08: Web Design

Review

Semantic HTML

CSS Selectors and Styling

CSS Box Model

Layouts with Flexbox and Grids

CSS Animations and Transitions

Adding Transitions

- Transitions are used to animate changes between CSS property values, creating a smooth shift from one state to another.
- They are triggered by a change in an element's state, such as a hover or focus.
- Commonly animated properties include opacity, color, transform, height, and width.
- Transition syntax:

```
transition: property duration timing-function delay;
```

Transition Properties

```
transition: property duration timing-function delay;
```

- **Property**: Specifies which CSS property will animate, like color, opacity, or transform.
- **Duration**: Defines how long the transition lasts, such as 1s or 500ms.
- **Timing Function**: Controls the speed of the transition, with options like ease, linear, or ease-in-out.
- **Delay**: Sets a wait time before the transition starts, such as 0s or 0.5s.

Examples of Transitions

- Hover effects, such as smooth color changes on buttons or links.
- State changes, like a gradual reveal or hide of content using opacity or height.
- Transformations, such as scaling, rotating, or moving elements for a smooth effect.

Examples of Transitions

```
button {
  background-color: blue;
  padding: 20px;
  color: white;
  border: none;
  transition: all 1s;
}

button:hover {
  background-color: green;
}
```

Walk-through

Mini Exercises

Using CSS Transitions that happen over 2 seconds:

1. Create a link using the anchor tag (<a>) and change the background color of a link element on mouse over using transitions
2. Display an image using the <img> tag, and on mouseover, rotate the image by 20 degrees (hint: use CSS Transformations)

Further Challenge:

- Create an Input box, and change the background color of the box when they're selected (use :focus pseudo-class)

CSS Animations

- Animations on the other hand allow for complex, multi-step effects by defining a series of keyframes.
- Unlike transitions, animations are not limited to state changes; they can loop or run continuously.
- Animations are set up using @keyframes, which define the start, end, and any intermediate stages of the effect.
- Syntax

```
animation: name duration timing-function delay iteration-count direction;
```

Animation Properties

- **Name**: Refers to the @keyframes animation by name.
- **Duration**: Length of one animation cycle, such as 2s.
- **Timing Function**: Controls the pacing of the animation, like ease, linear, or ease-in-out.
- **Delay**: Sets a wait time before the animation begins, such as 0s.
- **Iteration Count**: Number of times the animation repeats; infinite means continuous.
- **Direction**: Specifies how the animation plays, such as normal, reverse, or alternate.

Creating Keyframes for Animations

- Use @keyframes to define stages of the animation, such as 0% (start) and 100% (end).
- You can add intermediate steps like 50% to control the effect’s flow.

Creating Keyframes for Animations

```
@keyframes fadeIn {
 0% { opacity: 0; }
 100% { opacity: 1; }
}

button {
 animation: fadeIn 3s;
}
```

Walk-through

Mini Exercises

Using CSS Animations, define @keyframes for the following 3 animations

1. A fade-in effect (invisible at first, and slowly visible)
2. A fly effect (float from bottom-to-top *or* top-to-bottom)
3. A zoom effect (gradually grows larger or smaller)

