INTRODUCTION TO HTML & CSS

Instructor: Beck Johnson

Week 4

SESSION OVERVIEW

- Review borders, inline/block, and class/id
- Layout – floats and flex
- Using web fonts
- Responsive design

REVIEW!

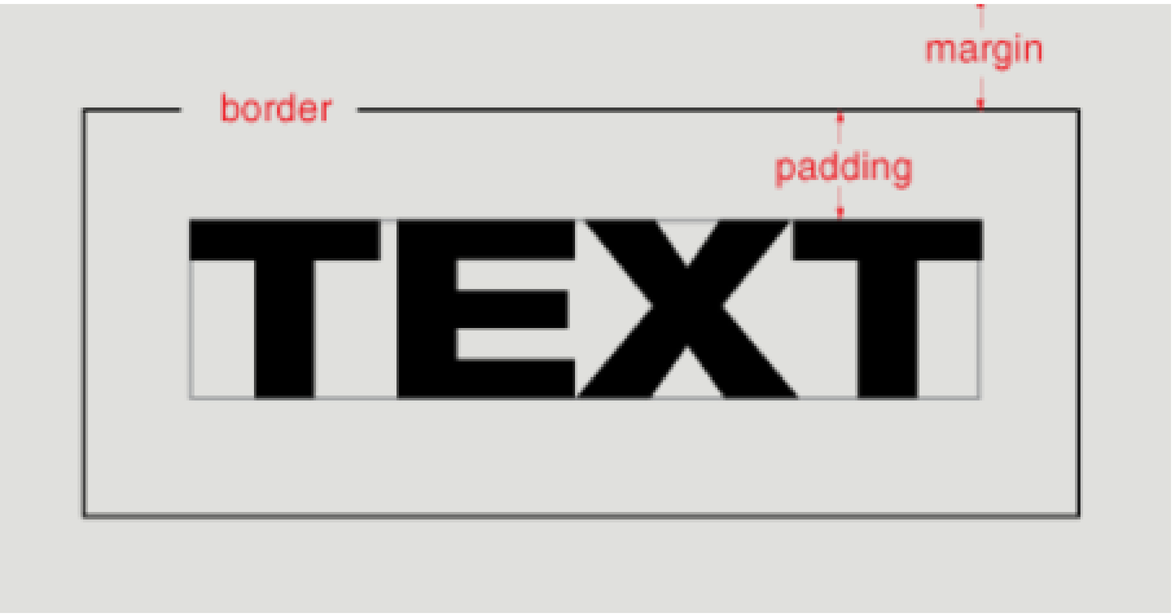
{ } CSS BOX MODEL

CONTENT: stuff in the box

PADDING: space inside the box

BORDER: sides of the box

MARGIN: space between multiple boxes



{ } BORDER

Between margin and padding, you can set a border

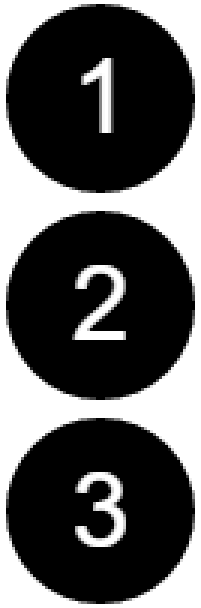
- Width (usually in pixels)
- Border style (solid, dotted, dashed, etc)
- Color

```
p
    border: 2px dotted #ff0000;
}
```

{ } BORDER-RADIUS

To make an element appear curved, use the property `border-radius`

- The value is a number (in px or em) or percentage
- You can use `border-radius` even if you don’t explicitly set a `border`

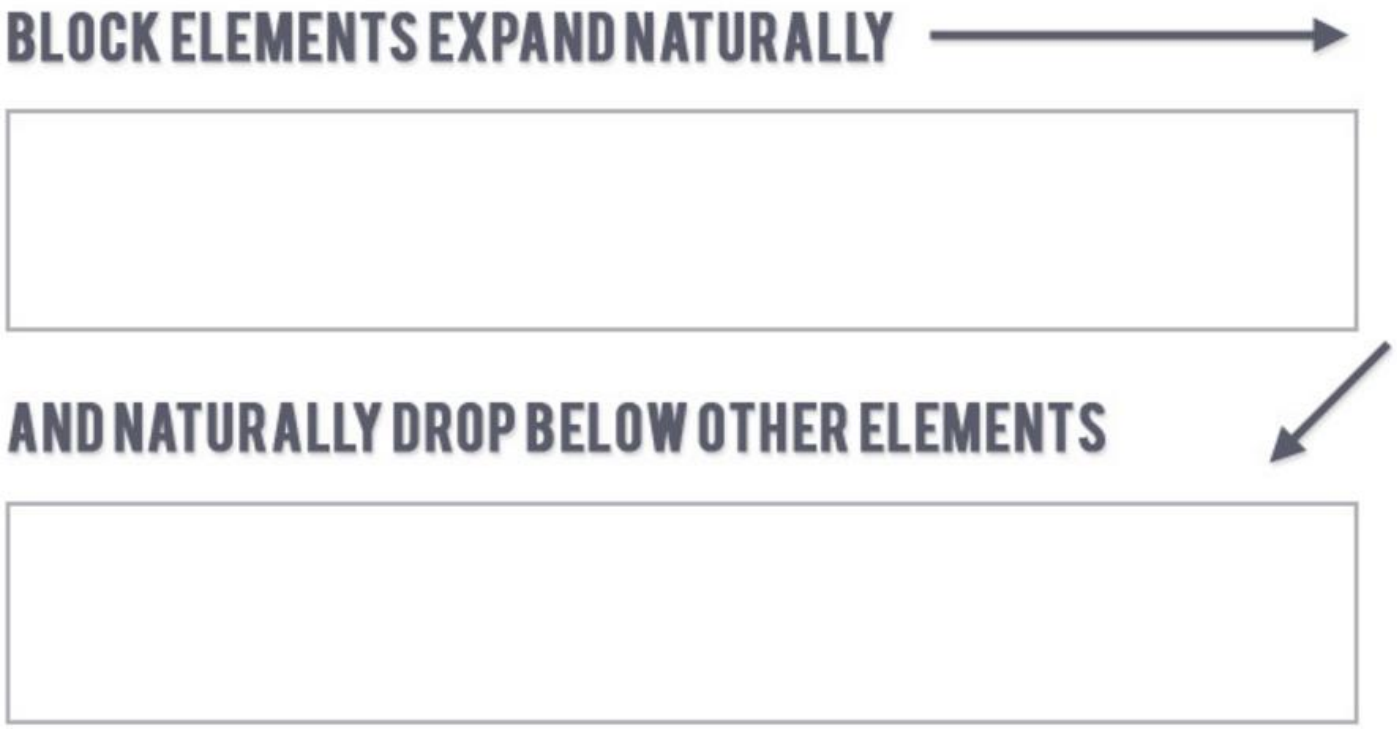


```
li {
      border-radius: 50%;
      height: 3em;
      width: 3em;
}
```

{ } BLOCK ELEMENTS

BLOCK ELEMENTS

- Expand naturally to fill their parent container
- Take up a full line
- Can have margin and/or padding



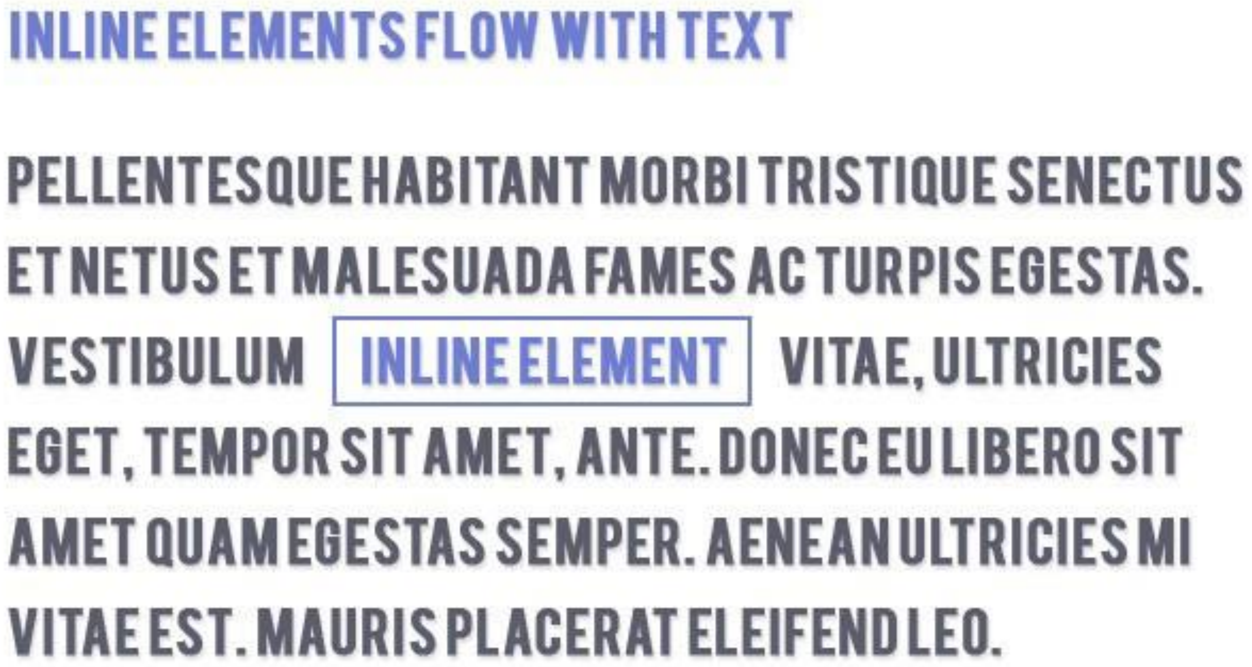
BLOCK ELEMENTS EXPAND NATURALLY

AND NATURALLY DROP BELOW OTHER ELEMENTS

{ } INLINE ELEMENTS

INLINE ELEMENTS

- Flow along with text content
- Ignores top and bottom margin/padding
- Honors left and right margin/padding



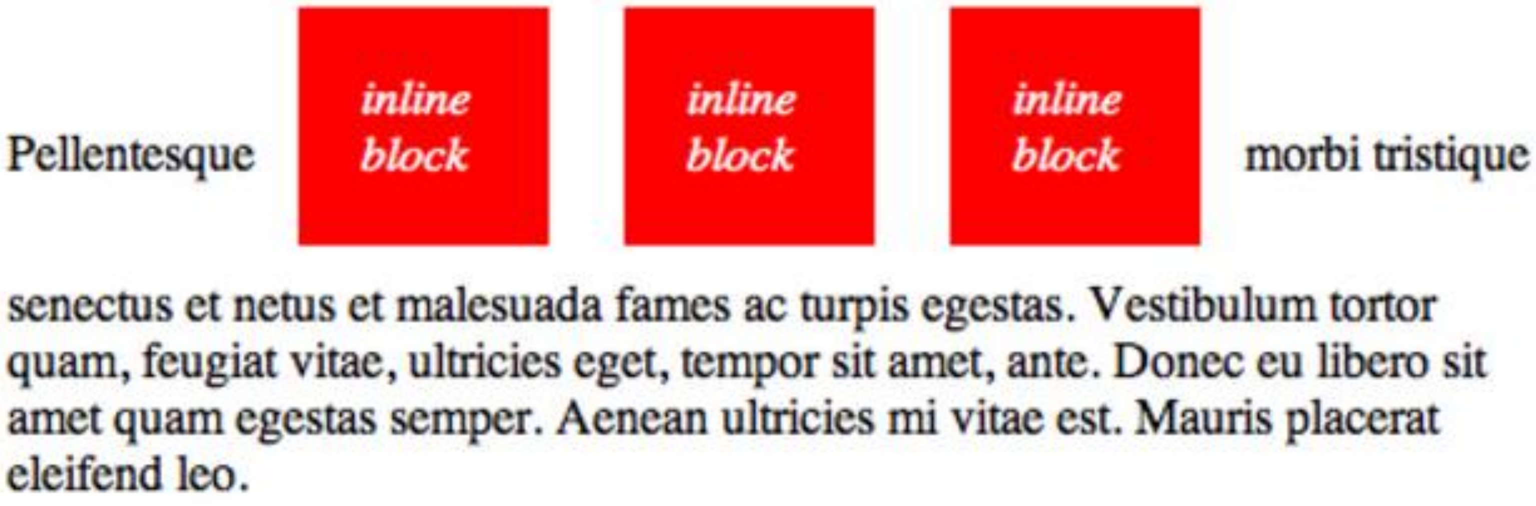
INLINE ELEMENTS FLOW WITH TEXT

PELLENTESQUE HABITANT MORBI TRISTIQUE SENECTUS ET NETUS ET MALESUADA FAMES AC TURPIS EGESTAS. VESTIBULUM INLINE ELEMENT VITAE, ULTRICIES EGET, TEMPOR SIT AMET, ANTE. DONEC EU LIBERO SIT AMET QUAM EGESTAS SEMPER. AENEAN ULTRICIES MI VITAE EST. MAURIS PLACERAT ELEIFEND LEO.

{ } INLINE-BLOCK ELEMENTS

INLINE-BLOCK ELEMENTS

- A hybrid of block and inline
- Flow along with text content
- Has height, width, margin, and padding



Pellentesque *inline block* *inline block* *inline block* morbi tristique

senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo.

{} 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 id="anyLettersOrNumbersOr_Or-">Still totally
valid</p>
```

{} CLASSES

Multiple elements can have the same `class`

```
<p class="muscles">Schwarzenegger</p>
<p class="muscles">The Rock</p>
```

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

```
.muscles { display: flex; }
```

{} IDS

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

```
<div id="tower-of-pisa">There can be only
one</div>
```

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

```
#tower-of-pisa { font-style: italic; }
```

An `id` on an element lets you link to it directly using a link `href` starting with a `#`

{} CLASS VS ID

id

"Lucy"

class

"dog"

"gray"

"small"

"charming"

"seated"

QUESTIONS?

WEB LAYOUTS

WEB LAYOUTS

With CSS, we can use a variety of properties to arrange elements on the screen by adjusting the flow of the page.

Basically, you can put elements anywhere…which can be both a good and a bad thing!

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

DISPLAY PROPERTY

The `display` property tells the browser what type of box model to use:

- `inline`
- `inline-block`
- `block`
- `flex`

This changes how padding, margin, height and width affect an element.

You also can set `display: none` to hide an element entirely.

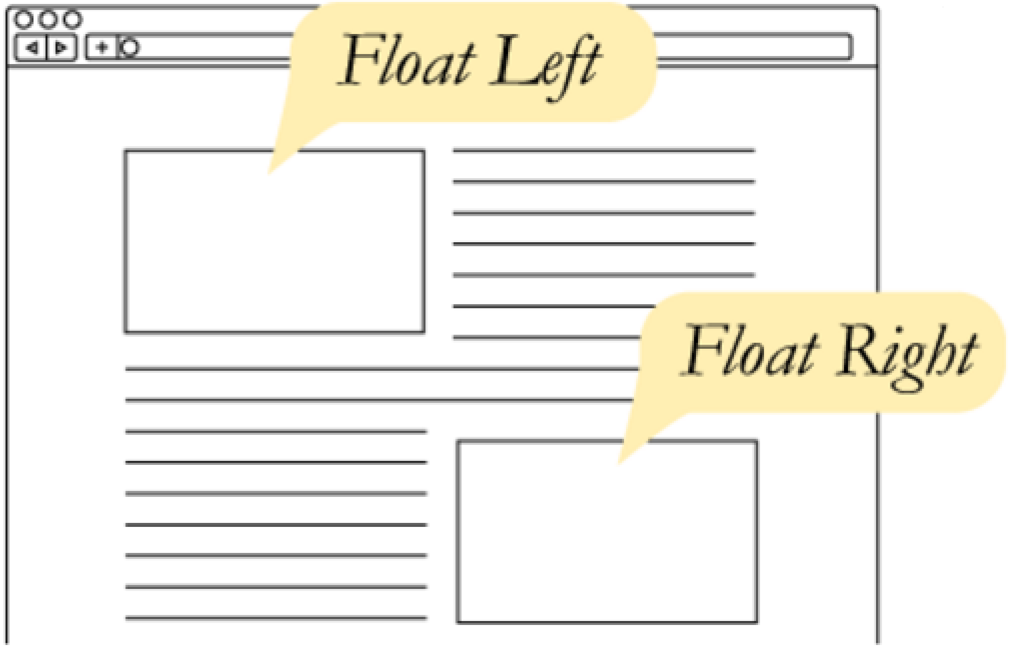
CSS FLOATS

Up to now, elements have displayed sequentially, in the order that you placed them in your HTML.

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; }
```

