creates a **regular** table cell.

- Put inside a row (`<tr></tr>`) inside the `<tbody>` element

- Table cells can hold any kind of content (images, links, lists, etc)

- Table cells always expand to fit their content, both horizontally and vertically

- Browsers will choose a table cell width that best fits the content (but you can change this in CSS)

Tables can be styled using the CSS properties we've already discussed.

| Employee | Salary | Bonus | Supervisor |
| --- | --- | --- | --- |
| Stephen C. Cox | $300 | $50 | Bob |
| Josephin Tan | $150 | – | Annie |
| Joyce Ming | $200 | $35 | Andy |
| James A. Pentel | $175 | $25 | Annie |

| FILM | YEAR |
| --- | --- |
| **Citizen Kane**<br>Orson Welles, Joseph Cotten, Dorothy Comingore | 1941 |
| **Gone With The Wind**<br>Clark Gable, Vivian Leigh | 1939 |
| **Casablanca**<br>Humphrey Bogart, Ingrid Bergman | 1942 |

# STYLING TABLES

If you try to add borders to table cells you may notice something odd:

```
td, th { border: 1px solid #cccccc; }
```

| One | Two | Three |
|-----|-----|-------|
| Apples | Carrots | Steak |
| Oranges | Potato | Pork |
| Pears | Peas | Chicken |

- This is because adjacent table cells each have their own distinct borders that do not merge together
- You can get rid of this behavior by setting this CSS style:

```
td, th { border-collapse: collapse; }
```

# STYLING TABLES

You can create zebra-striped tables by using the pseudo-selector `:odd` or `:even`

```
tr:odd td { background-color: #ccc; }
```

That applies a light grey background to only table data contained in odd rows.

| Starter | Professional | Business | Unlimited |
|---|---|---|---|
| $29 p/m | $59 p/m | $99 p/m | $149 p/m |
| **Unlimited** bandwidth | **Unlimited** bandwidth | **Unlimited** bandwidth | **Unlimited** bandwidth |
| **Free** setup | **Free** setup | **Free** setup | **Free** setup |
| 3 tracks | 10 tracks | 100 tracks | **Unlimited tracks** |
| 5% transaction fee | 3% transaction fee | 2% transaction fee | 1% transaction fee |
| 100mb storage | 500mb storage | 1000mb storage | **Unlimited** storage |
| SIGN UP NOW | SIGN UP NOW | SIGN UP NOW | SIGN UP NOW |

PRACTICE TIME!

# ASSIGNMENT

Create a new page and make a table with at least 4 rows and 2 columns.

- Style the table header and table body differently.

- Apply some CSS to make your table look nice:
  - box model properties (padding, margin, border)
  - font properties (size, color, font-family)
  - container properties like background-color

# FORMS

**Forms** are used to collect information from a user.

- The element `<form>` itself doesn't visually appear on the page – it's just a wrapper for form content

# INPUT

The `input` element allows users to enter data in various ways.

- Looks different depending on its `type`
- Displays inline by default

`<input type="text" />`

`<input type="radio" />`

`<input type="checkbox" />`

# INPUT

The `input` element **requires** an attribute called `name` to identify what data is being provided.

- Just like classes or ids, you pick what to call it – so be descriptive!

```
<input type="text" name="emailAddress" />
```

# INPUT

A `placeholder` attribute in a text input creates a "watermarked" label that is inside the box.

- Often used to show formatting suggestions

- By default, the browser displays as a light grey

- But, like anything else, it can be styled with CSS using the `:placeholder` pseudo-class

```
<input type="text" name="phoneNumber"
       placeholder="(999) 999-9999" />
```

(999) 999-9999

# LABELS

 A  label describes what should go inside the input on a form.

```
<label for="address">Address:</label>
<input type="text" name="address" />
```

You tell the label what it's `for` by giving it the `name` of the input field it's related to.

- `label` is also an inline element, so by default it appears on the same line as an `input`

Address: [                    ]

# DROPDOWN

A select element creates a dropdown box.

Cromwell ▼

```
<select name="cities">
     <option value="sydney">Sydney</option>
     <option value="melbourne">Melbourne</option>
     <option value="cromwell" selected="selected">Cromwell</option>
     <option value="queenstown">Queenstown</option>
</select>
```

# DROPDOWN

```html
<select name="cities">
    <option value="sydney">Sydney</option>
    <option value="melbourne">Melbourne</option>
    <option value="cromwell" selected="selected">Cromwell</option>
    <option value="queenstown">Queenstown</option>
</select>
```

Each `option` in a dropdown is a child of a `select` element.

