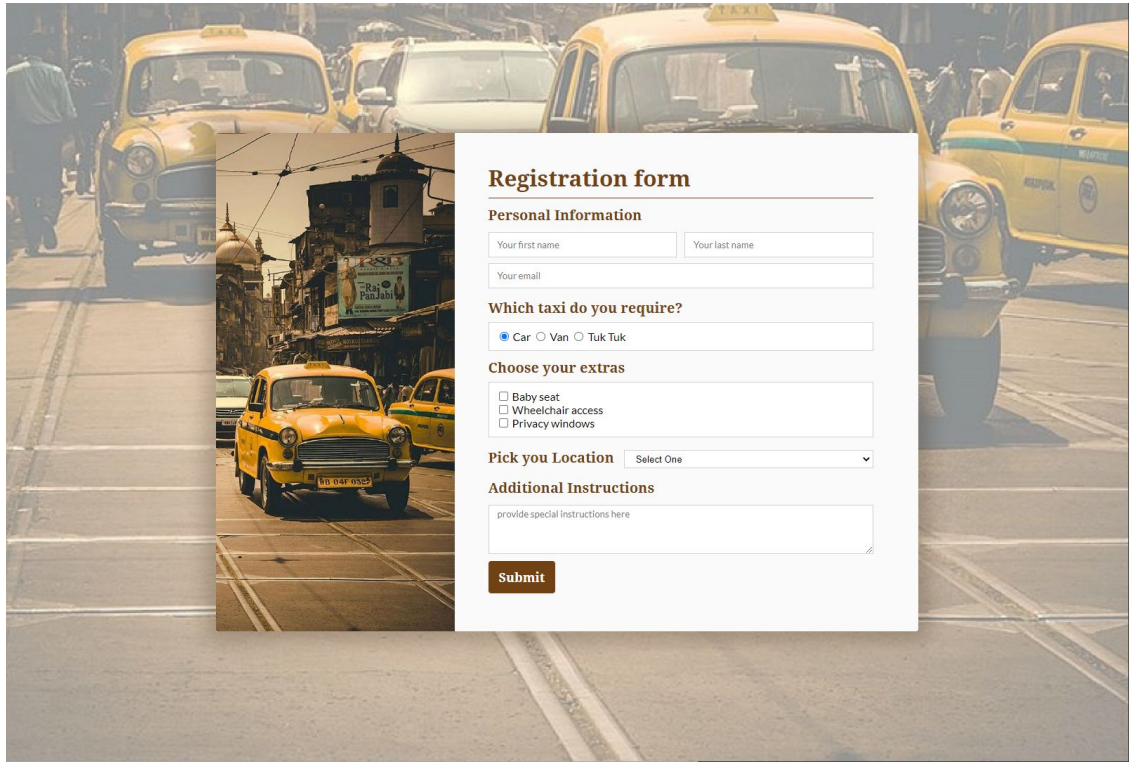
CSS Part 2

CSS: Page Layout

✋ Review!

Let’s review the activity!



CSS Pixels

- Your monitor is divided into pixels (screen pixels.)
- When you write CSS, you specify dimension and position in **px** units.

This **px** unit is NOT a screen pixel, but a CSS pixel.