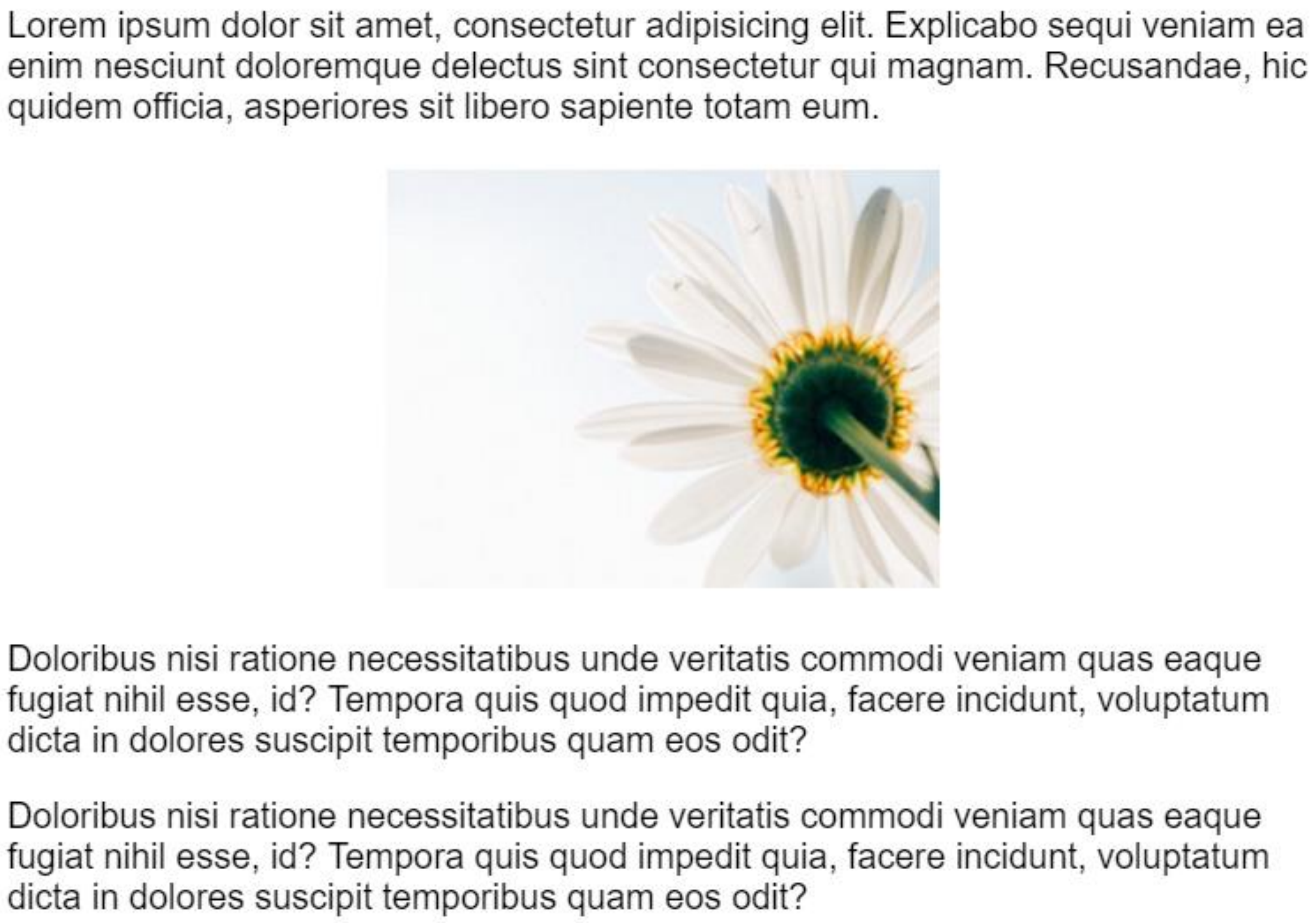


CSS FLOATS

The three values for `float` are:

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

By default, elements are `float: none`



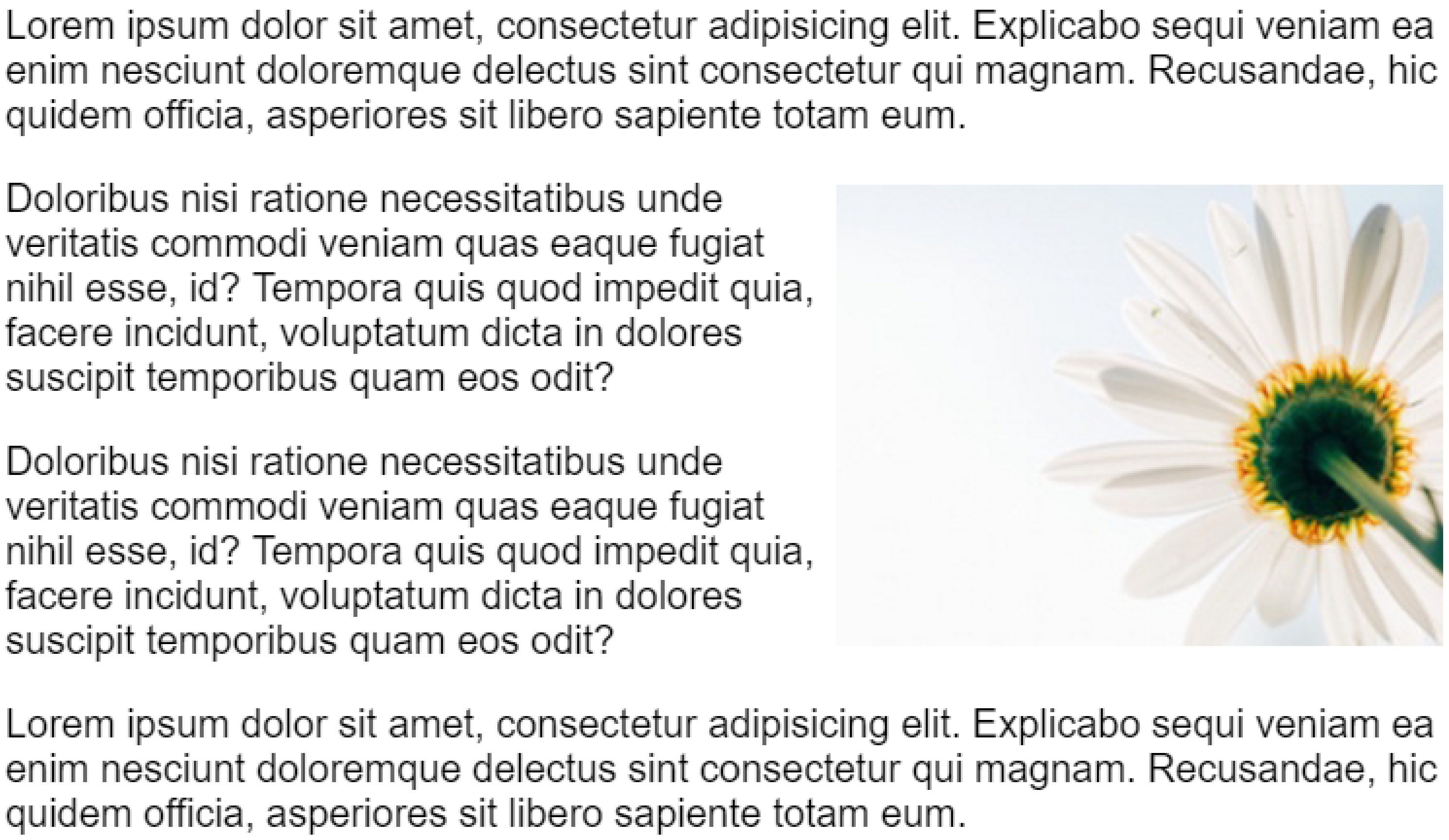
Lorem ipsum dolor sit amet, consectetur adipisicing elit. Explicabo sequi veniam ea enim nesciunt doloremque delectus sint consectetur qui magnam. Recusandae, hic quidem officia, asperiores sit libero sapiente totam eum.

Doloribus nisi ratione necessitatibus unde veritatis commodi veniam quas eaque fugiat nihil esse, id? Tempora quis quod impedit quia, facere incidunt, voluptatum dicta in dolores suscipit temporibus quam eos odit?

Doloribus nisi ratione necessitatibus unde veritatis commodi veniam quas eaque fugiat nihil esse, id? Tempora quis quod impedit quia, facere incidunt, voluptatum dicta in dolores suscipit temporibus quam eos odit?

CSS FLOATS

```
img { float: right; }
```



Lorem ipsum dolor sit amet, consectetur adipisicing elit. Explicabo sequi veniam ea enim nesciunt doloremque delectus sint consectetur qui magnam. Recusandae, hic quidem officia, asperiores sit libero sapiente totam eum.

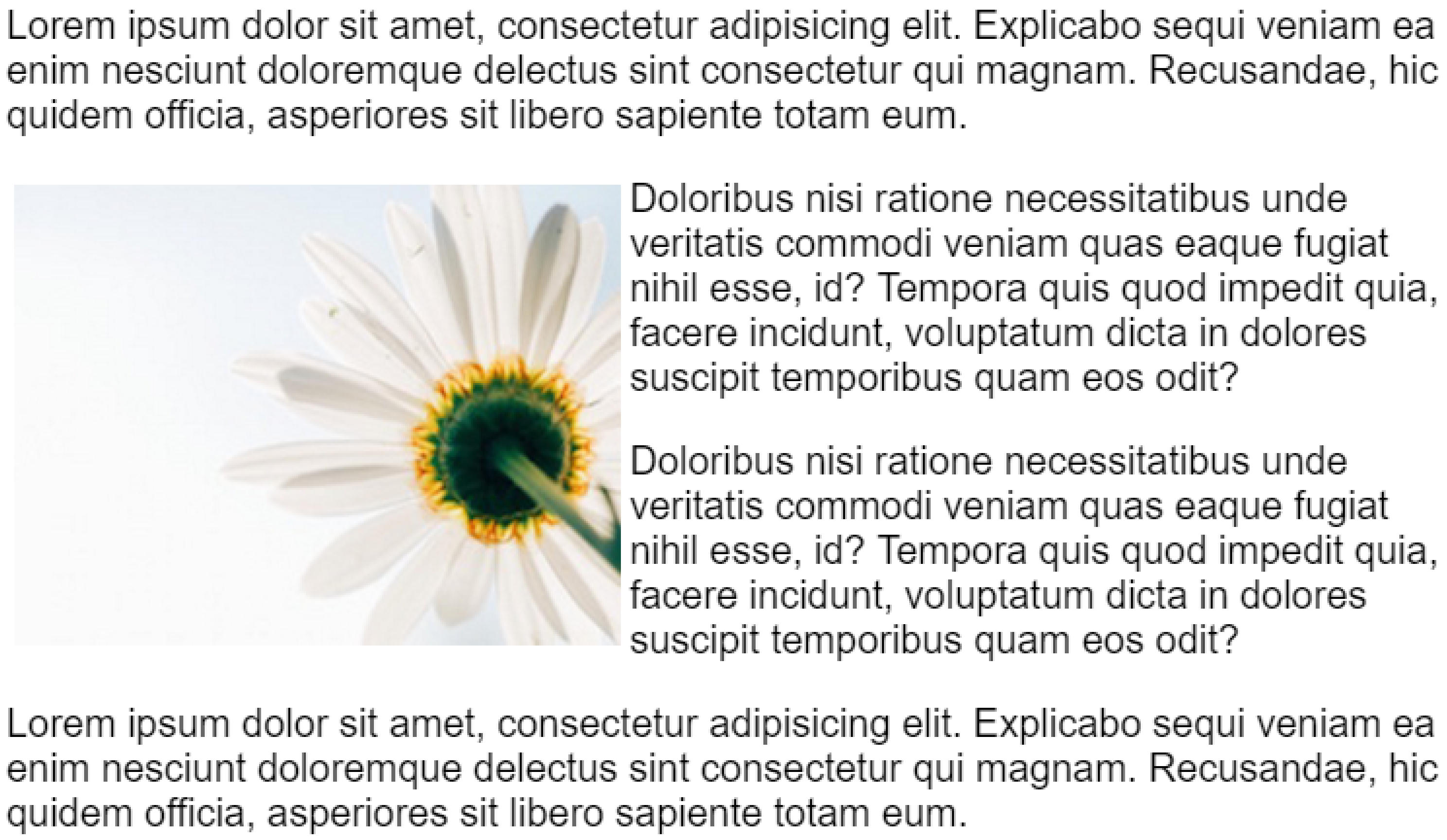
Doloribus nisi ratione necessitatibus unde veritatis commodi veniam quas eaque fugiat nihil esse, id? Tempora quis quod impedit quia, facere incidunt, voluptatum dicta in dolores suscipit temporibus quam eos odit?

Doloribus nisi ratione necessitatibus unde veritatis commodi veniam quas eaque fugiat nihil esse, id? Tempora quis quod impedit quia, facere incidunt, voluptatum dicta in dolores suscipit temporibus quam eos odit?

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Explicabo sequi veniam ea enim nesciunt doloremque delectus sint consectetur qui magnam. Recusandae, hic quidem officia, asperiores sit libero sapiente totam eum.

CSS FLOATS

```
img { float: left; }
```



Lorem ipsum dolor sit amet, consectetur adipisicing elit. Explicabo sequi veniam ea enim nesciunt doloremque delectus sint consectetur qui magnam. Recusandae, hic quidem officia, asperiores sit libero sapiente totam eum.

Doloribus nisi ratione necessitatibus unde veritatis commodi veniam quas eaque fugiat nihil esse, id? Tempora quis quod impedit quia, facere incidunt, voluptatum dicta in dolores suscipit temporibus quam eos odit?

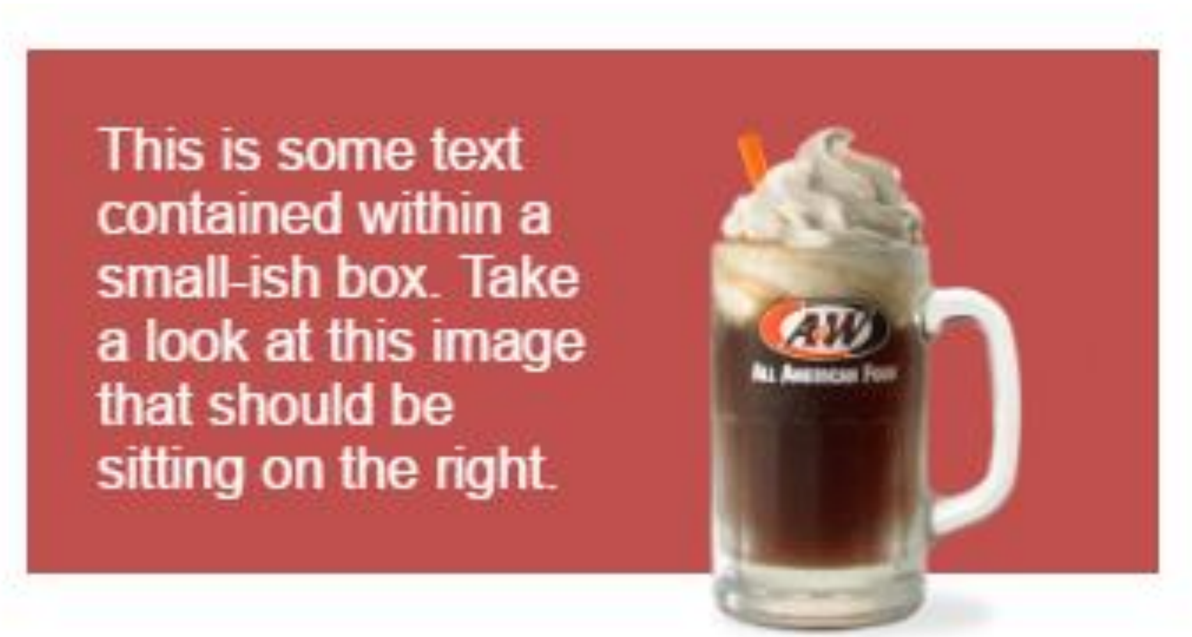
Doloribus nisi ratione necessitatibus unde veritatis commodi veniam quas eaque fugiat nihil esse, id? Tempora quis quod impedit quia, facere incidunt, voluptatum dicta in dolores suscipit temporibus quam eos odit?

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Explicabo sequi veniam ea enim nesciunt doloremque delectus sint consectetur qui magnam. Recusandae, hic quidem officia, asperiores sit libero sapiente totam eum.

