

Mobile pages and CSS animations

Michael Chang
Spring 2020

Updates

Please refer to pinned Ed post

Project due Wed 6/10

**Flexibility with remaining assignments /
milestone**

Plan for today

Making pages that work for mbile

Viewport meta, CSS media queries

CSS animations and transitions

Pointer events

Implementing gestures

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.  
    */  
}
```

Tips for obile-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

Plan for today

Making pages that work for mbile

Viewport meta, CSS media queries

CSS animations and transitions

Pointer events

Implementing gestures

Examples

animation

Gradually change CSS properties

Defined entirely in CSS

Example

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


Examples

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/smooth than setting top, left, etc.

translate(x, y)

translateX(x)

translateY(y)

Move element

rotate(angle)

Rotate element

E.g. rotate(90deg)

Plan for today

Making pages that work for mbile

Viewport meta, CSS media queries

CSS animations and transitions

Pointer events

Implementing gestures

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

Note: somewhat newer, e.g. only in Safari 13. There is a **polyfill** (written by JQuery folks, but not part of JQuery) you can use if needed. It may have slight differences.

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