CSS Pixels

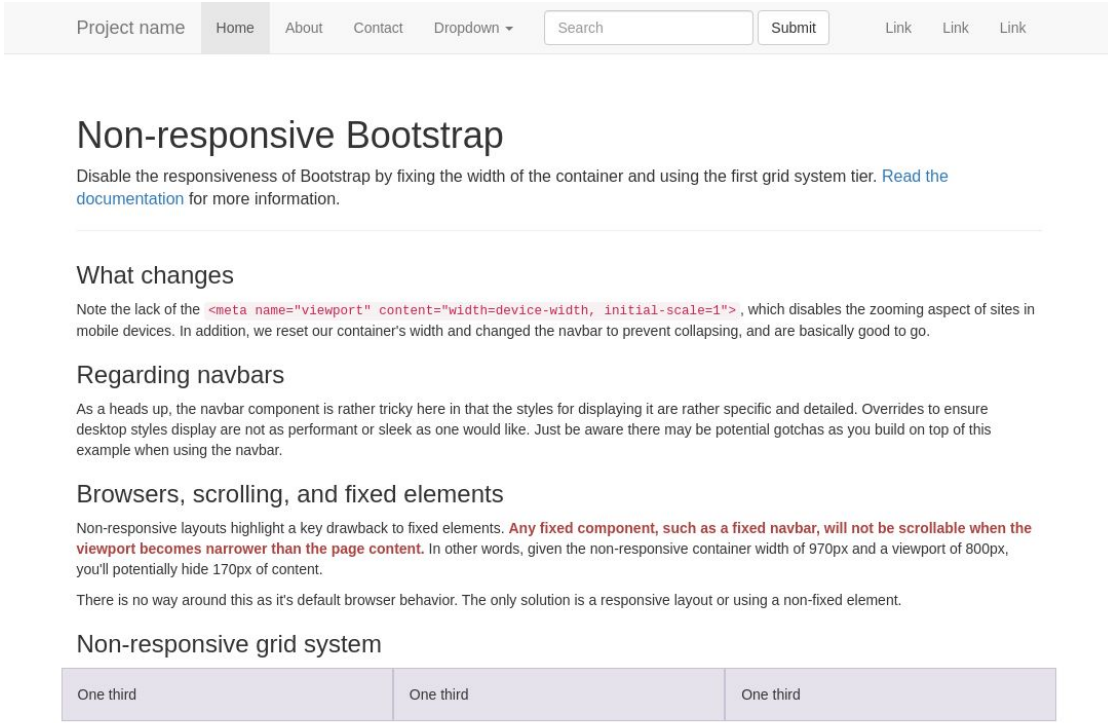
Different screens and their resolutions:

- Small laptop resolution: 1366 x 768
- “1080p” resolution: 1920 x 1080
- iPhone X screen resolution: 2436 x 1125

“Retina display”: Fancy Apple branding term that means that every *normal* pixel is made up of 4 pixels.