FLOWS BEFORE ROWS

Where you place your floated element in your HTML can cause different results.

Live demo here.



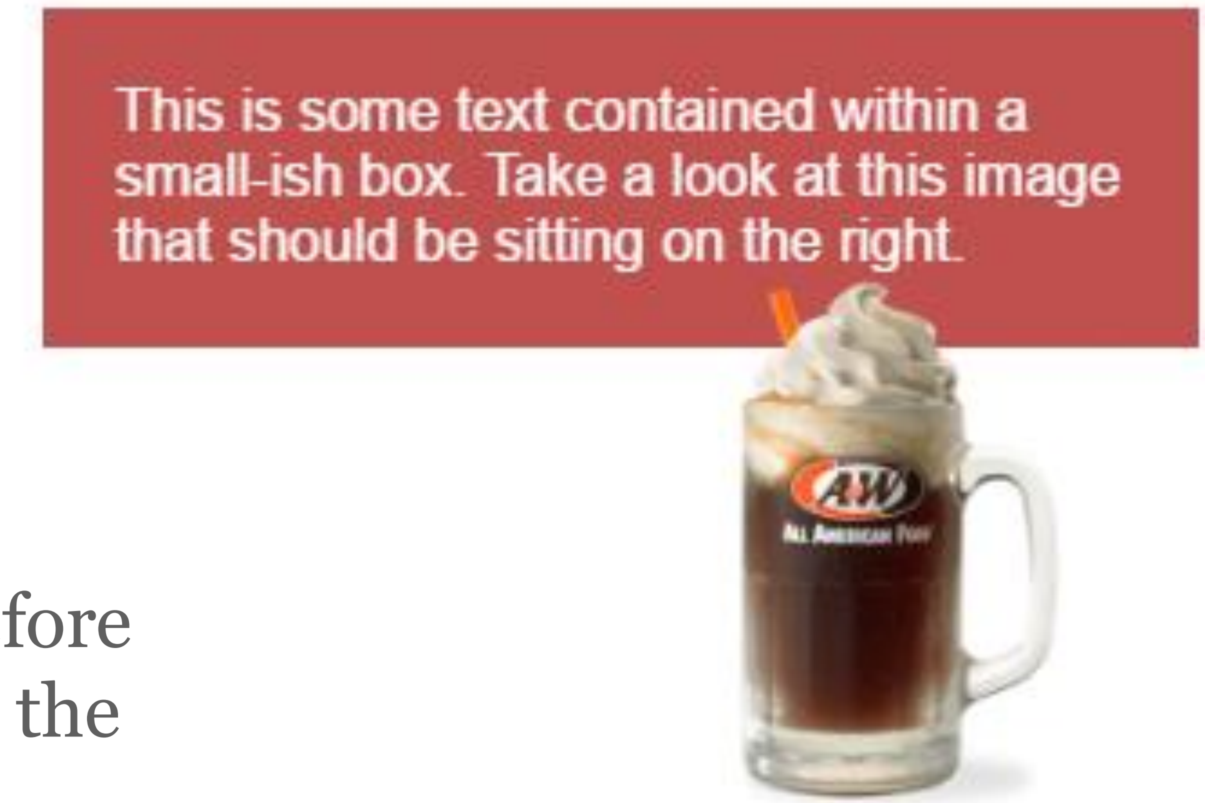
This is some text contained within a small-ish box. Take a look at this image that should be sitting on the right.

```
<div id="container">
   <img src="rootbeer-float.png" style="float:right" />
   <p>This is some text contained…</p>
</div>
```

FLOWS BEFORE ROWS

What happens if we move the image below the paragraph?

```
<div id="container">
   <p>This is some text contained…</p>
   <img src="rootbeer-float.png" style="float: right" />
</div>
```



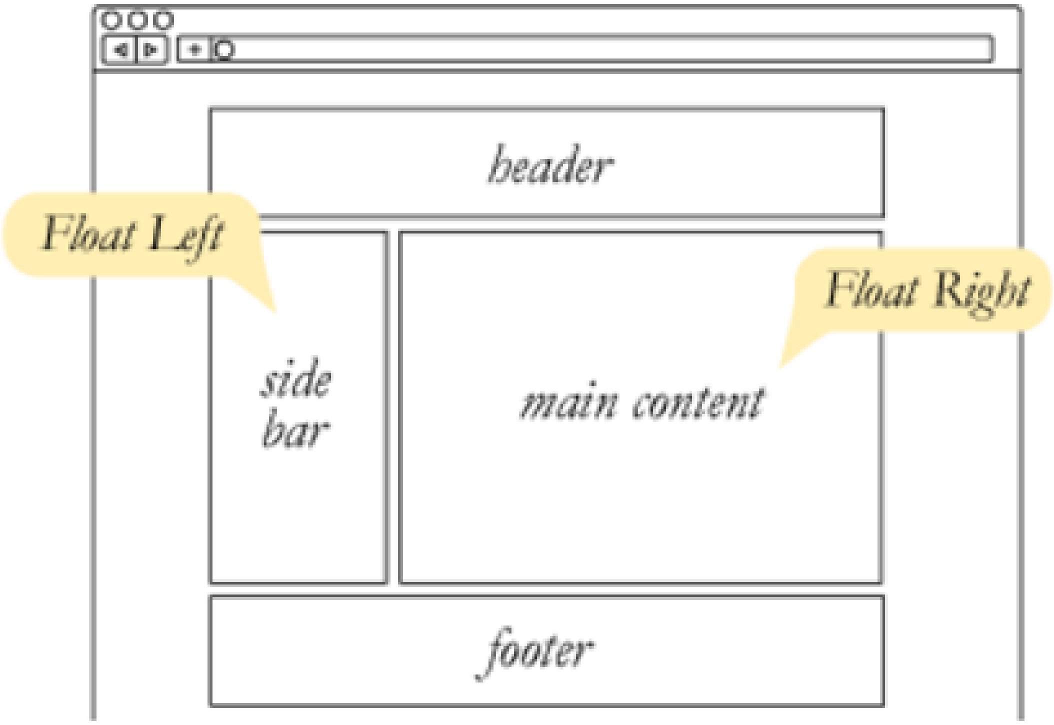
This is some text contained within a small-ish box. Take a look at this image that should be sitting on the right.

A general rule is **float first**.

- Place floated elements before any non-floated elements the float interacts with

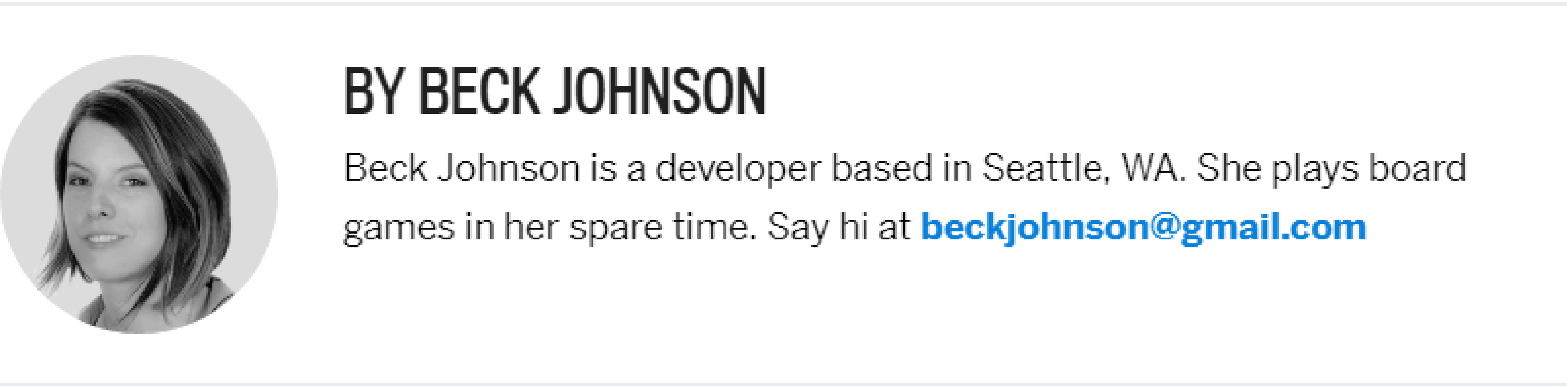
CSS FLOATS

`float` can be used to float text around images, but it also can be used to create entire page layouts.



CSS FLOATS

For example, this layout was built using `float`.

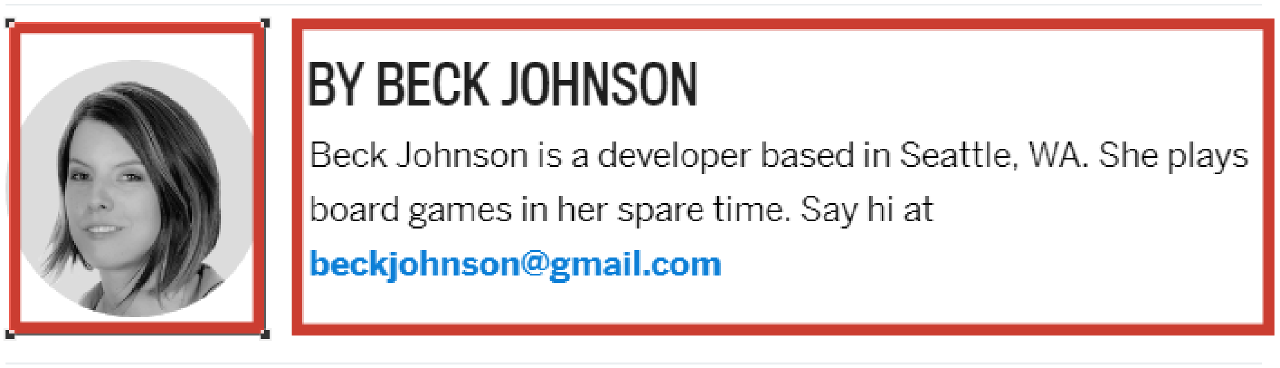


BY BECK JOHNSON

Beck Johnson is a developer based in Seattle, WA. She plays board games in her spare time. Say hi at **beckjohnson@gmail.com**

```html
<div>
   <img class="avatar" src="beck.png" />
   <h2 class="bio">By Beck Johnson</h2>
   <p class="bio">Beck Johnson is… </p>
</div>
```

CSS FLOATS



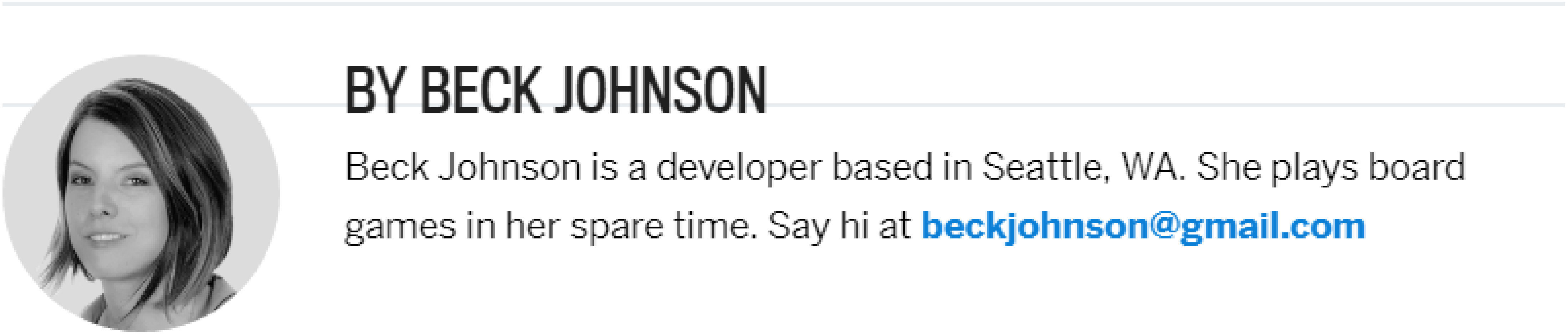
BY BECK JOHNSON

Beck Johnson is a developer based in Seattle, WA. She plays board games in her spare time. Say hi at beckjohnson@gmail.com

```
.avatar { float: left; }        .bio { float: right; }
```

Let’s try that…

CSS FLOATS



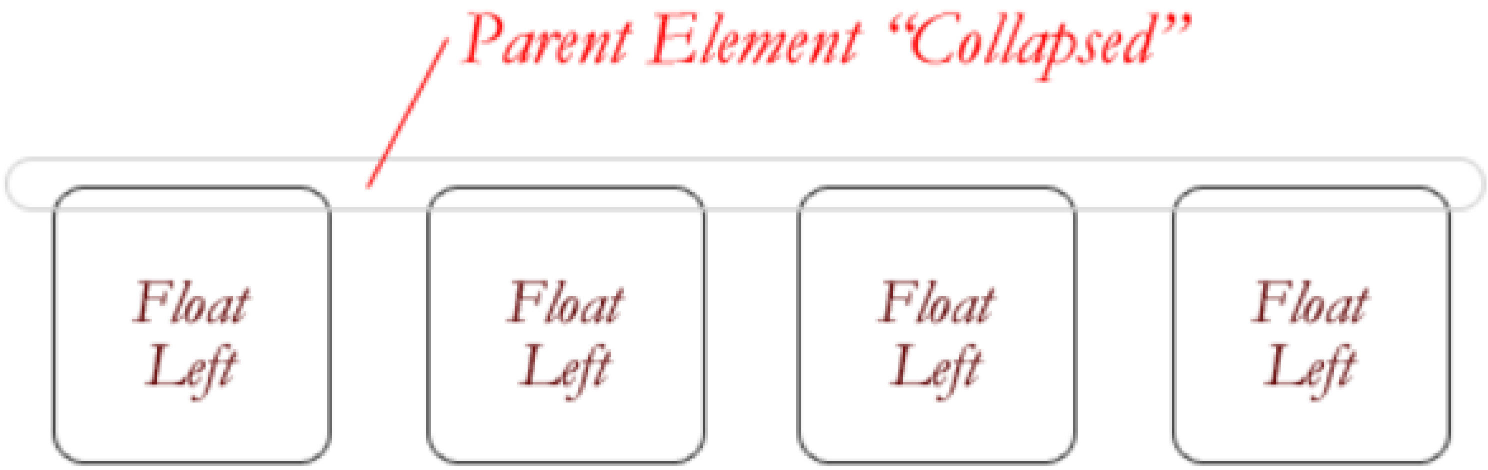
BY BECK JOHNSON

Beck Johnson is a developer based in Seattle, WA. She plays board games in her spare time. Say hi at **beckjohnson@gmail.com**

The container thinks it has no content!

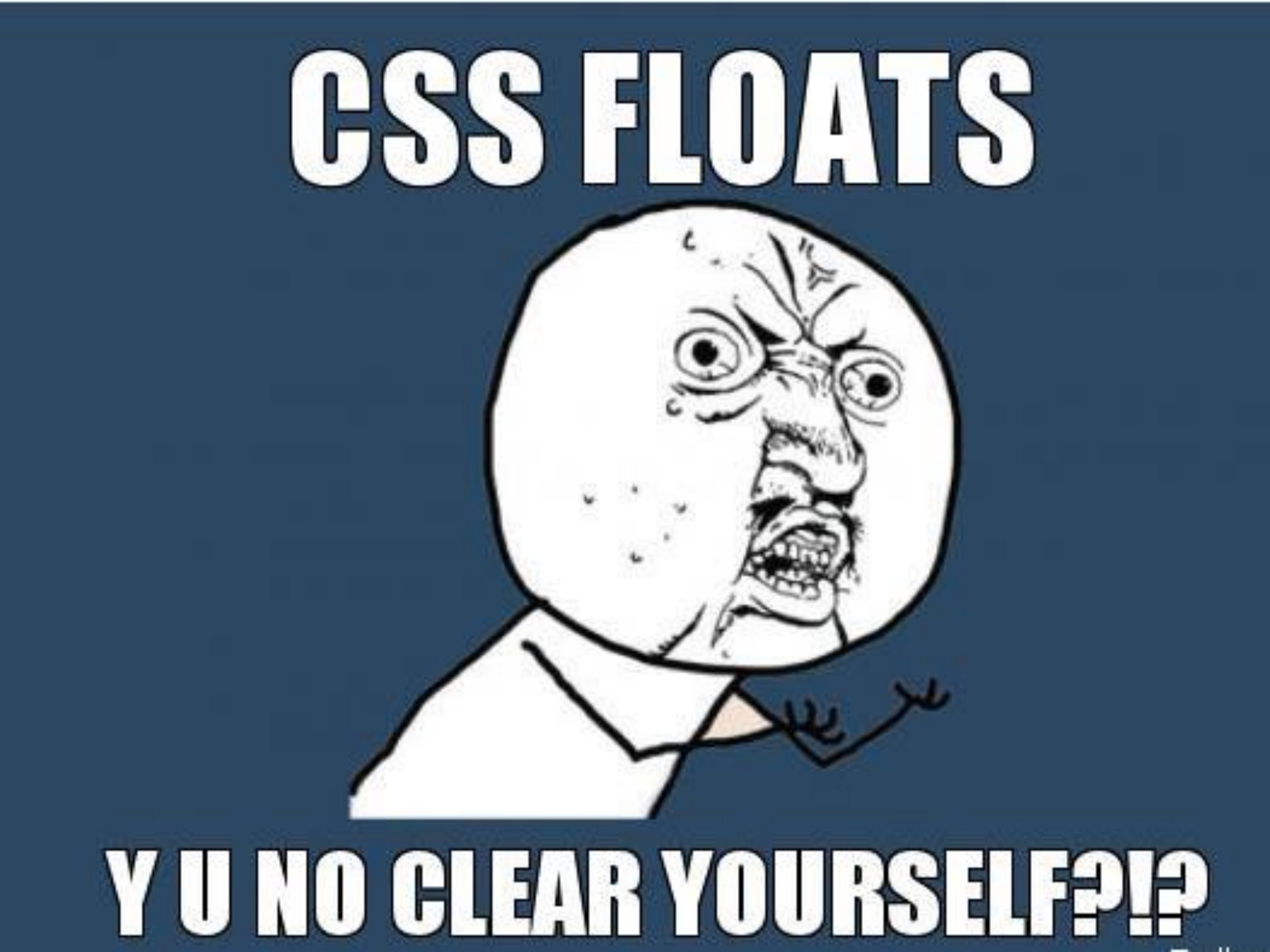
- It collapsed to the size of its padding (you can see the top and bottom `border`)
- The floated content is spilling out

CSS FLOATS



If you float an element, it is no longer in the normal document flow.

