Web Animations and User Experience

•••

Bau Kim

Presentation Slides

•••

goo.gl/MhoCQz

Table of Contents

- Basic CSS, Javascript and Web animation API
- Pros and cons of popular libraries
- Implementing animations for user experience

CSS Animations

Transitions and Animations

CSS Transitions

- Simple to learn
- Easy to implement
- Limited animation ability
- Most browsers supported
- Can act a little wonky

```
a {
    color: #999;
    text-decoration: none;
}
a:hover {
    color: #333;
    border-bottom: 1px dotted
       #333;
    font-size: 1.1em;
}
.fancy {
    transition: all 1s ease;
}
```

Let’s look at it in action!

•••

A Simple Transition

CSS Transitions ~

- Works on almost all CSS properties
- Different timings for different properties
- Timing styles and delays also possible
- JS event handling for transitions!

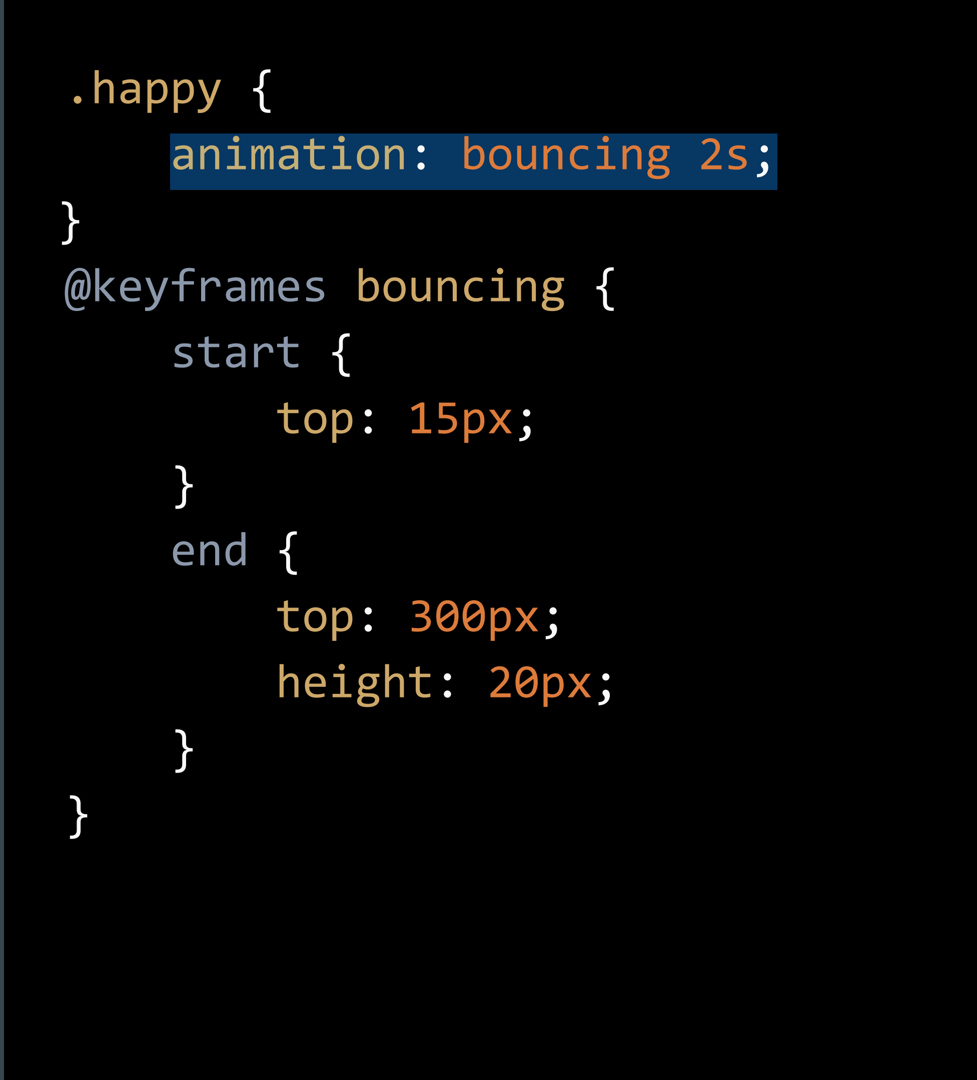
This is going to look Awesome!

...

An Advanced Transition

CSS Animations

- Almost as easy to use as transitions
- Uses CSS properties like transitions
- Intuitively set by “key frames”
- Can also be a bit wonky



```
.happy {
    animation: bouncing 2s;
}
@keyframes bouncing {
    start {
        top: 15px;
    }
    end {
        top: 300px;
        height: 20px;
    }
}
```

Let’s see something cool!

...

A Basic Animation

CSS Animations ~

- Near pinpoint accuracy with frames
- Can fine tune actions and motions
- Can play any number of times
- Can alternate direction of animation
- JS event handling available

Let’s fine tune this sucker!

An Advanced Animation

JS Animations

•••

Basics and the Animation API

JS Animations

- Very clunky and based on old technology
- Lots of setup code
- No automatic property calculations
- No key frames
- Much tighter control

```
function start() {
    id = setInterval(bounce, 20);
}

function bounce() {
    // I literally can’t show you
    // all the code because it’s
    // too much…
}
```

How bad could it be?

…

Basic JS Animation

JS Animations ~

- Works in almost all browsers
- Add any event handling at any point within the code
- Code can be used for later animations
- Does not play well with CSS

What happens when they fight?

…

Animation Breakdown

Animation API

- Best of both worlds!
- Frames like CSS
- More control like JS
- Forces consistent code between CSS and JS
- Bleeding edge technology

```
var animation = el.animate([{
        background: '#39c',
        top: '50px'
    }],
    {
        duration: 3000,
        iterations: infinity
    });
animation.play();
```

Let's look at an example!