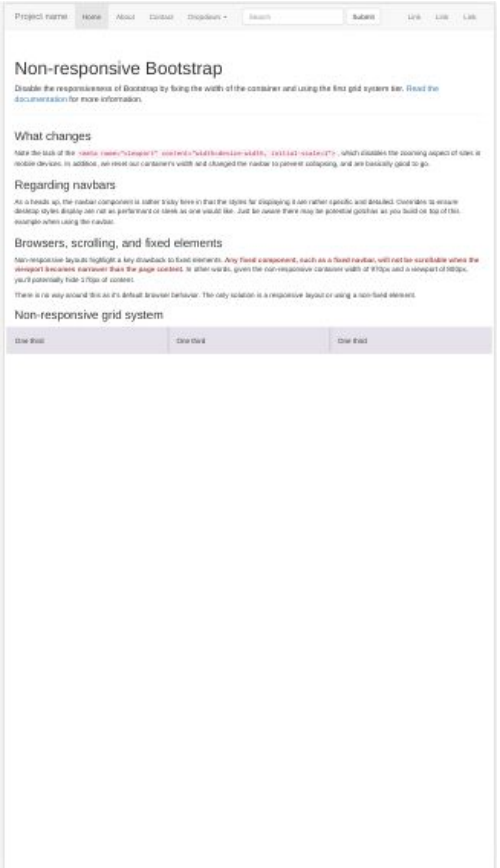
CSS Pixels

Example of a desktop site



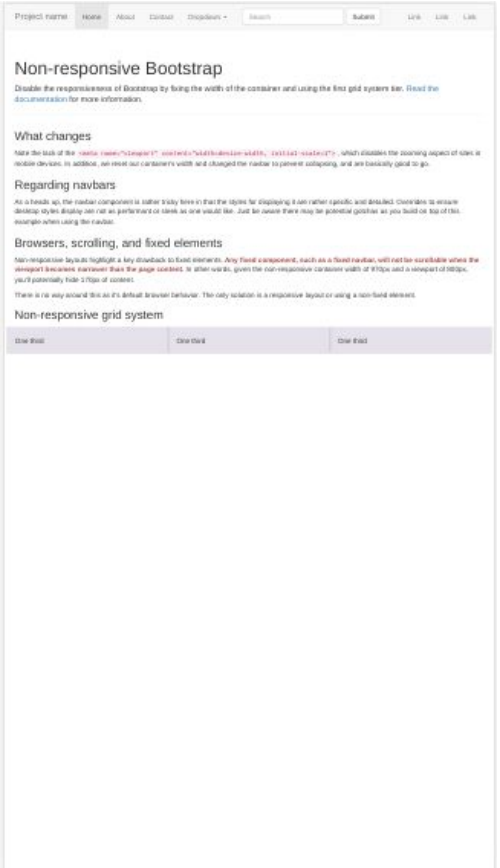
CSS Pixels

The same site on a mobile device.



CSS Pixels

The same site on a mobile device.



A quick fix... add the following to your HTML pages:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

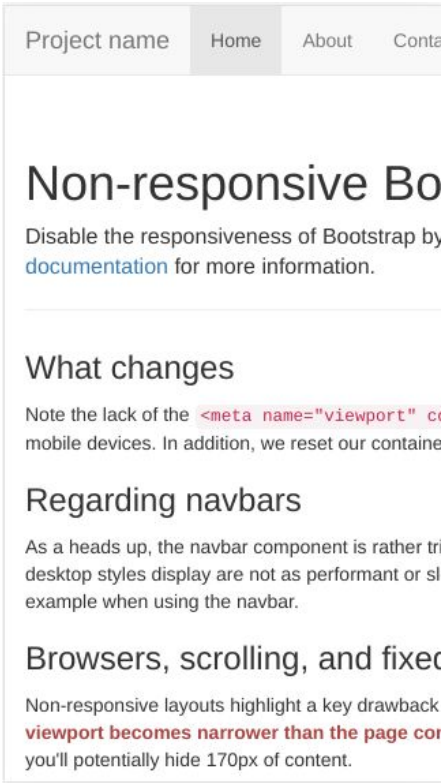
This will make the CSS pixels scale on mobile devices.

CSS Pixels

A quick fix... add the following to your HTML pages:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

This will make the CSS pixels scale on mobile devices.



Viewable on all screens!

Whatever you build will most likely need to be viewable on all kinds of screens.