If **all** elements in a container are floated, that means that the container is effectively "empty."


CSS FLOATS
Y U NO CLEAR YOURSELF?!?

HOW TO FIX FLOATS?

There are 2 ways to fix this:

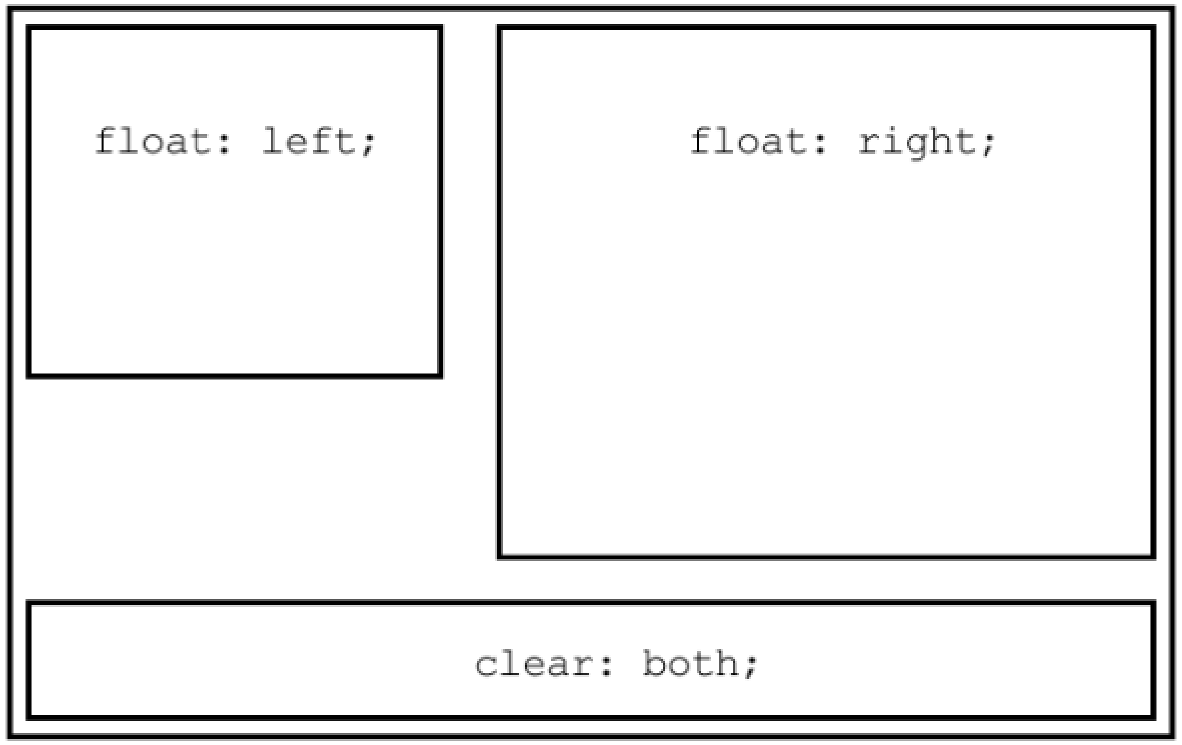
1. Apply the CSS rule `clear: both` to an element **after** the floated content

2. Apply a CSS rule using the property `overflow` to the **container**

THE CLEAR PROPERTY

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

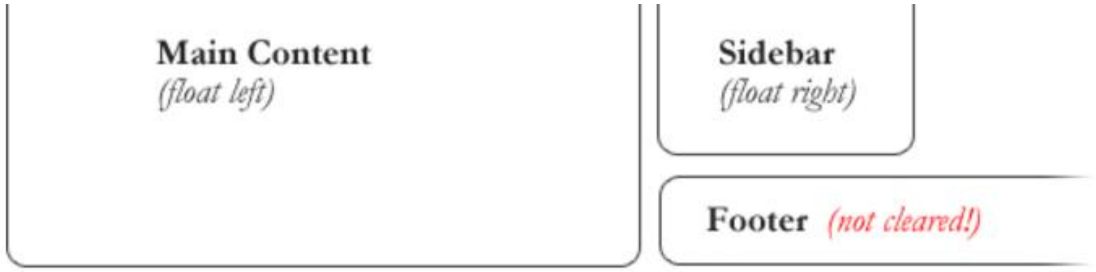
- It doesn't do much until there are floated elements on the page
- An element with `clear` applied to it will force itself **below** the floated element
- Everything after that will be back in the normal flow
- This "stretches" out the container and keeps it from collapsing



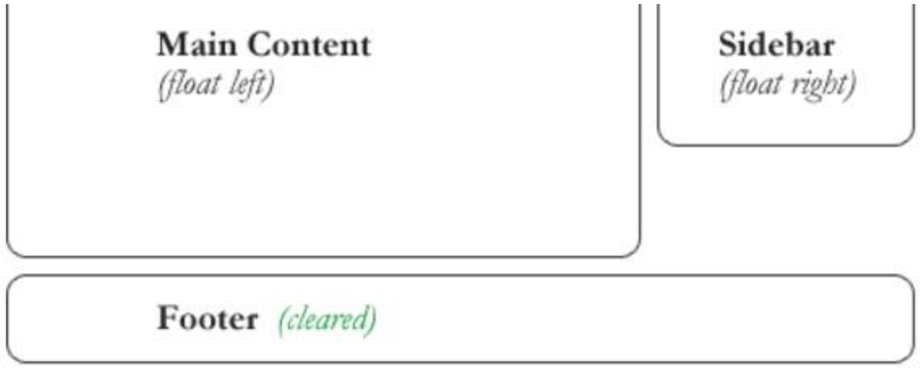
THE CLEAR PROPERTY

`clear` has similar values to `float`:

- `clear: none` – the element does **not** move down to clear past floating elements (this is the default value)



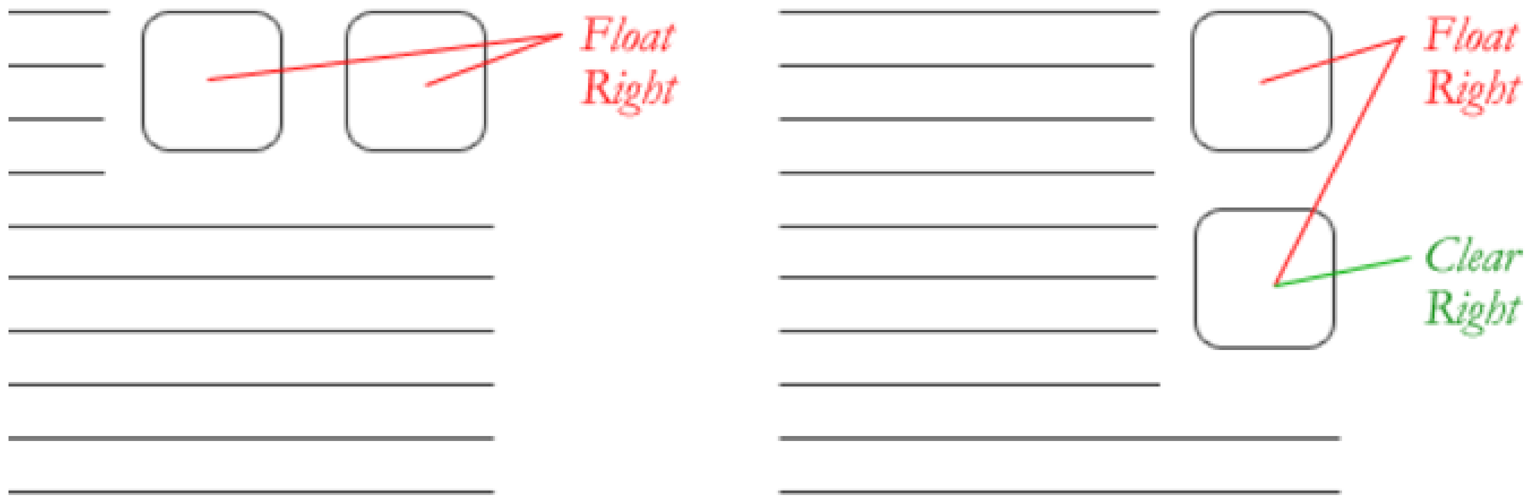
- `clear: both` – the element is moved down to clear **both** right- and left-floated elements



THE CLEAR PROPERTY

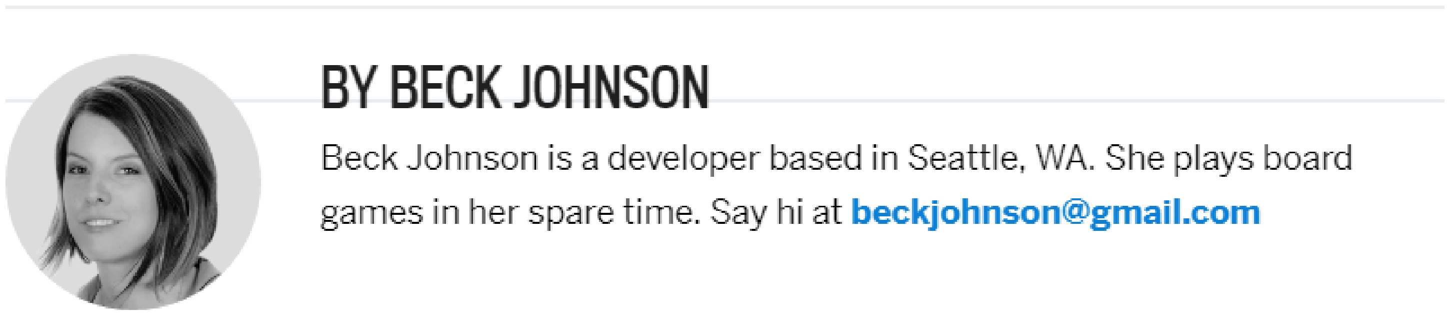
Sometimes, you want to let some content after a `clear` continue floating, but not other content:

- `clear: left` – only clear **left-floated** elements
- `clear: right` – only clear **right-floated** elements



THE CLEAR PROPERTY

So to solve our problem, you could add this empty `div` after the bio container:



BY BECK JOHNSON

Beck Johnson is a developer based in Seattle, WA. She plays board games in her spare time. Say hi at beckjohnson@gmail.com

```
<div style="clear: both"></div>
```

(We could apply the class to any type of element, but the benefit of using a div is that it has no style of its own.)

THE OVERFLOW PROPERTY

The other way to force a container to expand around floated content is to apply a CSS rule with `overflow` to the container that the floated content is inside.

Any valid value for `overflow` will cause floated content to stretch out the container

- Too complicated to explain, but it basically forces the container to re-assess the content inside it

THE OVERFLOW PROPERTY

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

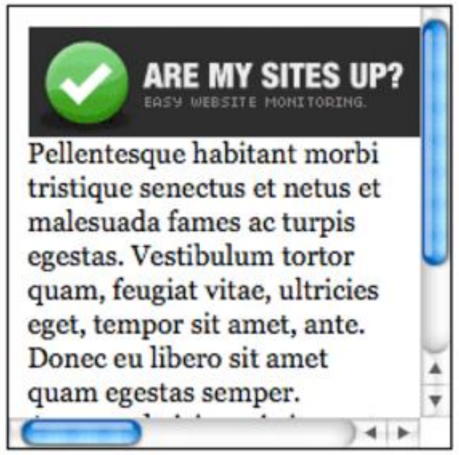
By default, elements have `overflow: visible`, which means all content is fully visible

- Even if that means overflowing its container!

THE OVERFLOW PROPERTY

`overflow: scroll` makes scrollbars appear both horizontally and vertically…even if they don't need to be there.

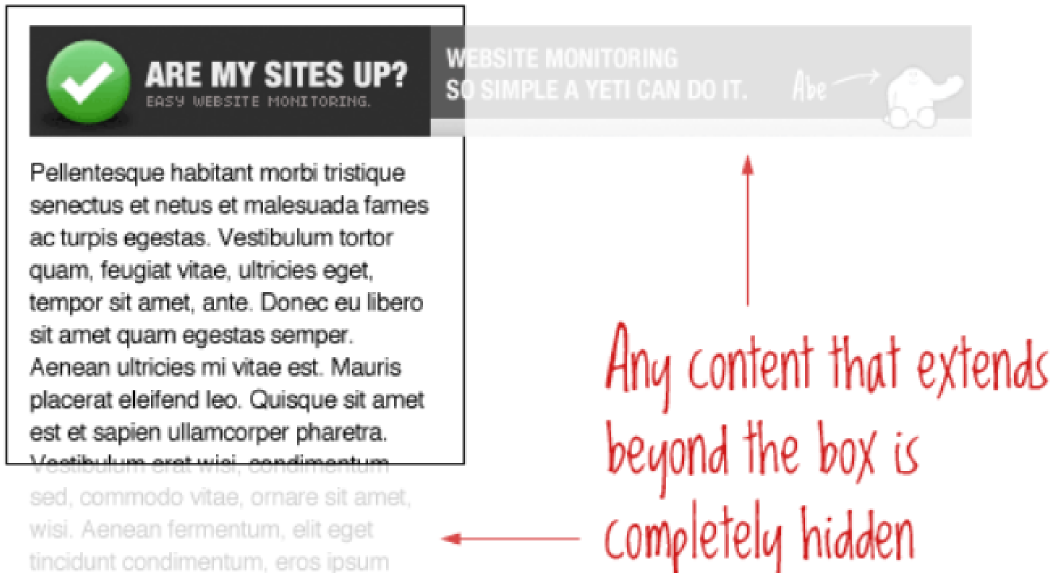
- None of the content that would overflow appears outside the box



Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper.