…

Web Animation API

Web Animations API ~

- Polyfill available for backwards compatibility
- Uses JS names for CSS properties
- Easy to use, manage and copy
- Some functionalities aren’t quite ready

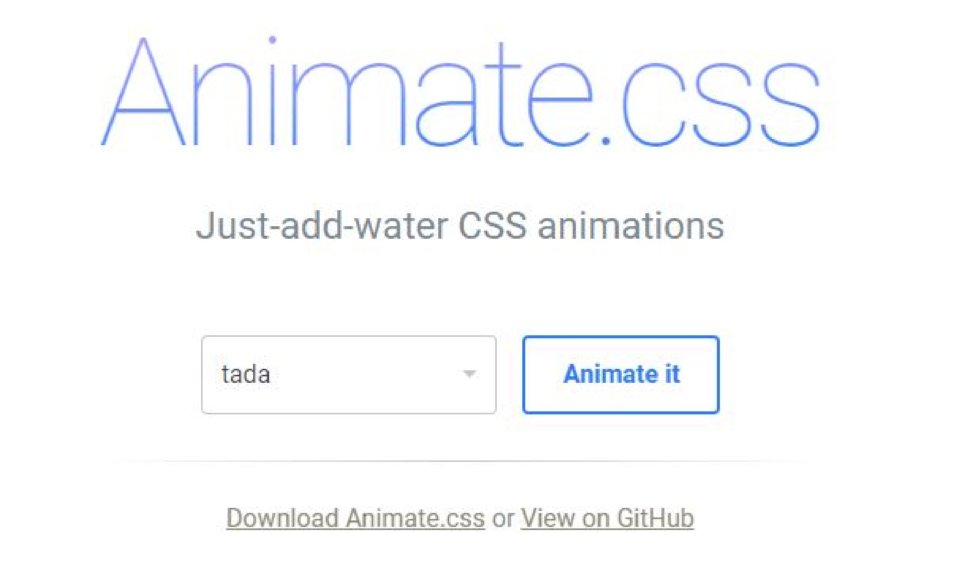
Popular Libraries

•••

The Good, the Bad, and the Useless

CSS Libraries

- Animate.css
  - Add animation using classes
- Bounce.js
  - GUI controls that print out CSS



Animate.css

Just-add-water CSS animations

Download Animate.css or View on GitHub

Bounce.js

Bounce.js is a tool and JS library that lets you create beautiful CSS3 powered animations.

Give it a spin.

JS Libraries

- Jquery UI
  - Robust UI library
- GASP
  - Very efficient animation coding
- AniJS
  - Allows custom coding

AniJS

A Library to Raise your Web Design without Coding

So which library should I use?

•••

The age old question…

It depends...

- CSS animations are easy and light but not good for backwards compatibility
- Animations API is going to be the new standard but not yet
- JS libraries if your user can handle the file size and memory needs

My Advice

- Use CSS animations for prototyping and minor animations
- Use JS libraries for big animation projects and web applications
- Keep the Animations API in your back pocket for the future

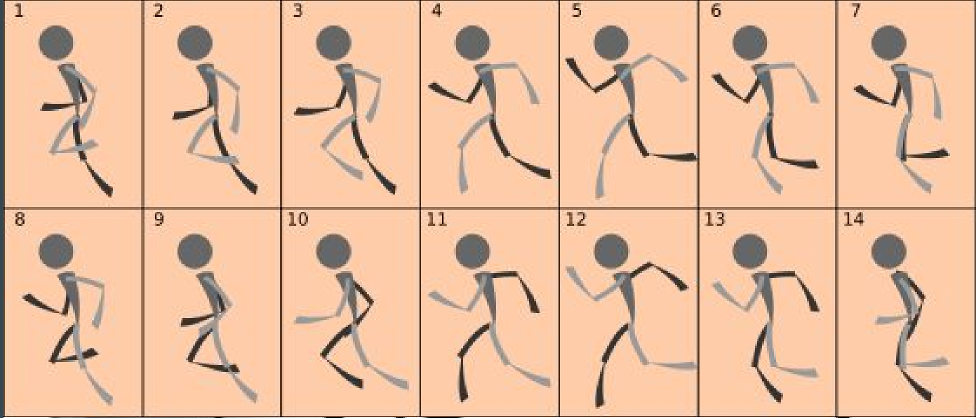
Animations and UX

•••

Theory and Guidelines

Theory

- Adding a dimension of time
- Making the binary a gradient
- Adding movement into the reactions
- More intuitive for users



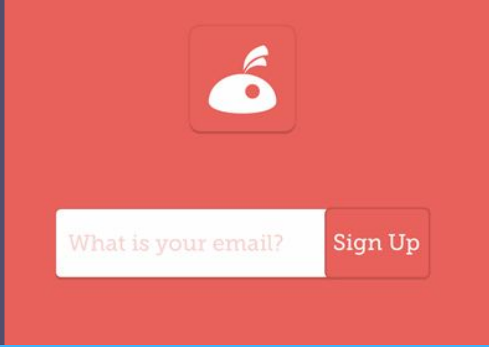
Why should I use animations?

•••

The purpose of animations

The Purpose

- Guide: Highlight important objects
- Interact: React to the users actions
- Inform: Show the user their progress
- Entertain: Have fun



When should I use Animations?

...

The Guideline

The Guideline

- Orientation: What page or step is the user on?
- Object focus: What should the user look at?
- Object reactions: How do objects act on success?
- Showing Progression: Does a user have to wait?
- Sharing Information: How do we share info?
- Making things fun and pretty: How can we make the process enjoyable?

Let’s take a look at an example!

•••

A UI with Animations

Questions!

...

goo.gl/MhoCQz