JavaScript APIs

Fresno Summer Bootcamp 2015

Code

bit.ly/fresno-4-code

Slides

bit.ly/fresno-4-slides

DOM manipulation

What is DOM?

- The DOM is basically an API that represents and interacts with objects in HTML documents. The nodes of every document are organized in a tree structure, called the DOM tree. Objects in the DOM tree may be addressed and manipulated by using methods on the objects.


jQuery
write less, do more.

Exercise

Convert jQuery selectors to plain JavaScript

Audio

Audio methods

- **load()**
  - *Re-loads the audio/video element*
- **play()**
  - *Starts playing the audio/video*
- **pause()**
  - *Pauses the currently playing audio/video*

Audio properties

- **currentTime**
  - *Sets or returns the current playback position in the audio/ video (in seconds)*
- **duration**
  - *Returns the length of the current audio/video (in seconds)*
- ***volume***
  - *Sets or returns the volume of the audio/video*

Audio properties

- **played**
  - *Returns a TimeRanges object representing the played parts of the audio/video*
- **paused**
  - *Returns whether the audio/video is paused or not*
- **muted**
  - *Sets or returns whether the audio/video is muted or not*

Exercise

Create an audio player with
Play, Pause, Volume Up, Volume Down buttons
without using natives controls

Video

Video events

- **abort**
  - *Fires when the loading of an audio/video is aborted*
- **loadedmetadata**
  - *Fires when the browser has loaded meta data for the audio/video*
- **progress**
  - *Fires when the browser is downloading the audio/video*

Video events

- **seeking**
  - *Fires when the user starts moving/skipping to a new position in the audio/video*
- **timeupdate**
  - *Fires when the current playback position has changed*
- **volumechange**
  - *Fires when the volume has been changed*

Exercise

Create a video player with
Play, Pause, Replay buttons,
Current time, Duration time
without using natives controls

Form validation

Validation-related attributes

- pattern
- min
- max
- required
- step
- maxlength

Validation-related styling

- The **:required** and **:optional** pseudo-classes allow writing selectors that match form elements that have the required attribute, or that don't have it.

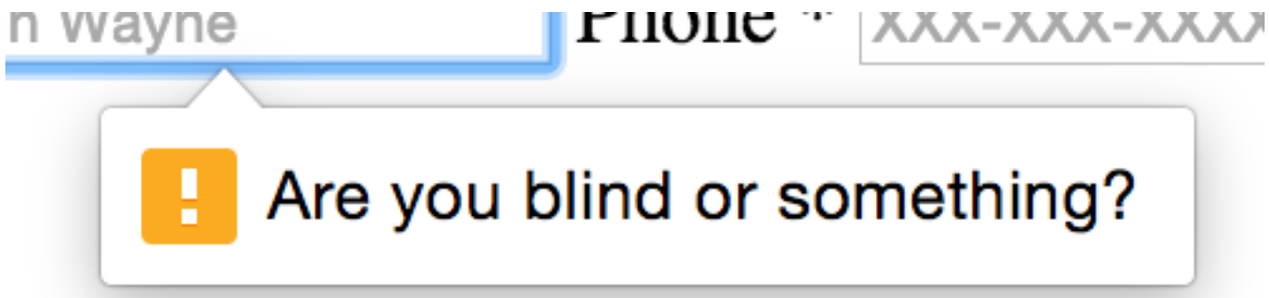
- The **:valid** and **:invalid** pseudo-classes are used to represent <input> elements whose content validates and fails to validate respectively according to the input's type setting.