- To choose an option when the page loads, apply the attribute `selected="selected"`

# SUBMIT

Finally, a submit button at the bottom of the form.

```
<button type="submit">Click me!</button>
```

Click me!

When a button has `type="submit"`, hitting enter will automatically submit the form.

PRACTICE TIME!

# ASSIGNMENT

Create a new page that contains a form.

- Include at least 2 different types of `input`, at least one `label`, and a submit button.

- Style your form.

- What happens when you click the submit button?

# JAVASCRIPT/JQUERY

**Javascript** is a scripting language that developers use to manipulate the Document Object Model.

- jQuery is a Javascript library for easily creating animations and interactions with very little code
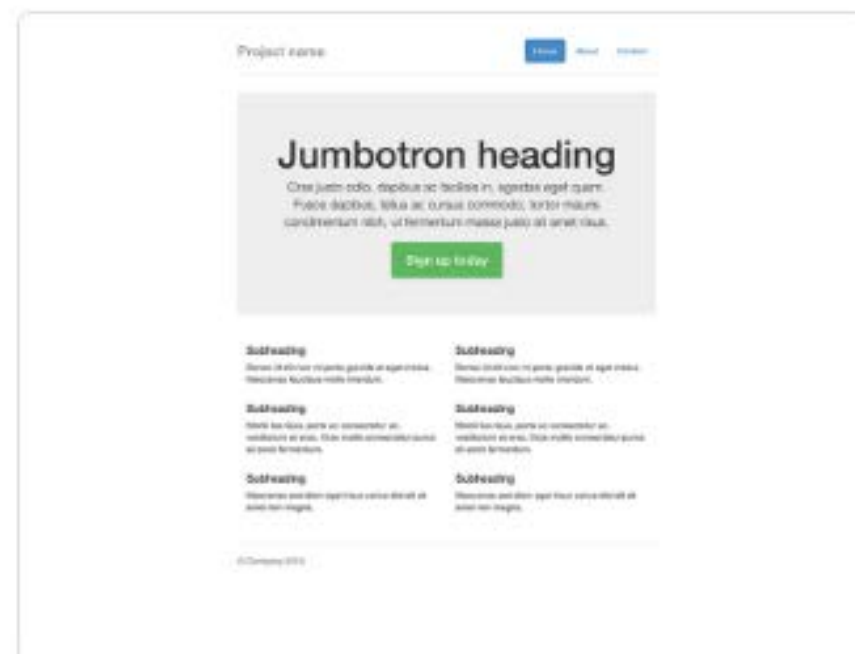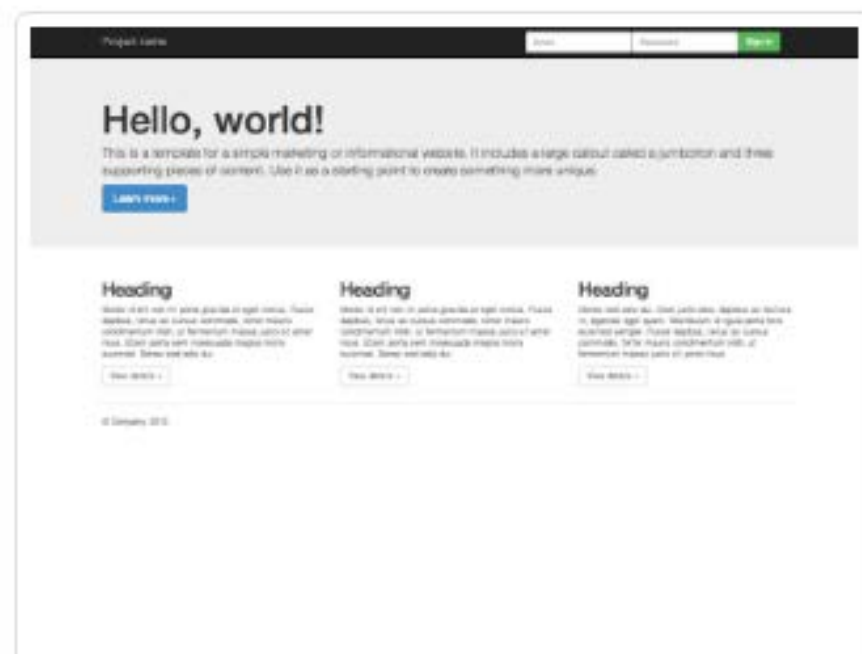


- Find elements on the page using CSS syntax

# BOOTSTRAP

**Bootstrap** is a CSS library you can import into a project to quickly provide some base styles for your project.

http://getbootstrap.com/getting-started/

# THATS ALL FOLKS!

Please provide feedback for this class!

http://svcseattle.com/evaluation