

Transitions and Animations

Transition

- shows CSS property change over time
 - color/opacity/size/position/etc
- limited control over speed
- runs to completion and is done

Animation

- defines CSS property changes over time
 - color/opacity/size/position/etc
- good control over speed
- run once, N times, or infinite

Animation Accessibility

Animations (with or without transitions) can be great

- reduce change blindness
- "feel smooth"

But some people are sensitive to motion

- giving someone a migraine is not good

All transitions should be placed in media queries

- confirm `prefers-reduced-motion` is `no-preference`

I omit in slides/notes for space

Transitions

When a CSS property changes...

- opening a menu
- expanding content

...you can slow down and show the change

Transition Syntax

- `transition-property` - CSS prop(s) to animate
- `transition-duration` - how long transition takes
- `transition-timing-function` - fast/slow start/end?
- `transition-delay` - pause before starting

Or `transition` shorthand to take all 4

- minimum of property and duration

Placed in "base" selector, not "changed"

transition-property

```
transition-property: color;
```

- comma-separated list of properties to animate
- or `all`, but avoid `all`
 - performance hit
 - new properties could animate in the future

transition-duration

```
transition-duration: 1s;
```

- How long to animate to completion
- For interaction, don't be too slow

transition-timing-function

```
transition-timing-function: ease-in-out;
```

- how quick to move through progress of animation
 - `linear` - advance steadily
 - `ease` - (default) start/end slow, faster in middle
 - `ease-in` - slower start, speeds up
 - `ease-out` - starts fast, slows down
 - `ease-in-out` - like `ease`, but slower start/end
 - `cubic-bezier()` - define with MATHS
 - a few others

transition-delay

```
transition-delay: 250ms;
```

- Delay before starting transition
- Why?
 - different property transitions
 - prevent "flyby" hover effects

transition

Most common way to transition

```
transition: background-color 1s;
```

```
transition: color 1s ease-in-out 250ms;
```

```
transition: width 1s ease, height 1s ease;
```

```
transition: 1s ease;  
transition-property: width, height;
```

- Remember `prefers-reduced-motion`!
- property and duration do not have defaults

Applying Transitions

Examples will often use `:hover`

Actual usage *might* involve `:hover`

More often uses JS to apply/remove a class

- so element now matches different selectors
- triggering animation for changed properties

A few other ways, but these are most common

Animation overview

Transitions are animations but not `CSS Animations`

- **defines** CSS property changes over time
 - color/opacity/size/position/etc
 - transitions only show from existing rules
- good control over speed
 - `transition-timing-function` get complex fast
- run once, N times, or infinite
 - transitions stop at the new state
 - animations can repeat, reverse, or loop

Animation Syntax

Two parts:

- `animation` related CSS properties
 - similar to `transition`
 - more control
- `@keyframes` definition
 - defines behavior at different points of animation

Transitions animate between two states

Animations define states to animate between

Simple Animation Property

```
.target1 {  
  display: inline-block;  
  width: 3rem;  
  height: 3rem;  
  background-color: green;  
}  
  
.container1:hover .target1 {  
  animation-name: bounce; /* our defined keyframe */  
  animation-duration: 2s;  
  animation-direction: alternate;  
  animation-iteration-count: infinite;  
  /* animation: bounce 2s infinite alternate; */  
}
```

Simple Animation Keyframe

```
.container1:hover .target1 {  
  animation: bounce 2s infinite alternate;  
}  
  
@keyframes bounce {  
  from { /* starting values */  
    margin-left: 0%;  
  }  
  
  50% {  
    background-color: red; /* changing it up! */  
  }  
  
  to {  
    margin-left: calc( 100% - 3rem); /* end values */  
  }  
}
```

Animation Property Details

- Similar to `transition`
 - has `-duration`, `-timing-function`, `-delay`
- `animation-name`: name of keyframe set
- `animation-iteration-count`: number or `infinite`
- `animation-direction`: `normal`, `reverse`, `alternate`, `alternate-reverse`
- `animation-fill-values`: styles w/not animating
 - `none`: (default) As if no animation
 - other values: Use styles from a keyframe (depends on value)
- `animation-play-state`: `running` or `paused`

Keyframe details

- `from` & `to` required
- no inheritance from other `@keyframes`
- per-property if step is repeated
- animation is "smart" w/missing properties

```
@keyframes bounce { /* name is up to you! */  
  from {  
    margin-left: 0%;  
  }  
  
  50% {  
    background-color: red; /* what do you expect? */  
  }  
  
  to {  
    margin-left: calc( 100% - 3rem);  
  }  
}
```


Applying Animations

Similar to transitions:

- always use `prefers-reduced-motion` media query
- examples will use `:hover`
- actual usage *might* use `:hover`
 - might animate on load
 - might be when JS add/removes a class
 - triggers animation directly, unlike transition