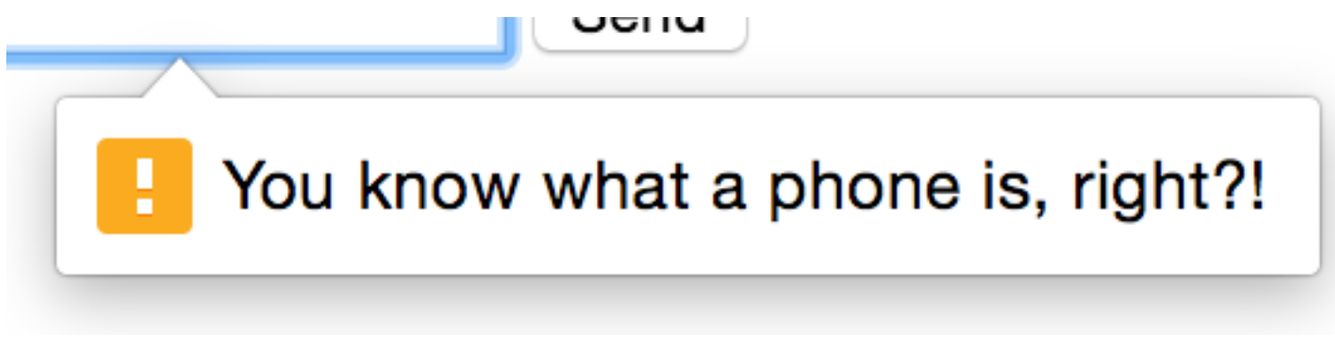
Controlling the validation text

- The **element.setCustomValidity(error)** method is used to set a custom error message to be displayed when a form is submitted. The method works by taking a string parameter error.

Exercise

Set custom validation messages and styling
for name and phone fields


n Wayne
Phone *
xxx-xxx-xxxx
Are you blind or something?


Send
You know what a phone is, right?!

Web Storage

Web Storage APIs

- localStorage
- sessionStorage

Web Storage APIs

- localStorage.length
- localStorage.key(0)
- localStorage.removeItem()
- localStorage.clear()

Exercise

Don’t let the user lose his/her diary story
even if browser closes

How was your day?

Dear Diary, What is it with that Mary girl? Dragging me to school every day. As if I had a choice. What you don't hear in those nursery rhymes is that she starves me if I don't go to school with her; it's the only way I can stay alive! I'm thinking about being adopted by Little Bo Peep, sure I may get lost, but anything is better than being with Mary and those little brats at school (shudder, shudder).