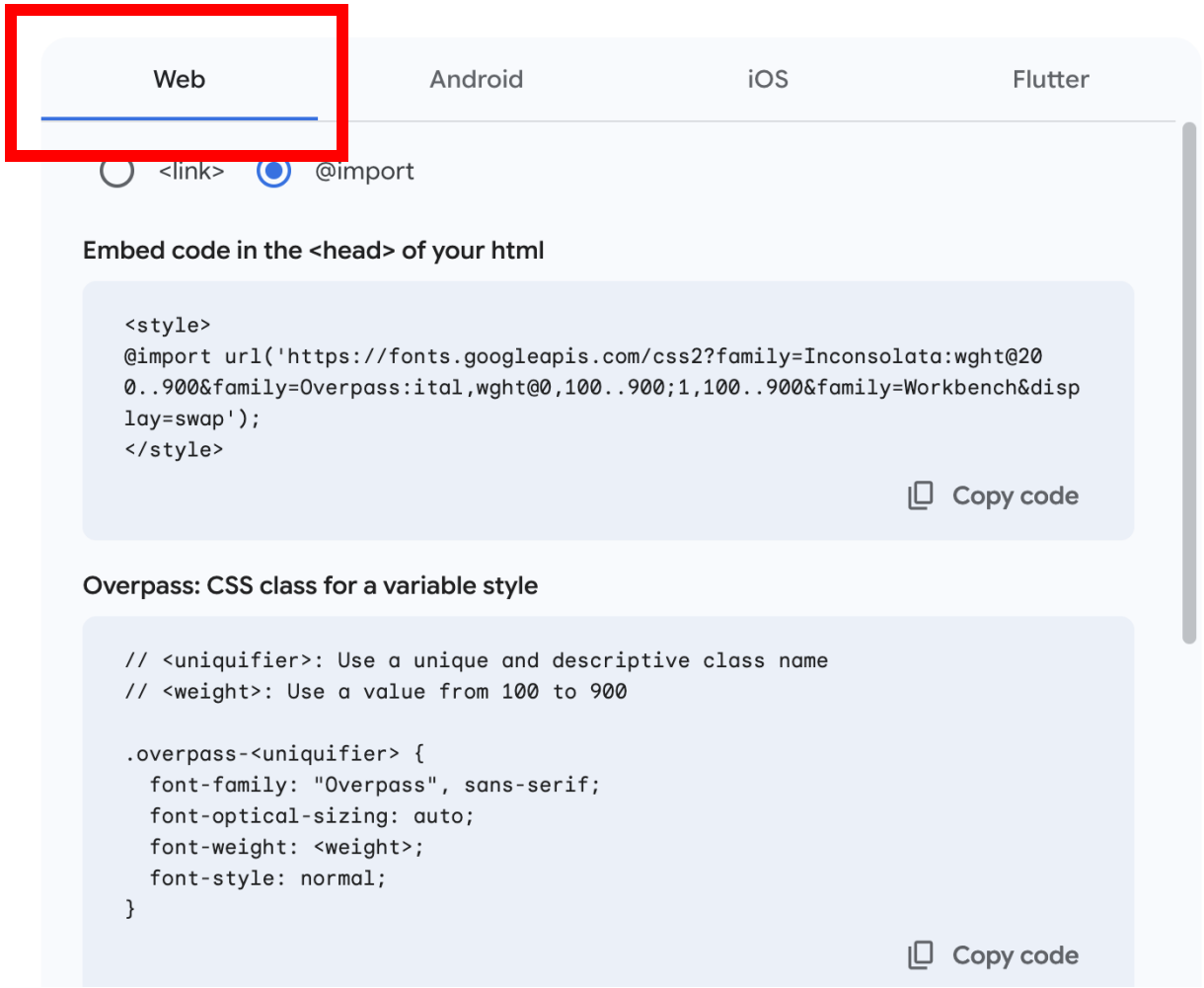
Apply these animations on any HTML elements to demonstrate

Using External Webfonts Fonts

- Collections
  - Google Fonts (free)
  - Adobe Fonts (paid)
  - Fontshare (free)

- Independent Font Foundries

Using External Webfonts Fonts



Walk-through

Lab Exercise

For your final website projects, think about:

1. Interactivity: How can animations make buttons or links feel more interactive when hovered over?

2. Visual Focus: Can you use a simple animation to highlight important content or sections?

3. Transitions: How can you animate elements like menus or images to smoothly appear or change?

4. Brand Feel: How can small animations reflect the style or mood of your site?

5. Simplicity: Are there places where animations might distract the user? How can you keep them subtle?

Examples

- https://www.awwwards.com/awwwards/collections/animation/
- https://www.awwwards.com/awwwards/collections/transitions/
- https://www.webinteractions.gallery/
- https://www.framer.com/gallery/styles/animations