FRONT END WEB DEVELOPMENT:

LESSON 18: TRANSITIONS, MAPS & LOCALSTORAGE

JAMES WILLOCK, SOFTWARE ENGINEER
GENERAL ASSEMBLY

Objectives

- CSS transitions
  - Creating transitions between one or more property's values
- CSS animations
  - Keyframes, repetition and direction
- Google Maps
  - Plotting markers and creating popups
- LocalStorage
  - Persisting data locally

CSS TRANSITIONS

CSS Transitions

- CSS transitions allow a property or set of properties to progress between two values, over a duration and with a timing function
- Can be used for making less-jarring hover states, or even to create fairly complex interactions like tooltips without JavaScript

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

A CSS property

Seconds transition will last

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

Timing function, like ‘linear’

Seconds before transition begins

CSS Transitions: properties

- Multiple properties can be transitioned by providing the transition property with a list
- Alternatively, the all keyword in place of a property will cause all possible effects to be transitioned

CSS Transition simple example

```
a {
  transition: background-color 0.5s linear;
  background-color: green;
}

a:hover {
  background-color: red;
}
```

CSS Transition multiple effects example

```
a {
  transition: background-color 0.5s linear,
              width 1s linear;
  background-color: green;
  width: 200px;
}

a:hover {
  background-color: red;
  width: 300px;
}
```

“The <timing-function> CSS data type denotes a mathematical function that describes how fast one-dimensional values change during transitions or animations. This in essence lets you establish an acceleration curve, so that the speed of the animation can vary over its duration. These functions are often called easing functions.”

CSS transition timing functions

- There are a number of default transition timing functions available in CSS3:
  - `ease, ease-in, ease-out, ease-in-out, linear, step-start, step-end, step(n, end)`
  - Check the timing functions out on the MDN

CSS transition properties

- If you wish to transition all properties, you can use the all keyword in place of a list of properties to transition:
  - `transition: all 1s linear;`

CODE ALONG:
CSS TRANSITIONS

LOCALSTORAGE

Browser APIs

- Browser APIs let us talk to browser or device features like webcams, geolocation and storage
- Previously we needed plugins like Flash to get access to features like this, but the Browser APIs let us access them with JavaScript

LocalStorage

- LocalStorage lets you store information in the browser to retrieve and use later
- It will remain there – even if the users leaves the page. It's persistent across browser sessions
- No server / backend needed, it's all stored in the user's browser

LocalStorage limitations

- Local Storage is volatile; the user can delete the information at any time by clearing the browser cache
- Local Storage is per-browser. If you set something in Chrome, it doesn't exist in Firefox

LocalStorage API

```
// Object that does the work for us
localStorage;

// Function to save a value under a key name
localStorage.setItem(key, value);

// Function to get a value using a key name
localStorage.getItem(key);
```

Checking for support

```
if(typeof(localStorage) == ‘undefined’) {
  alert(“No LocalStorage!”);
} else {
  // do LocalStorage stuff!
}
```

LocalStorage further limitations

- Only store simple datatypes
  - Supports `string`, `number`, `boolean`, `null`
  - No `object`, or `array`
- If you need to store complex objects, use JSON encoding
  - `JSON.stringify(obj)` to encode
  - `JSON.parse(obj)` to decode

LocalStorage example

- Let's build a site that will remember our name forever (or until we empty our browser cache)
- We'll need to use form events to capture data, and the LocalStorage API to remember that data across sessions

CODE ALONG:
LOCALSTORAGE

GOOGLE MAPS

Google Maps

- The Google Maps JavaScript API is a simple way to interact with the Google Maps service via JavaScript
- The API is highly performant and very well documented
- We can mash the API up with other services to create infographics, or simply add markers to a map for contact details

CODE ALONG:
GOOGLE MAPS

HTML5 VIDEO

HTML5 video

- With the advent of HTML5, video is now a first-class citizen, and plugins like Flash or Silverlight are no longer required
- HTML5 can use the on-board graphics processors of a machine rather than the CPU, meaning no more loud fans!
- HTML5 video can be controlled easily using JavaScript

HTML5 video

- It's a good idea to provide multiple sources for your video, as not all browsers can decode the same kinds of video
- A poster image can be set for the video instead of the first frame
- The normal browser controls can be hidden entirely

HTML5 video syntax

```
<video poster="images/tron_legacy.jpg" id="tron">
  <source src="video/tron.m4v" type="video/mp4">
  <source src="video/tron.webm" type="video/webm">
</video>
```

HTML5 video JavaScript methods

Name	Method	Notes
Play	`.play()`	Plays the video
Pause	`.pause()`	Pauses the video
Volume	`.volume`	Retrieves or sets the volume of the video – a floating point number between 0 and 1
Current time	`.currentTime`	Retrieves or sets the current position of the video in seconds

HTML5 video syntax

```
var video = document.getElementById('my_video');

video.play(); // Plays video

video.volume += 0.1; // Notch volume up

video.currentTime = 20; // Seeeks to 20 seconds
```

CODE ALONG:
HTML5 VIDEO

THANK YOU

- james.willock@generalassemb.ly
- @niceguyjames