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 motion 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 using `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
  - Humans get impatient in less than 1s

1s   2s

transition-timing-function

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

- Speed of progress of animation
    - `linear` - Advance steadily
    - `ease` - (default) Middle faster than start/end
    - `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

delay 0

delay 250ms

transition

- Common to accept defaults
    - `-property` and `-duration` lack defaults

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

- Setting all 4 properties

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

- Setting multiple properties

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

- Alternative way to set multiple properties

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

- Remember `prefers-reduced-motion`!

Applying Transitions

Examples will often use `:hover`

Actual usage *might* involve `:hover`

- And `focus`, `focus-within`

More often uses JS to apply/remove a class

- Element now matches different selectors
- Triggers animation for changed properties

A few other ways, but these are most common

CSS Animation overview

- Transitions are "animations", not `CSS Animations`

CSS Animations

- **Define** CSS property changes over time
    - Color/opacity/size/position/etc
    - Transitions only show from existing rules
- Good control over speed
    - `transition-timing-function` is limited
- 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

```css
.target1 {
  display: inline-block;
  width: 3rem;
  height: 3rem;
  background-color: green;
}

.container1:hover .target1 {
  animation-name: bounce; /* our defined keyframe (coming)*/
  animation-duration: 2s;
  animation-direction: alternate;
  animation-iteration-count: infinite;
  /* animation: bounce 2s infinite alternate; */
}
```

Simple Animation Keyframe

```css
.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 definition
- `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
- `animation-play-state`: `running` or `paused`

Keyframe details

- `from` and `to` required
- No inheritance from other `@keyframes`
- Per-property if step is repeated
- Animation is "smart" w/missing properties

```css
@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`/`:focus`/etc
  - Might animate on page load
  - Might be when JS add/removes a class
    - Triggers animation directly
    - Transitions triggered by *property* change

Applying Transitions to our Dropdown

- We had a menu initially "invisible"
  - Text still existed (not `display: none;`)
  - `height: 0` OR
  - `max-height: 0` OR
  - off-screen with `position`
- We made it visible on `hover`/`focus-within`
  - `(max-)height: initial`/`auto;` OR
  - `top`/`right`/`bottom`/`left: initial;`

Transition slows down change

- Instead of instant, will show steps along the way
    - Requires steps along the way

Transitioning Height

Height seems like it would have steps

- But we aren't going from 0 to a number
- We would go from height 0 to "auto"
    - Which is not a defined number
    - Can't calculate space needed

Can transition from 0 to a set height

- But set height is probably wrong

Transitioning Max-height

- Can change `max-height`
    - 0 to a given size (big enough)
    - `height` will stop at auto size
- Transition `max-height` works
    - Mostly
    - Animates from 0 to full `max-height`
        - Some of that may not be visible
            - Changing `max-height`, not `height`
- Requires given size be big enough
    - Safer than a fixed `height`
    - But not great

Transitioning top/right/bottom/left

- Transitioning position looks weird for dropdown
    - Makes sense for a "slide-in" menu
- Fixed position, so no height problem
- But often a big "offscreen" position
    - Will animate whole thing, like `max-height`
    - Weird if big chunk of transition is off-screen

Rotation of Menu

Different way to "hide" menu:

- Rotate on an axis to hide from sight
    - Ex: X axis rotates top towards/away
- At 90 degree rotation - invisible
    - "side view"

To show:

- Set rotation back to original 0 degree

Transitioning a rotation

Degree of rotation is a fixed start/end

- No confusion like with height
- But rotation defaults to rotating at center
    - Looks weird
    - We can rotate from "top"
- `transform-origin: top;`
    - Even if using `rotate` as standalone
        - Instead of using `transform` property

Rotation for slide-in menu

- Similar to dropdown
    - Just rotate on a different axis (y)
    - And `transform-origin` a different side

Rotation can look weird

- If too long orthogonal axis
- Or too slow an animation
- Can see the rotation effect
- "Slide" an understandable concept