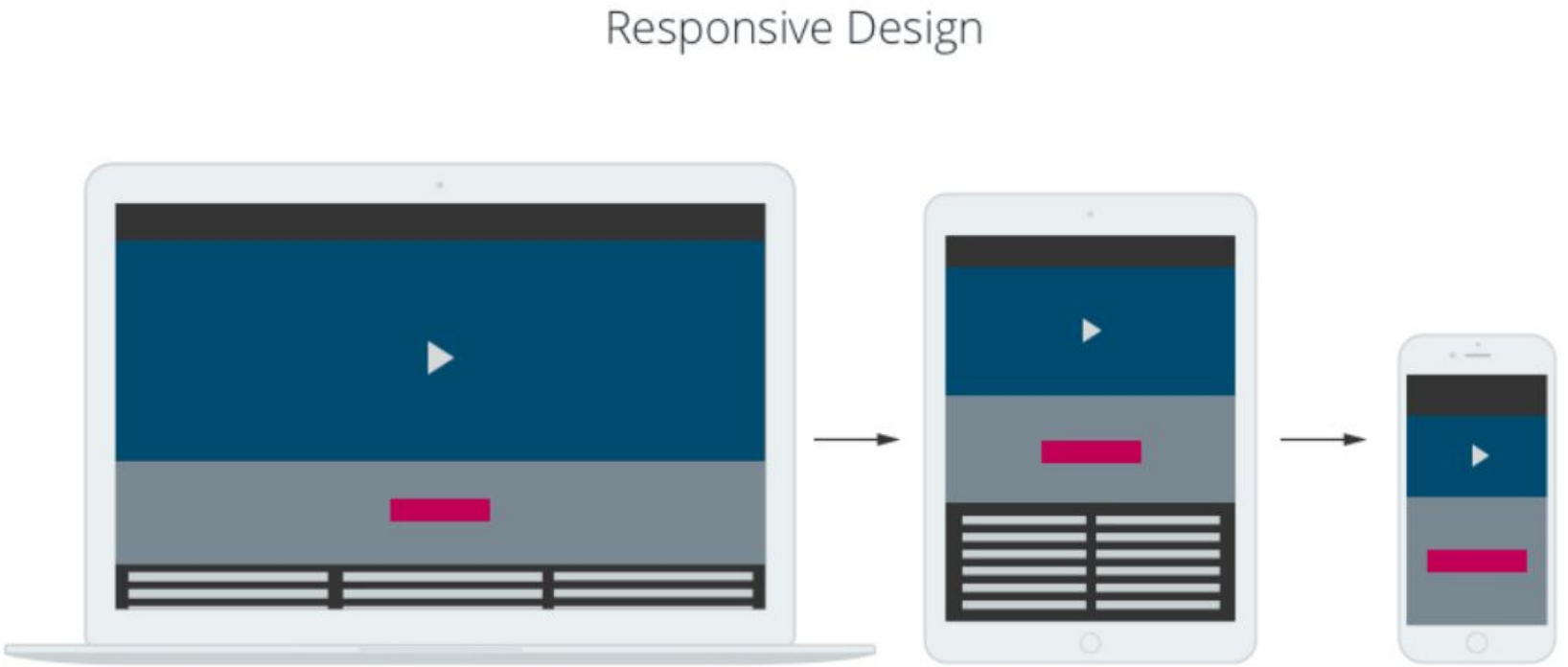
Websites shift content around depending on the screen size.