`overflow: hidden` cuts off any content that "sticks out" of its box

- No way to scroll, so content is no longer accessible

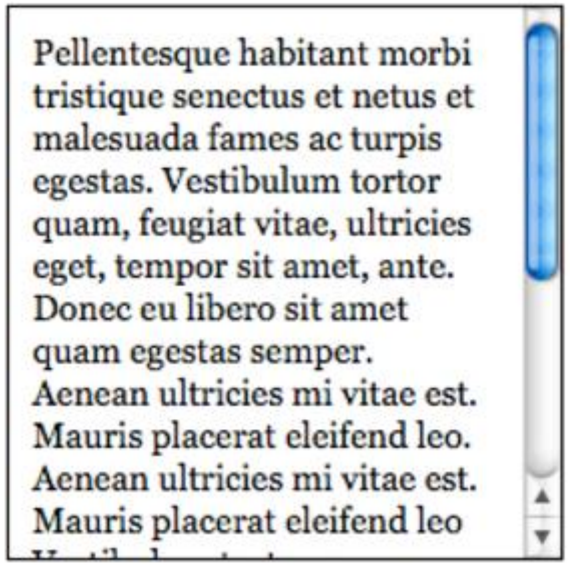


Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo. Quisque sit amet est et sapien ullamcorper pharetra. Vestibulum erat wisi, condimentum sed, commodo vitae, ornare sit amet, wisi. Aenean fermentum, elit eget tincidunt condimentum, eros ipsum

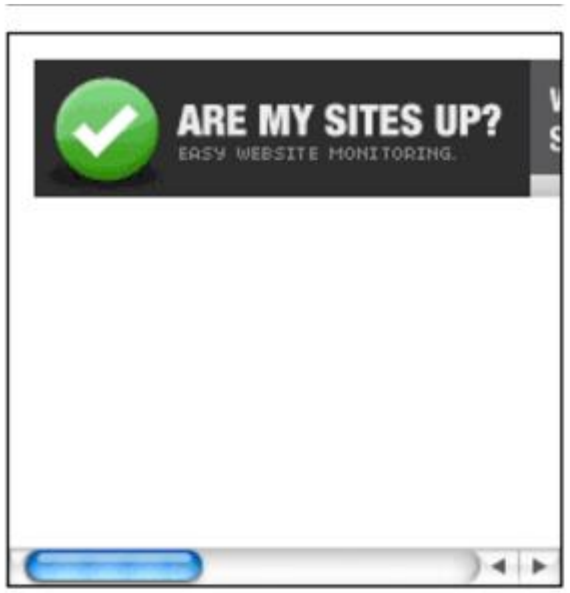
Any content that extends beyond the box is completely hidden

THE OVERFLOW PROPERTY

`overflow: auto` only adds scrollbars when the content requires it (which may mean no scrollbars are added at all)



Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo. Aenean ultricies mi vitae est. Mauris placerat eleifend leo



THE OVERFLOW PROPERTY

So to solve our problem, you could add this CSS rule to the floated `div`:

```
.bio { overflow: visible; }
```

PRACTICE TIME!

ASSIGNMENT

Create a container that has an image floated to the side of some text.

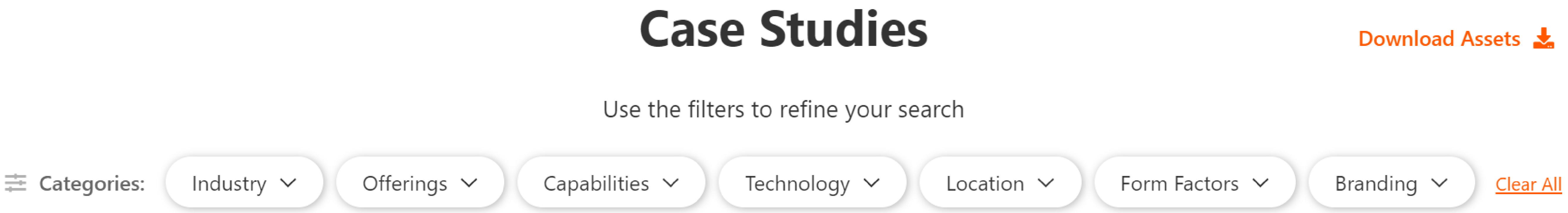
- Give the container a background color, gradient, or borders (to make sure floated content is really clearing properly and doesn’t just “look” cleared)
- Try both techniques to force the container to expand around floated content
- Apply box model properties like padding and margin so that your content looks nice

WEB LAYOUTS

LAYOUT WITH FLEXBOX

Instead of using `float` to change the layout of your page, you can use `display: flex`

Adding `display: flex` to a container puts every direct child inside it one line



Case Studies

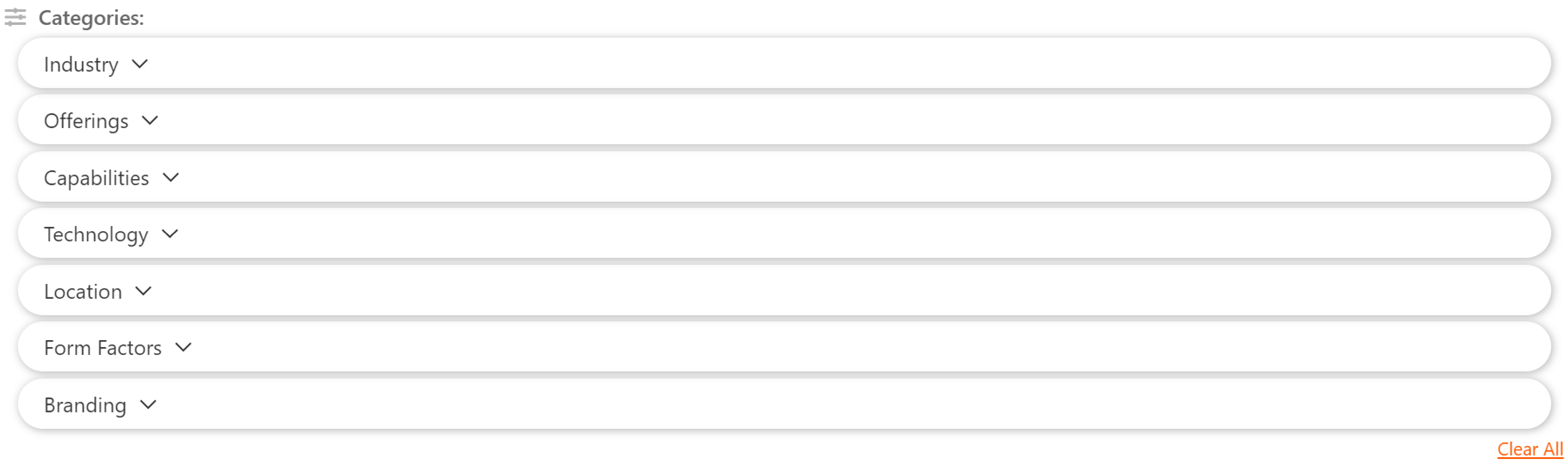
Use the filters to refine your search

LAYOUT WITH FLEXBOX

```
<section class="filter-container">

    <p><i class="fa fa-sliders-h"></i>Categories</p>

    <div>
        <div class="filter-button">Industry</div>
        <div class="filter-button">Offerings</div>
        <div class="filter-button">Capabilities</div>
    </div>

    <a id="clear" role="button">Clear All</a>

</section>
```

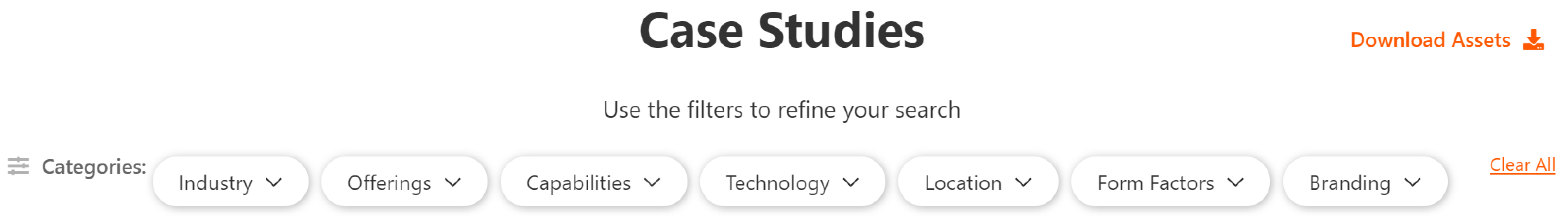
LAYOUT WITH FLEXBOX

With no flex at all, this is how that layout looks:



LAYOUT WITH FLEXBOX

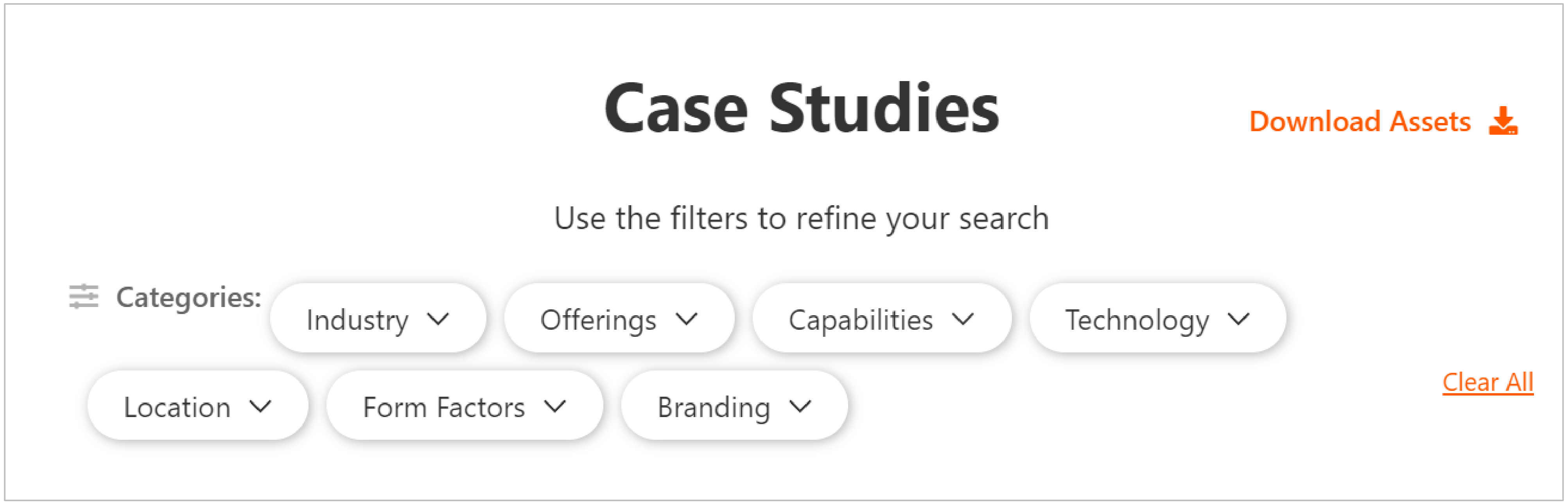
Using `float` instead of `display: flex`



Case Studies

Download Assets

Use the filters to refine your search



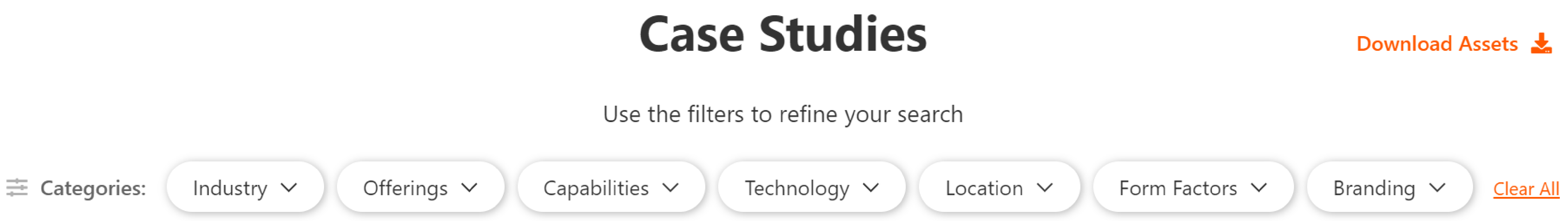
Case Studies

Download Assets

Use the filters to refine your search

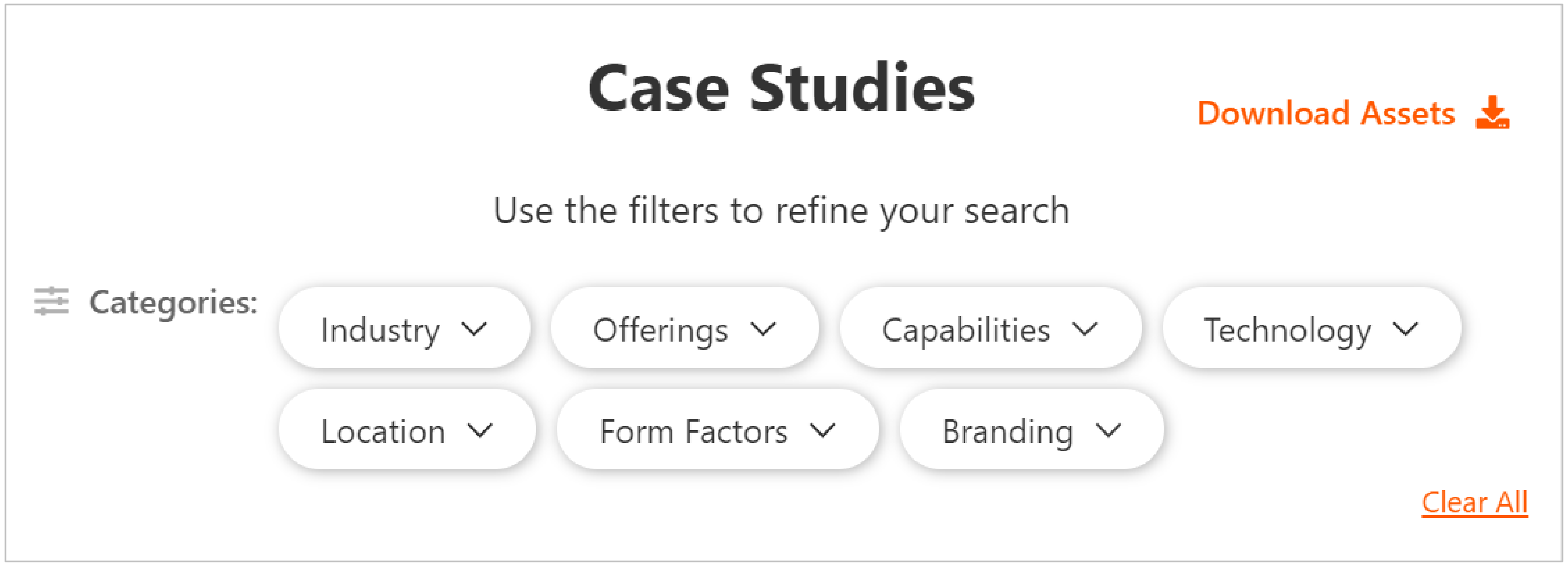
LAYOUT WITH FLEXBOX

```
.filter-container { display: flex; }
```



Case Studies

Use the filters to refine your search



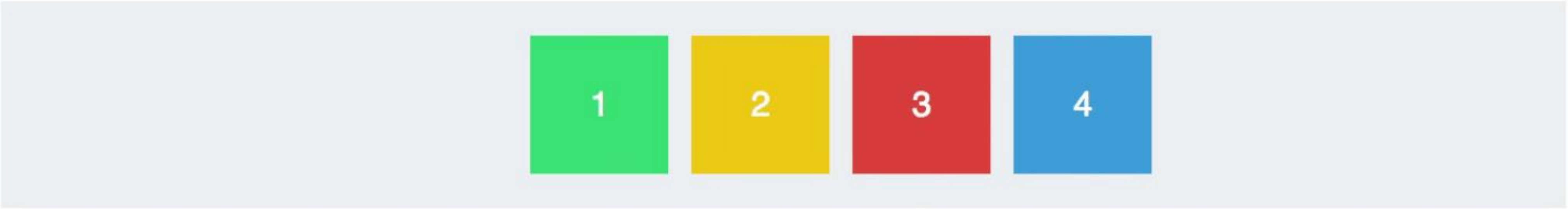
Case Studies

Use the filters to refine your search

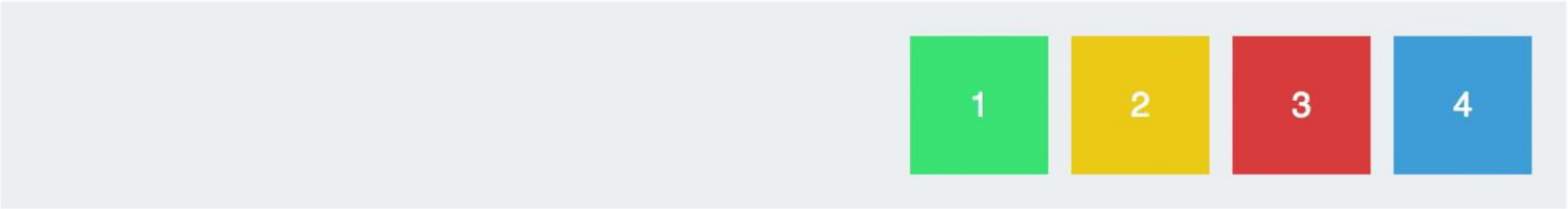
JUSTIFY-CONTENT

`justify-content:` controls how space is distributed between flex children horizontally

