

Mobile pages and CSS tricks

Michael Chang
Spring 2023

Plan for today

Making pages that work for mbile

Viewport meta, CSS media queries

CSS animations and transitions

Pointer events

Implementing gestures

Web fonts and icons

Mobile behavior

By default, devices assume page not optimized for mobile

Zooms page out to ~1k pixel width

If your page handles mobile layouts

Add a [viewport meta](#) to <head>

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

Tells browser to use device width, don't zoom out

Probably should have this on all your pages

Media queries

```
@media <condition> { <rules> }
```

Apply CSS rules only if condition is satisfied

screen: page displayed on a screen

print: page is being printed

(max-width: 500px): width \leq 500px

Example

```
@media screen and (max-width: 500px) {  
  /* Small screen (e.g. a phone).  
   Make fonts smaller, adjust layout, etc. */  
}
```

Media queries in JS

`window.matchMedia(query)`

Check media query in JavaScript

Use `.matches` to check current status

Listen to change event to be notified of changes

Example

```
let m = window.matchMedia("(max-width: 500px)");
if (m.matches) console.log("Your screen is small");
m.addEventListener("change", e => {
  console.log("Is screen small?", e.matches);
});
```

Tips for mobile-friendly CSS

Flexbox can help with layouts

`flex-wrap` can wrap elements if width is small

`flex-grow` and `min-width` ensures enough space for content while filling extra space

Sometimes need different layout

Switch from row to column, use `block/inline` vs. `flex`

For even more precise control, can use `position`

Use relative units where appropriate

E.g. `width: 10em;` vs. `width: 500px;`

Allows you to adjust sizes at higher level, descendants will resize automatically

CSS animation

animation

Gradually change CSS properties

Defined entirely in CSS

Example

```
@keyframes fadein {  
  from { opacity: 0; }  
  to { opacity: 1; }  
}  
.box {  
  animation: fadein 1s;  
}
```

CSS transitions

transition

When CSS property changes, apply it gradually

Properties are changed from JS

Example

CSS:

```
transition: transform 1s;
```

JS:

```
elem.style.transform = "translateX(500px)";
```


CSS transform

transform: <op> [<op>...];

Rotate or translate the element

"Optimized for animation"--faster/smoothier than setting top, left, etc.

translate(x, y)

translateX(x)

translateY(y)

Move element

rotate(angle)

Rotate element

E.g. rotate(90deg)

Pointer events

Mouse events don't apply to touchscreens

Except for click, that one's fine

Touch events don't apply to mice

Solution: pointer events

Handles moving/dragging with mouse or on touchscreen

Pointer events

CSS: touch-action

Prevent browser from handling touch gestures

E.g. panning, zooming

Event: pointerdown

Event: pointerup

Handle when mouse/touch hold/release

Event: pointermove

Fired repeatedly as pointer moves inside element

Event: pointercancel

Dragging stops e.g. due to browser-handled gesture

Pointer events

elem.setPointerCapture(pointerId)

Send all future pointer events for a particular drag/hold to this element

elem.releasePointerCapture(pointerId)

Stop capturing pointer events

event.pointerId

Pass to the above functions

External fonts and icons

Google fonts

Library of fonts you can add to your page

Helps ensure all users see your page in same font

Choose font + styles, then add the `<link>` (or `@import`) tag

Then set font family

E.g. `:root { font-family: "Open Sans", sans-serif; }`

Multiple values define "fallback" options if loading the font fails

External fonts and icons

Material icons

(There are lots of other icon packs too)

~~Jan~~kily cleverly repurposes web fonts

```
/* In CSS */
```

```
@import "https://fonts.googleapis.com/icon?  
family=Material+Icons";
```

```
/* In HTML */
```

```
<span class="material-icons">done</span>
```

Can style these icons just like text

Aside: external frontend JS libraries

Can't import from npm

Packages from `node_modules` not served to browser

Look for packages on [jsDelivr](#) or [cdnjs](#)

Try using the "ESM" version

```
import binky from "https://...";
```

Sometimes no default export

```
import * as binky from "https://...";
```

If no exports, can sometimes still import

```
import "https://...";
```

Typically will define a global variable

Sometimes need to use a `<script>` tag