👀 **Montreal Gazette** 👀

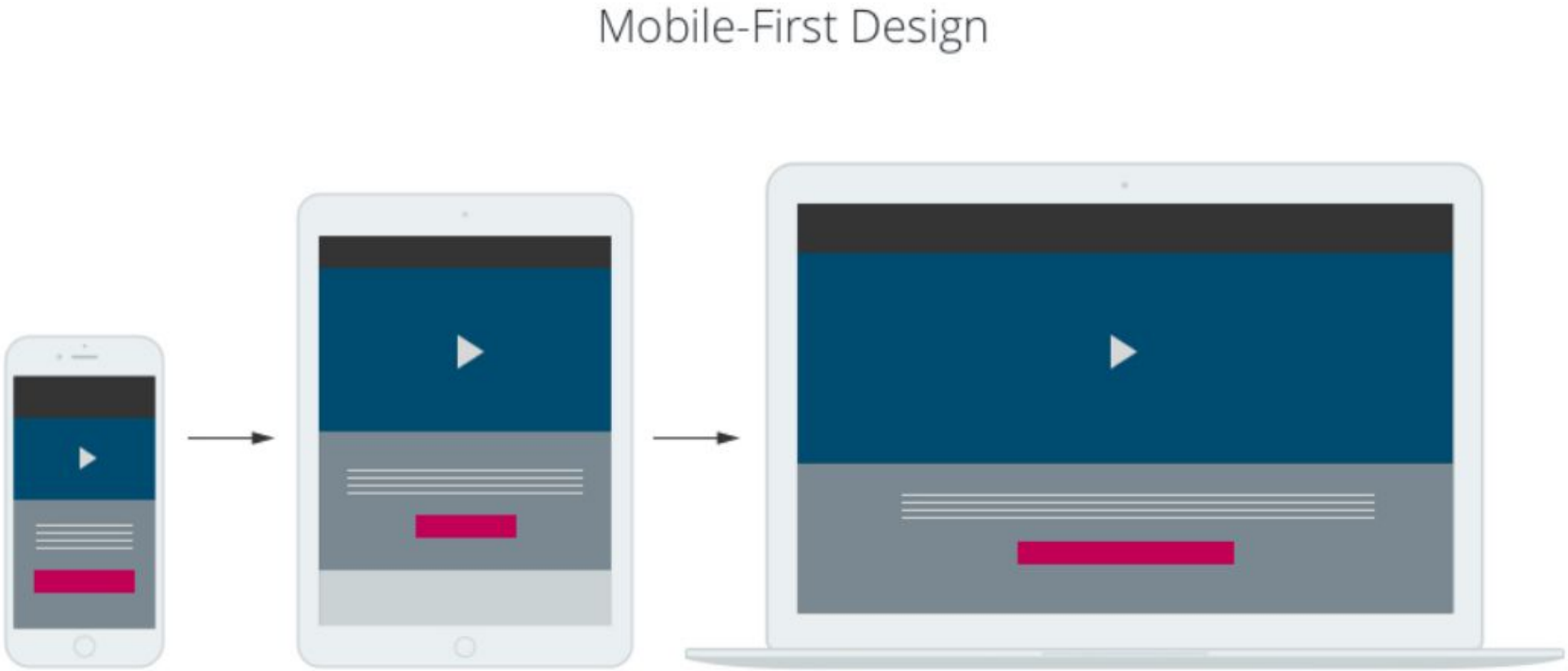
How do we go about building this fluidity into our web apps?

Enter **responsive** and **mobile-first**.

Responsive



Mobile-first



CSS media queries 🥳

How do we implement this?

Media queries can be used to check many things.

- width and height of the viewport
- width and height of the device
- orientation (landscape / portrait)
- resolution

Mobile-first please.

In general, we recommend adopting a mobile-first philosophy.

How do we do this?

1. Write all of the CSS for a mobile version.
2. Increase the viewport to the next viewport width.
3. Add CSS (within a media query) to override the base CSS.
4. Repeat steps 2 and 3 as you move the next biggest viewport.

Exercises

Let’s practice using media queries!

Exercise 1 (2 min)	https://codepen.io/gnomecircle/pen/VwKgQQE
Exercise 2 (2 min)	https://codepen.io/gnomecircle/pen/poEGaKN
Exercise 3 (3 min)	https://codepen.io/gnomecircle/pen/vYXbdaP

CSS Viewport units

There are many units in CSS.

These could definitely be useful:

- **vh**: percentage of the viewport height
- **vw**: percentage of the viewport width

```
<p class=”saying”>Where is my bacon?!</p>
```

```
.saying {
  background: red;
  color: white;
  height: 100vh;
  width: 100vw;
}
```

CSS Pseudo Selectors

A CSS pseudo-class is a keyword added to a selector that specifies a special state of the selected element(s).

For example, **:hover** can be used to change a button's color when the user's pointer hovers over it.

```
<a class=”btn”>Click!</p>
```

```
.btn {
  background: red;
  color: white;
  padding: 10px 14px;
}

.btn:hover {
  background: yellow;
  color: black;
}
```

CSS Pseudo Selectors

Other popular pseudo selectors:

- :hover
- :focus
- :checked
- :first-child

https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes#Index_of_standard_pseudo-classes

CSS Reset

Browsers have their own style by default. This is fine most of the time until:

- The browser's style interferes with ours.
- The browser's style overwrites ours.

As programmers, we don't like it when things interfere with our code. To put the browser in its place, we use what is known as a CSS reset:
https://meyerweb.com/eric/tools/css/reset/

Copying the code in this link at the top of our CSS file will remove the browser's default styles and set everything to 0.