`justify-content: center` puts everything together in the middle

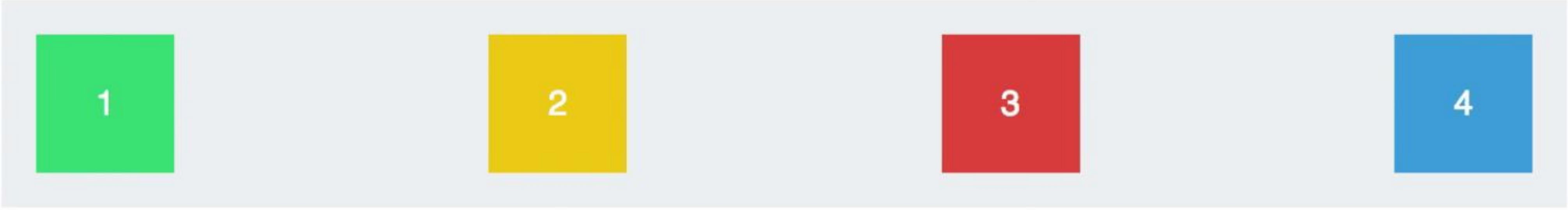


`flex-start` and `flex-end` move everything to either the beginning or end of the row

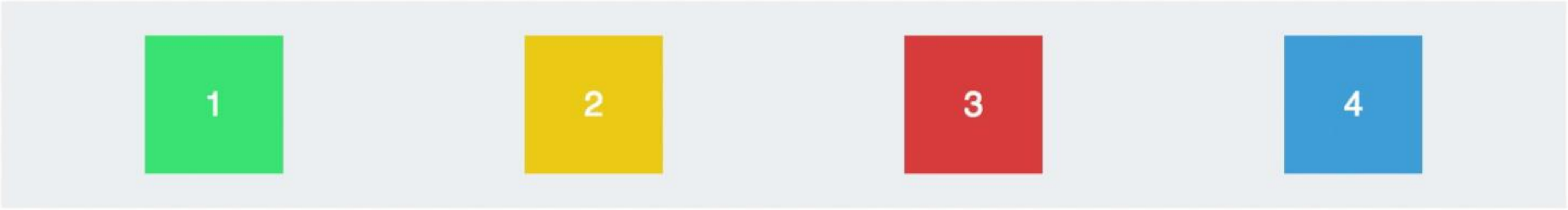


JUSTIFY-CONTENT

`justify-content: space-between` spreads all the items out evenly, with the first and last items flush against the ends of the container



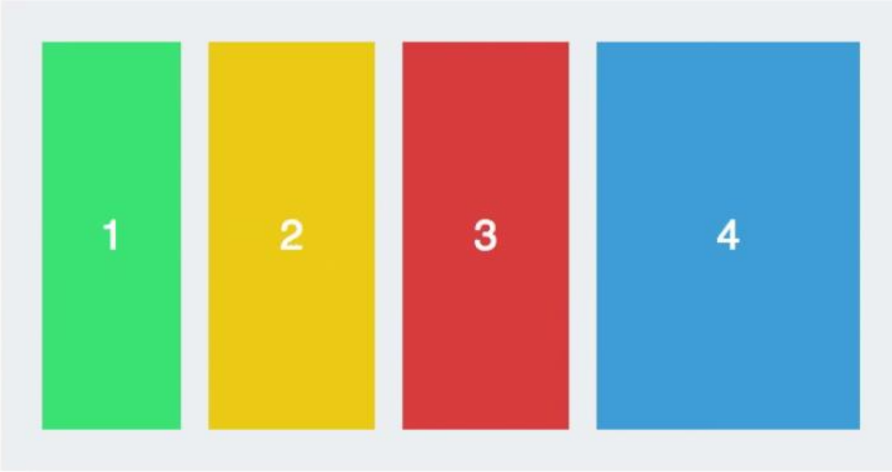
`justify-content: space-around` distributes all the items with equal space around them



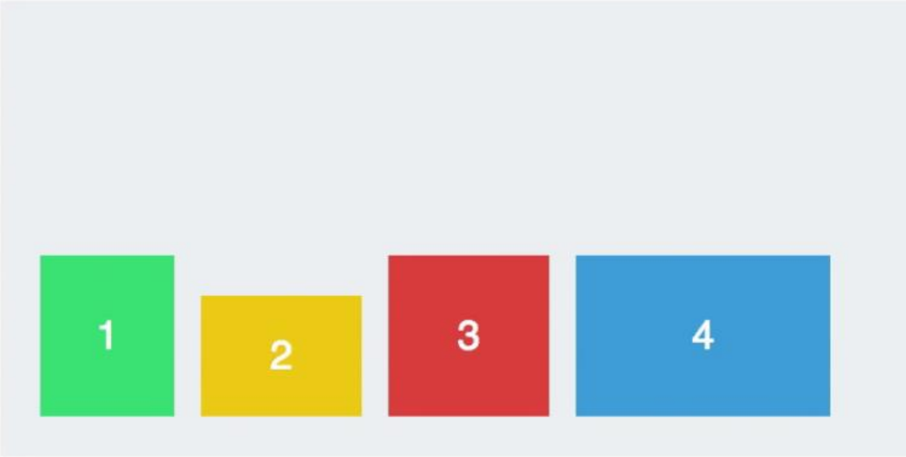
ALIGN-ITEMS

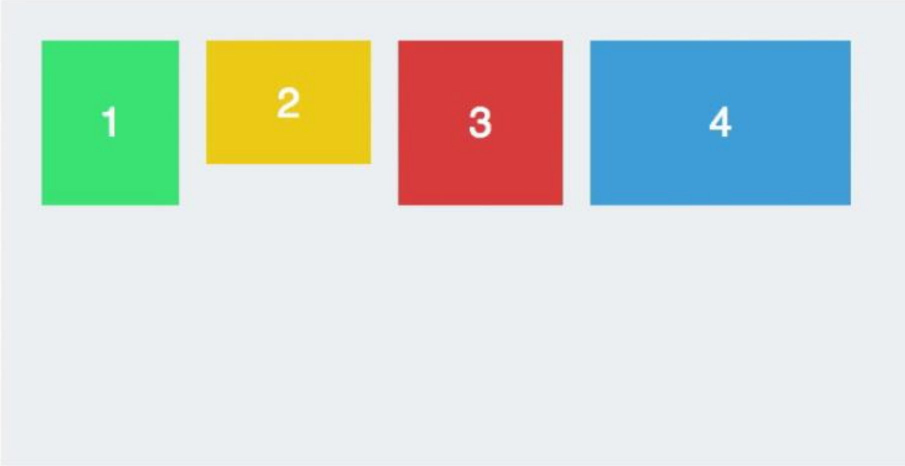
`align-items:` controls how items are aligned relative to one another vertically

`align-items: stretch` makes all children stretch to the same height, (unless they already have a `width` defined by CSS)



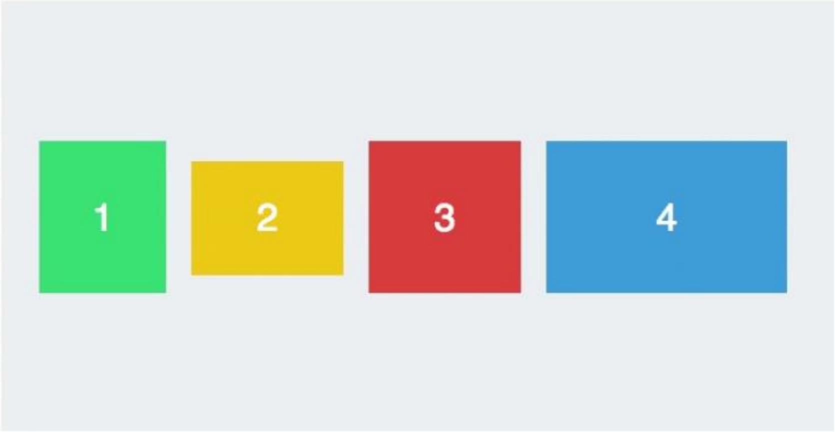
`flex-start` and `flex-end` align all children to either the top or bottom of their container



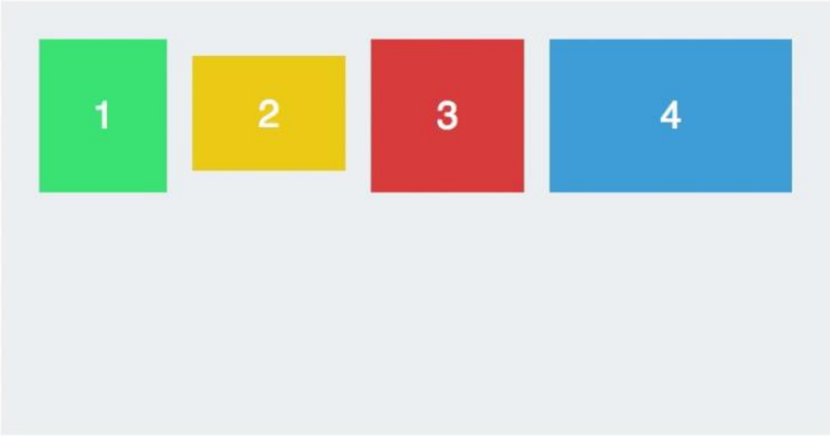


ALIGN-ITEMS

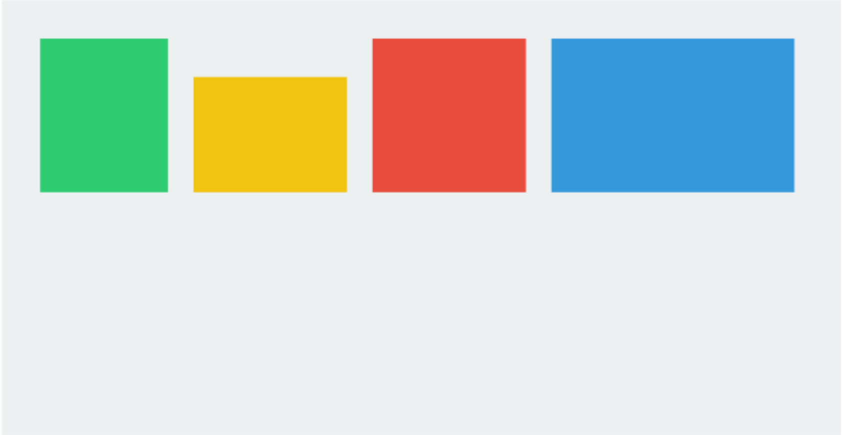
`align-items: center` makes all children vertically centered, with equal space on top and bottom



`align-items: baseline` vertically centers children so that their text lines up



If there is no text, the bottoms of the items will be aligned instead



LAYOUT WITH FLEXBOX

PROS

- Don’t have to worry about the clearing problem
- Have better control of how children are aligned
- Can use CSS to change the order in which children are shown

CONS

- May require extra markup around flex children to prevent everything from becoming `flex`
- May require extra markup to create a flex parent
- Limited support in old browsers (Internet Explorer 9, 8, 7)

PRACTICE TIME!

ASSIGNMENT

Using last weeks last “match the comp” html file

- Add a `<main>` element (if you don’t already have one) that surrounds your card markup, directly after `<body>`
- Copy and paste the cards until there are at least 3
    - Note: if you added a `height` to the card you will want to remove it
- Give `main { display: flex; }`
- Play with `flex` properties until your cards look nice

WEB FONTS

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:

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

Remember that 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

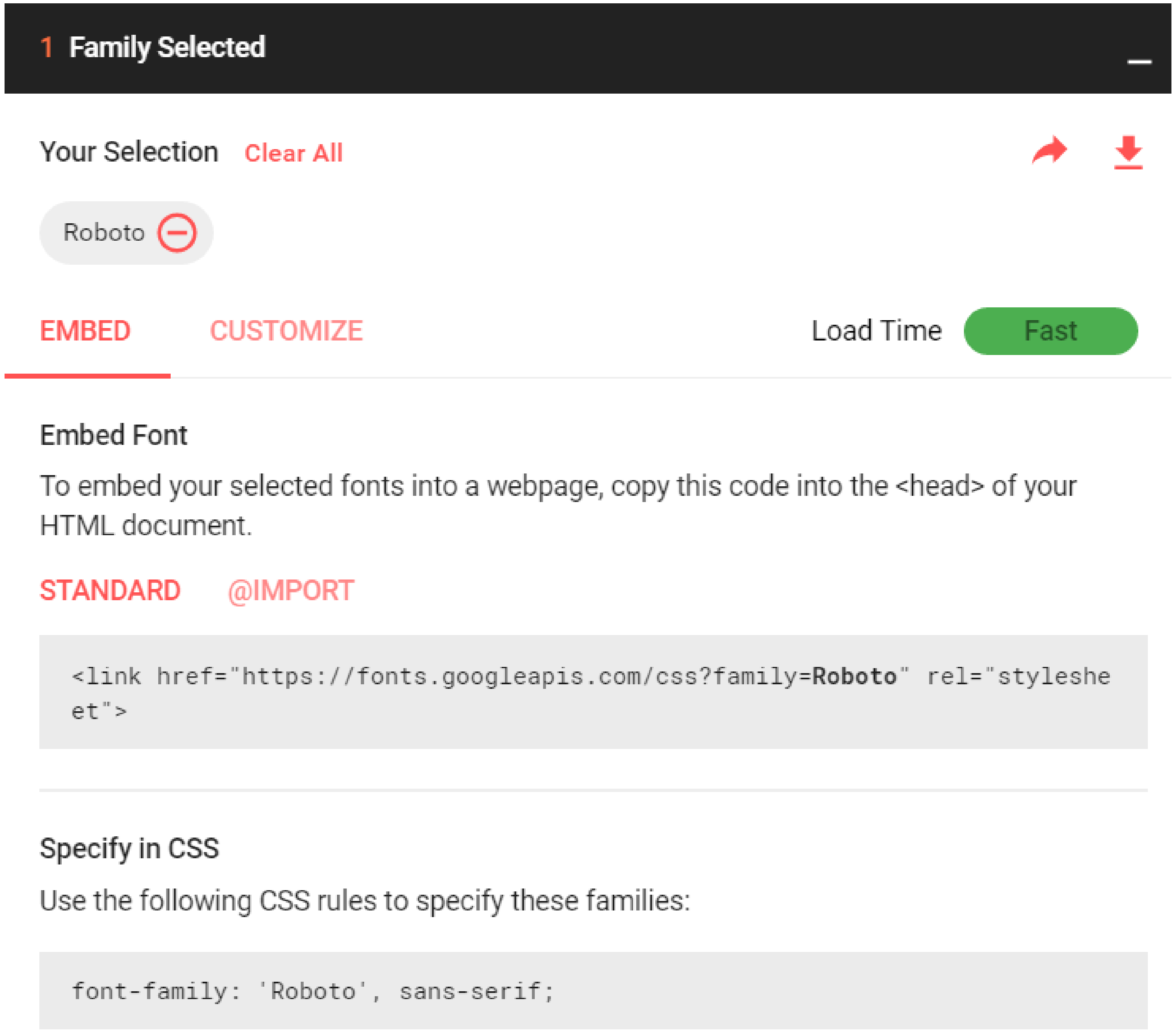
WEB FONTS

```
<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:

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

WEB FONTS



Embed Font

To embed your selected fonts into a webpage, copy this code into the <head> of your HTML document.

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

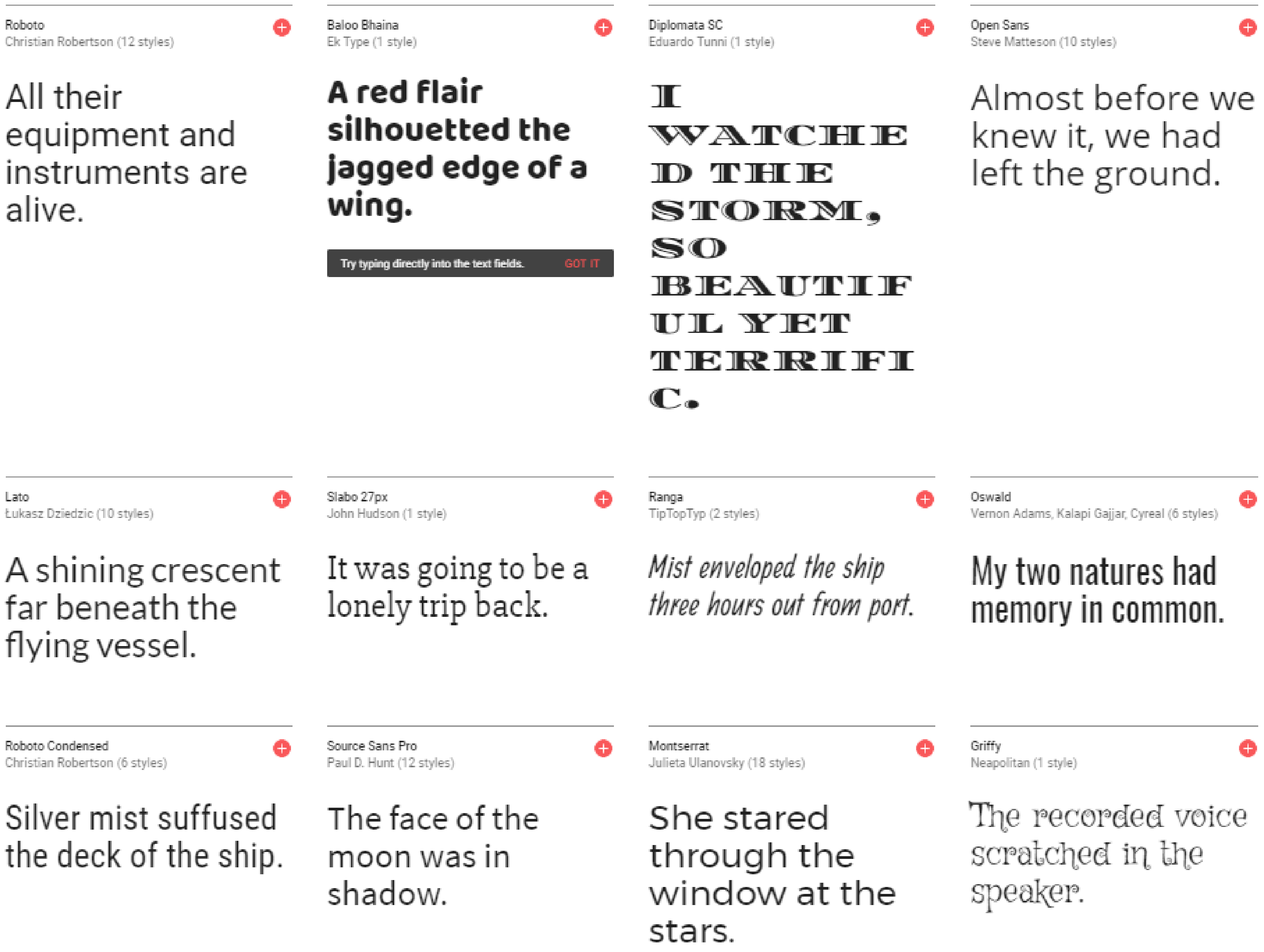
Specify in CSS

Use the following CSS rules to specify these families:

```
font-family: 'Roboto', sans-serif;
```

When you choose a font Google will tell you what name to use for it in your CSS

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/

WEB FONTS

CONS

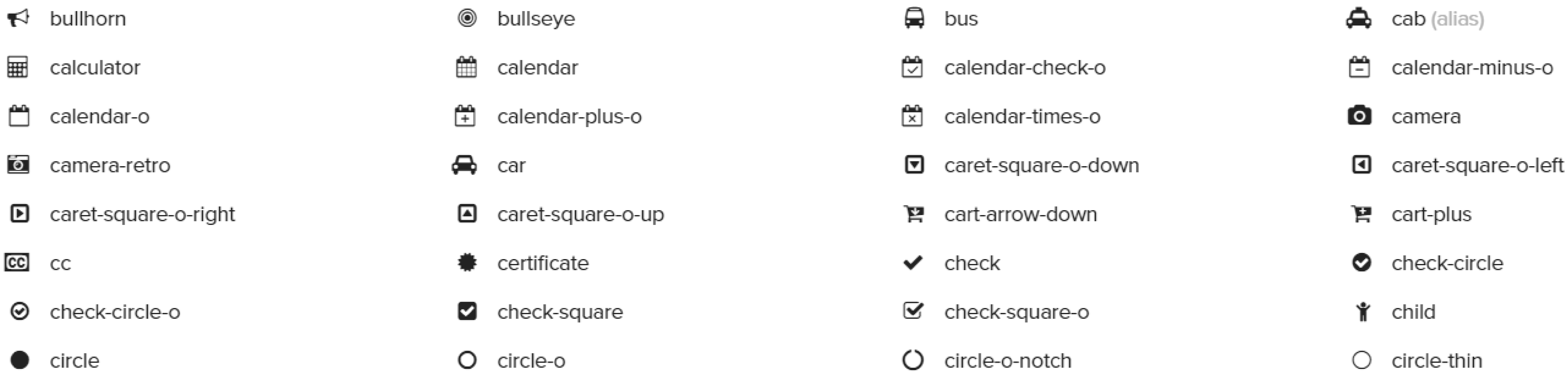
- Relies on a 3rd party to provide assets
- If the request times out, the font won’t download (so always provide fallbacks!)

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
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 rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.7.2/css/all.css">
```

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

```
<i class="fas fa-heart"></i>
```

FONT AWESOME

You can put a Font Awesome class on any element:

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

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

```
<h1 class="fas 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

WEB FONTS

Some considerations with hosting fonts:

- May have to pay licensing fees – many fonts cost money
- Some free fonts prohibit commercial use, or limit the number of page views

@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.woff2') format('woff2'),
       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

```
@font-face {
  font-family: MyWebFont;
  src: url('webfont.eot');
       url('webfont.eot?#iefix') format('embedded-opentype’),
       url('webfont.woff2') format('woff2'),
       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 or woff2
- IE needs eot
- Old 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:

```
<link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.7.2/css/all.css" >
```

- Display at least two different icons
- Make them different sizes and/or colors
- Bonus points: what other CSS can you apply to the icons?

RESPONSIVE DESIGN

MOBILE FIRST

An important principle of responsive design is "Mobile First"

- Both design and code should default to mobile resolution, adding progressive enhancements as the screen gets larger

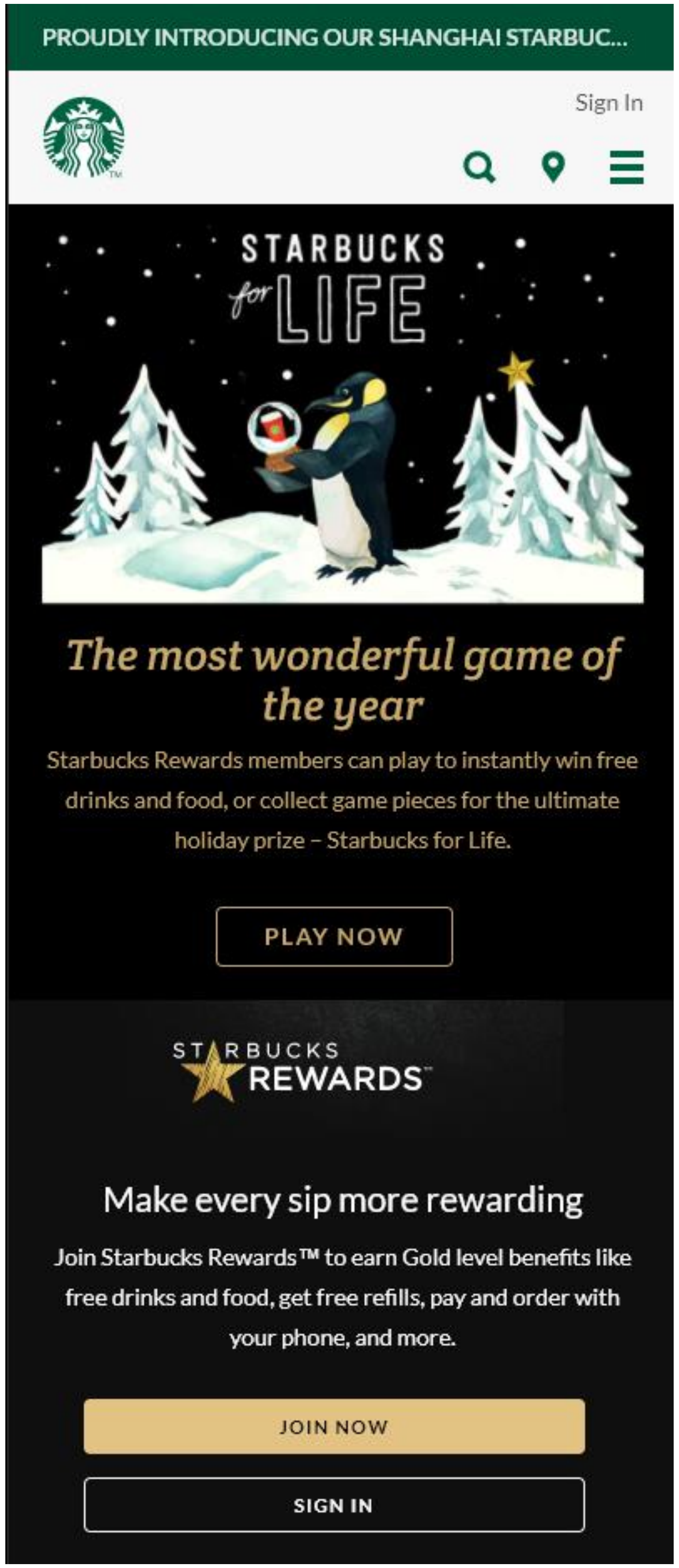
- One benefit of considering mobile first is that it trims down website content to its most vital elements
  - Mobile first = content first

RESPONSIVE != ADAPTIVE

Responsive design means that your design (and code) needs to function on a continuum of devices and screen sizes

- Although you should make care that things look great at specific “breakpoints” (640px wide for iPhone 4/5, 768px for iPad), it’s just as important to make sure things look good at *any* resolution
  - Users resize their browser windows
  - Technology changes

EXAMPLE — MOBILE



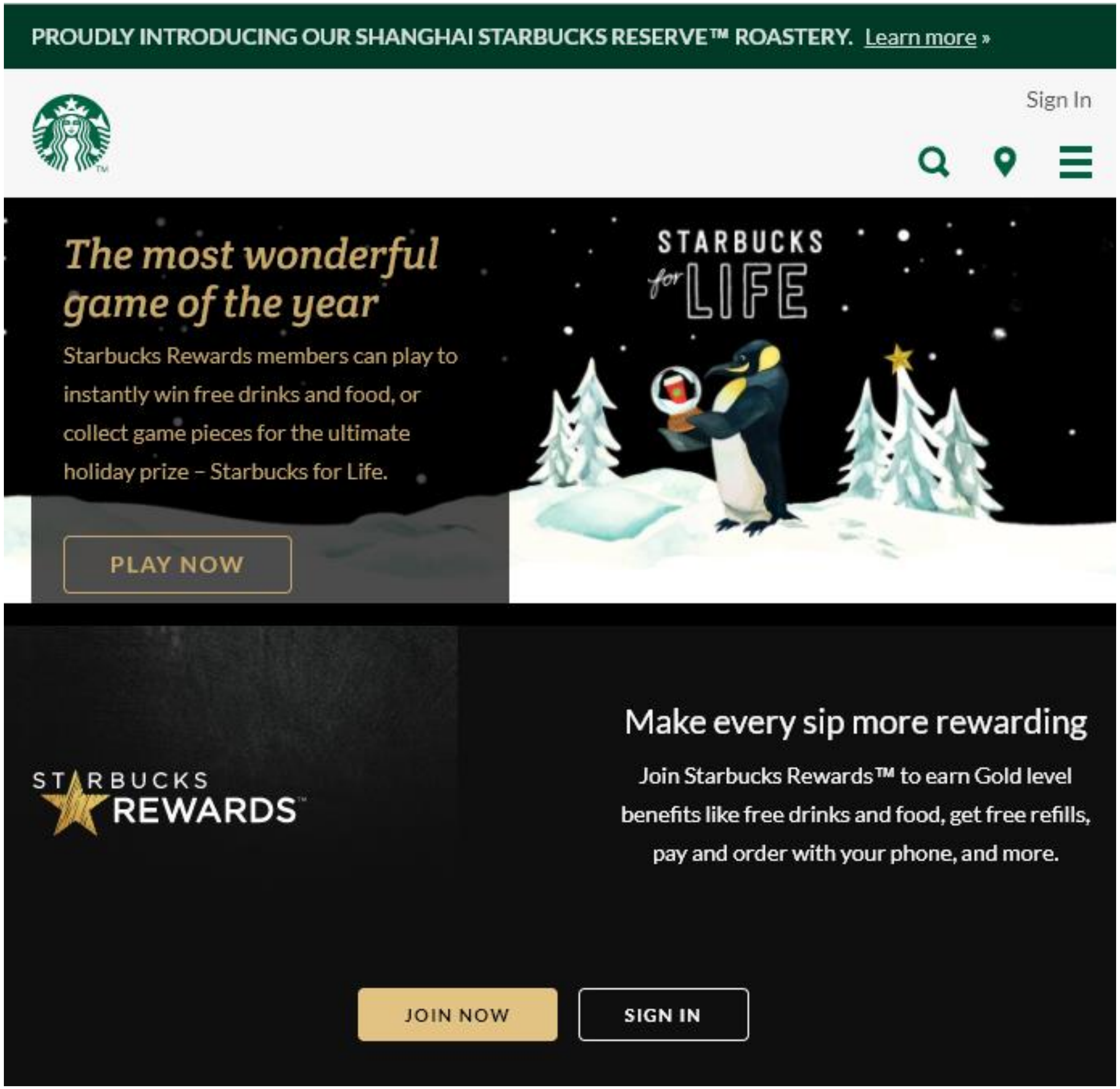
PROUDLY INTRODUCING OUR SHANGHAI STARBUC...

The most wonderful game of the year

Starbucks Rewards members can play to instantly win free drinks and food, or collect game pieces for the ultimate holiday prize – Starbucks for Life.

Make every sip more rewarding

Join Starbucks Rewards™ to earn Gold level benefits like free drinks and food, get free refills, pay and order with your phone, and more.



PROUDLY INTRODUCING OUR SHANGHAI STARBUCKS RESERVE™ ROASTERY. Learn more »

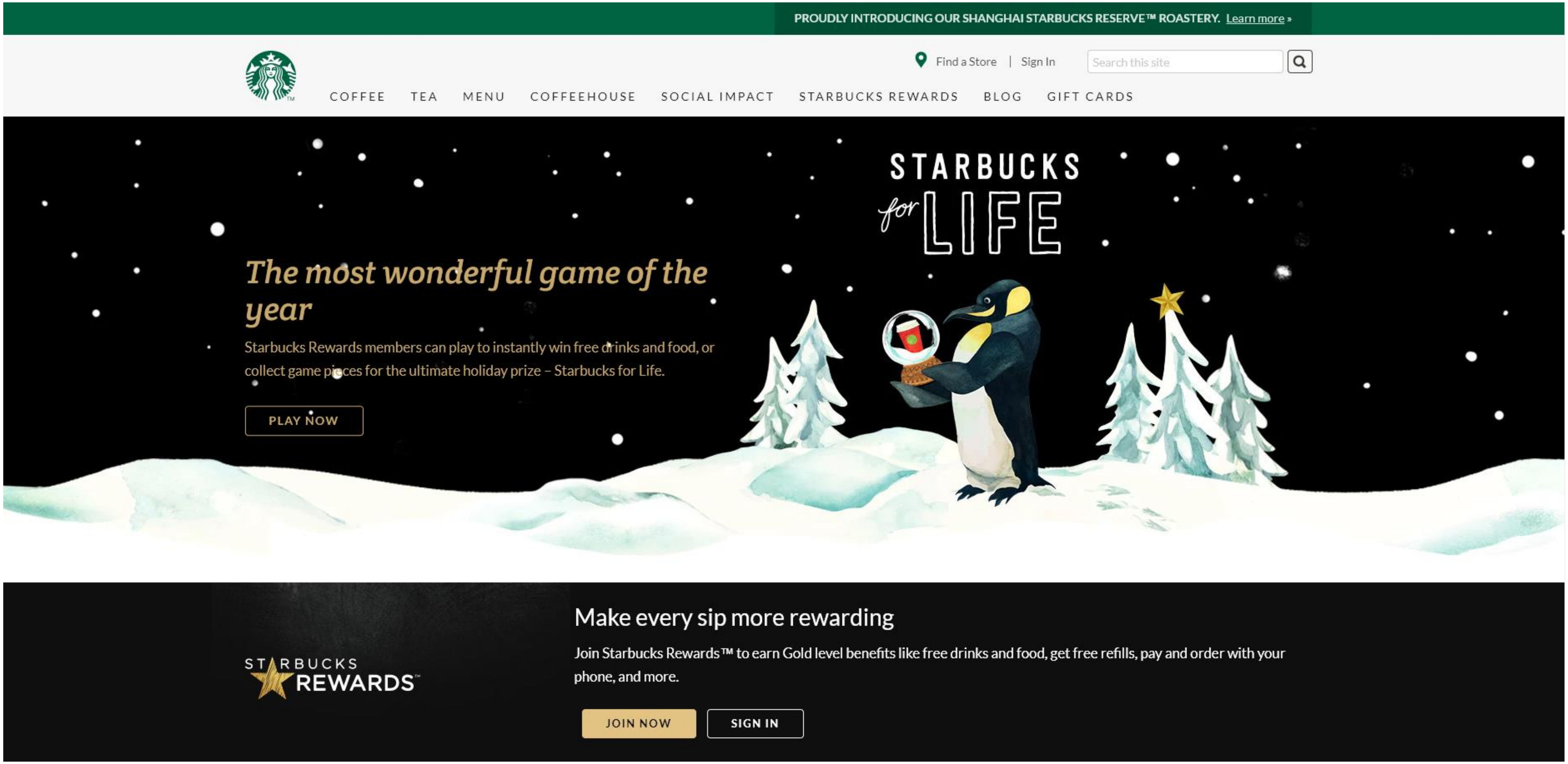
The most wonderful game of the year

Starbucks Rewards members can play to instantly win free drinks and food, or collect game pieces for the ultimate holiday prize – Starbucks for Life.

Make every sip more rewarding

Join Starbucks Rewards™ to earn Gold level benefits like free drinks and food, get free refills, pay and order with your phone, and more.

EXAMPLE — DESKTOP



PROUDLY INTRODUCING OUR SHANGHAI STARBUCKS RESERVE™ ROASTERY. Learn more »

The most wonderful game of the year

Starbucks Rewards members can play to instantly win free drinks and food, or collect game pieces for the ultimate holiday prize – Starbucks for Life.

Make every sip more rewarding

Join Starbucks Rewards™ to earn Gold level benefits like free drinks and food, get free refills, pay and order with your phone, and more.

MEDIA QUERIES

Media queries are used to apply different CSS to different devices.

Some things you can use a media query to detect:

- The minimum or maximum screen height or width
- Whether the screen is rotated (in “landscape view”)
- If the page is being printed
- If the user is on a touch screen device
- The screen’s resolution

MEDIA QUERIES

Media queries have a different format than any other CSS we've seen so far

They always start with `@media` and have curly braces that contain all the CSS that applies to that media query rule

Multiple rules can be tested for, separated by `and`

```
@media (max-width: 480px) and (orientation: landscape) {
     h1 { font-size: 20px; }
}
```

MEDIA QUERIES

To specify different styles when a webpage is being printed, use this media query:

```
@media print { }
```

This allows you to format your page so that it looks better on paper

- Make the page full screen
- Remove non-essential page elements (such as navigation links or social media icons)
- Set font color to black and background color to white

MEDIA QUERIES

Most modern “mobile first” websites have CSS that applies to phone-sized screens first

Then, anything specific to bigger screens goes in media queries that test for a **minimum** screen width, like this:

```
@media (min-width: 768px) {
}
```

MEDIA QUERIES

```
/*   CSS that is used for phones, and also
     applies generally to all resolutions  */

@media (min-width: 768px) {
     /* CSS that is different for tablets */
}

@media (min-width: 920px) {
     /* CSS that is different for desktops */
}
```

MEDIA QUERIES - EXAMPLE

```
h1 {
     font-size: 12px;
     color: gray;
     font-family: sans-serif;
}


@media (min-width: 768px) {

     h1 { font-size: 20px; }

}


@media (min-width: 920px) {

      h1 { font-size: 25px; }

}
```

PRACTICE TIME!

ASSIGNMENT

Apply a media query to an element on your page so that it looks different when you resize your browser screen

Things to consider doing:

- Make a two- or three-column layout only in desktop (for example a sidebar)
- Make font sizes bigger for larger screen sizes, especially headings like `h1` or `h2`

HOMEWORK

HOMEWORK

Create a footer for your webpage (see examples on following pages)

- Add icons that link to your social media accounts using Font Awesome (or another icon library)
  - Remember you can use CSS to draw circles around them!

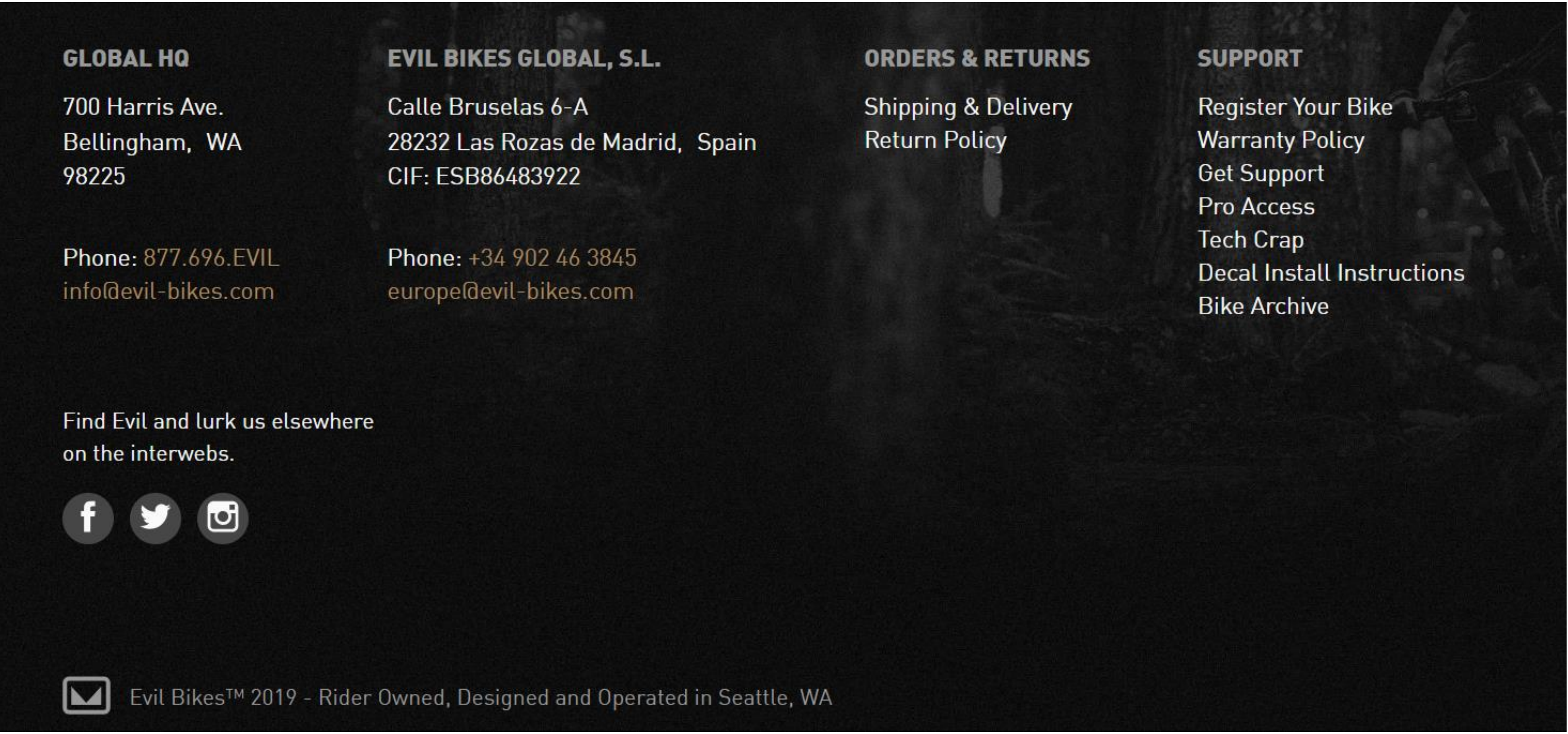
- Use either float or flex to put your footer content in at least two columns

- Include a copyright as the final row in the footer

- Bonus: Switch to single-column layout for mobile only

Email me your files at beckjohnson@gmail.com

EXAMPLE



GLOBAL HQ

700 Harris Ave.
Bellingham, WA
98225

Phone: 877.696.EVIL
info@evil-bikes.com

EVIL BIKES GLOBAL, S.L.

Calle Bruselas 6-A
28232 Las Rozas de Madrid, Spain
CIF: ESB86483922

Phone: +34 902 46 3845
europe@evil-bikes.com

ORDERS & RETURNS

Shipping & Delivery
Return Policy

SUPPORT

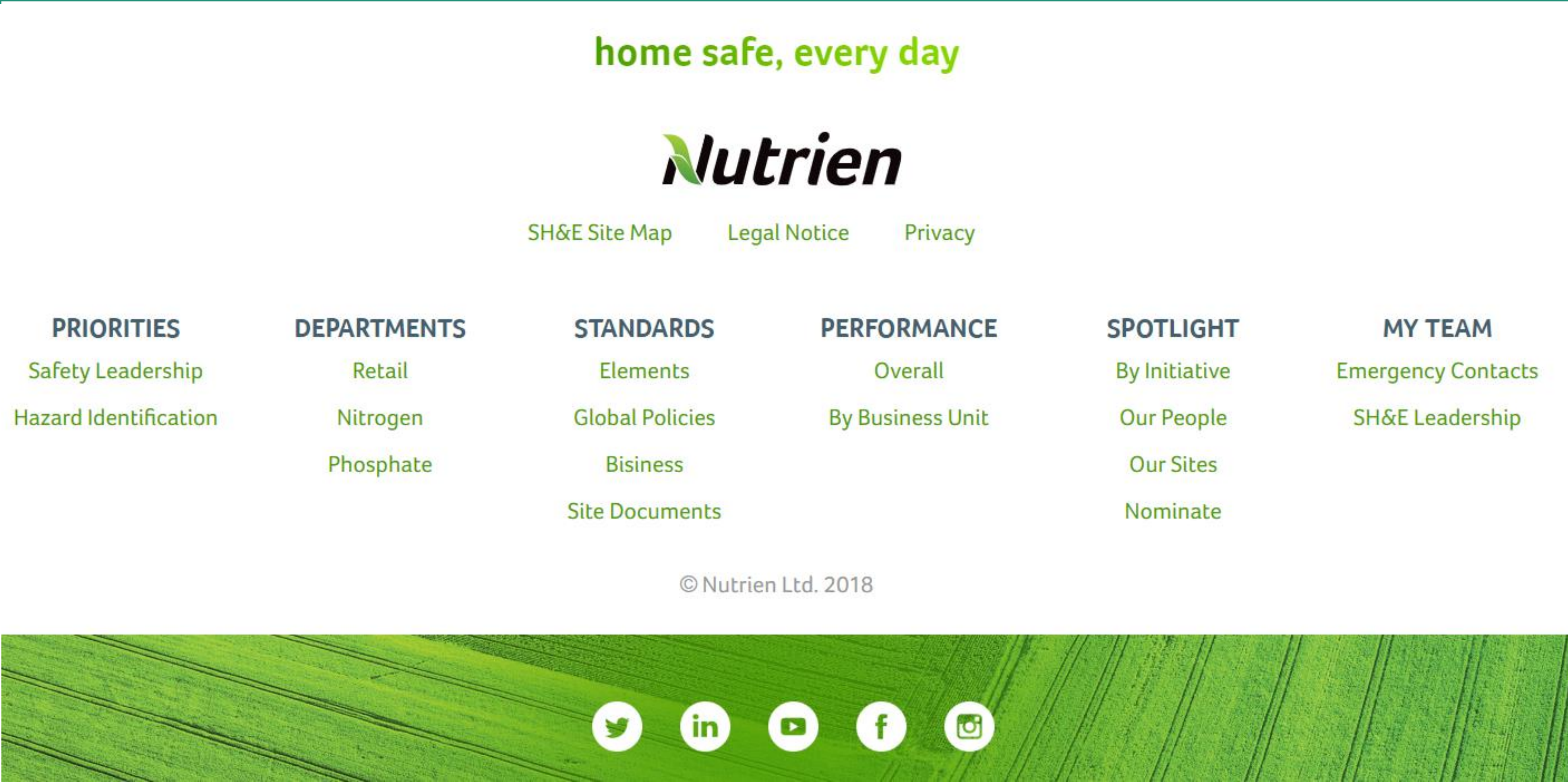
Register Your Bike
Warranty Policy
Get Support
Pro Access
Tech Crap
Decal Install Instructions
Bike Archive

Find Evil and lurk us elsewhere on the interwebs.

Evil Bikes™ 2019 - Rider Owned, Designed and Operated in Seattle, WA

https://www.evil-bikes.com/

EXAMPLE



home safe, every day

Nutrien

SH&E Site Map Legal Notice Privacy

PRIORITIES
- Safety Leadership
- Hazard Identification

DEPARTMENTS
- Retail
- Nitrogen
- Phosphate

STANDARDS
- Elements
- Global Policies
- Bisiness
- Site Documents

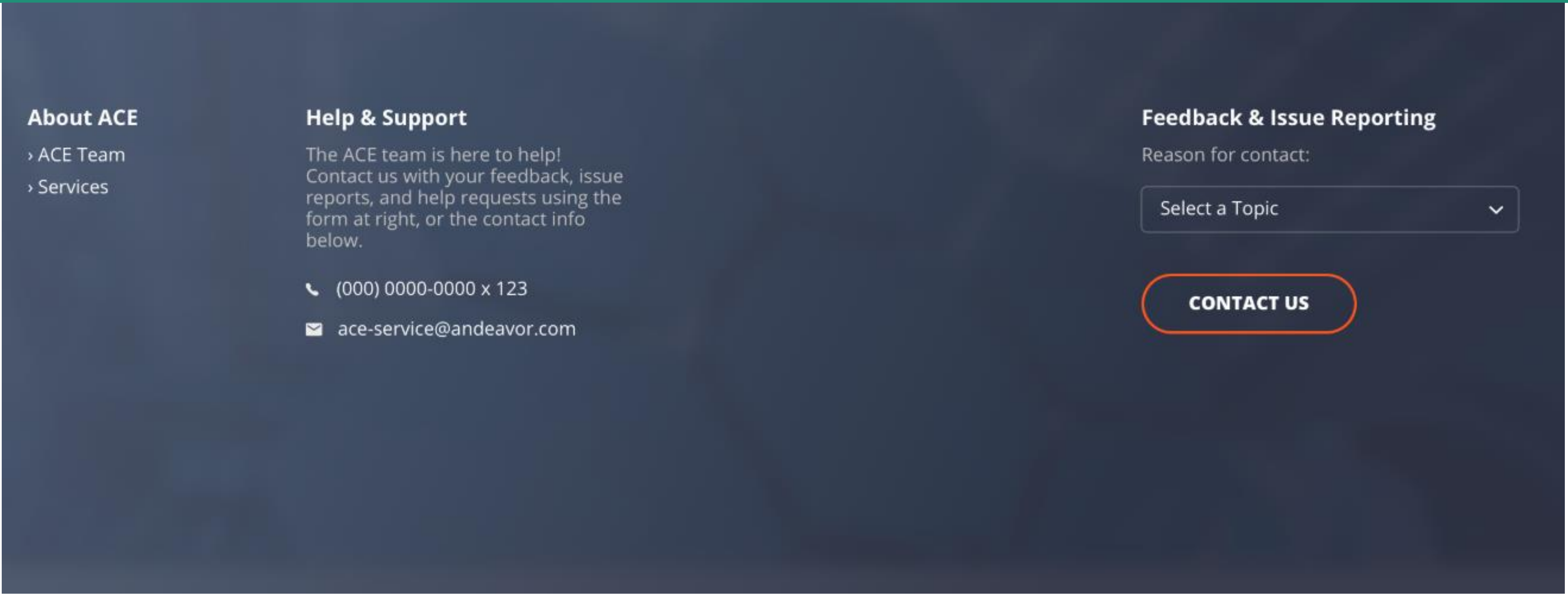
PERFORMANCE
- Overall
- By Business Unit

SPOTLIGHT
- By Initiative
- Our People
- Our Sites
- Nominate

MY TEAM
- Emergency Contacts
- SH&E Leadership

© Nutrien Ltd. 2018

EXAMPLE



About ACE

› ACE Team

› Services

Help & Support

The ACE team is here to help! Contact us with your feedback, issue reports, and help requests using the form at right, or the contact info below.

☎ (000) 0000-0000 x 123

✉ ace-service@andeavor.com

Feedback & Issue Reporting

Reason for contact:

“HOMEWORK”

Reminder: no class next week

I’ll send out a homework assignment next Wednesday via email so you guys don’t forget what you’ve learned!

- Practice!

- Optional: read chapters 15 and 17 of *HTML and CSS: Design and Build Websites*