Exercise

Don’t let the user lose his/her todo list
even if computer turns off

To-do list

1. Buy milk
2. Go to Fresno
3. Learn HTML5

Clear

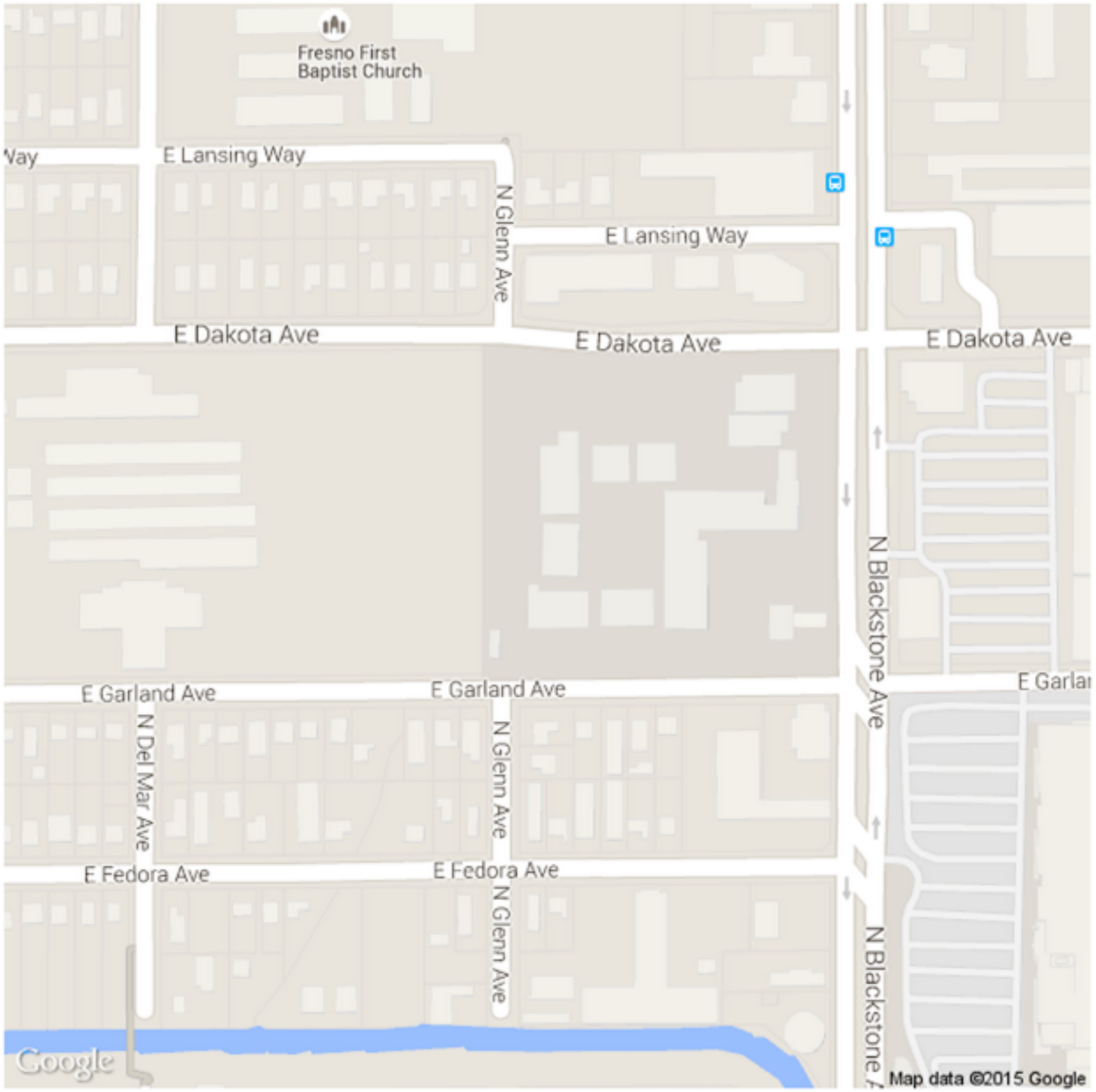
Geolocation

Geolocation

- **navigator.geolocation**
  - *The geolocation API is published through this object. If the object exists, geolocation services are available.*
- **navigator.geolocation.getCurrentPosition()**
  - *Initiates an asynchronous request to detect the user's position, and queries the positioning hardware to get up-to-date information.*

Exercise

Get user’s latitude and longitude,
then plot it on a Google Map


Fresno First
Baptist Church
Way
E Lansing Way
N Glenn Ave
E Lansing Way
E Dakota Ave
E Dakota Ave
E Dakota Ave
N Blackstone Ave
E Garland Ave
E Garland Ave
N Del Mar Ave
N Glenn Ave
E Fedora Ave
E Fedora Ave
N Glenn Ave
N Blackstone Ave
Google
Map data ©2015 Google

User Media

User Media

- **navigator.getUserMedia(options, success, error)**
  - *Prompts the user for permission to use one video and/or one audio input device such as a camera or screensharing and/or a microphone.*

User Media

- **navigator.getUserMedia(options, success, error)**
  - *Prompts the user for permission to use one video and/or one audio input device such as a camera or screensharing and/or a microphone.*

Exercise

Fetch webcam stream and put it on a video

Drag & Drop

Drag events

- **dragstart**
  - *Fired on an element when a drag is started.*
- ***drag***
  - *Fired on an element that is being dragged.*
- ***dragend***
  - *Drag operation is complete, whether it was successful or not.*

Drag events

- **dragenter**
  - *Fired when the mouse enters an element while a drag is occurring.*
- **dragleave**
  - *This event is fired when the mouse leaves an element while a drag is occurring.*

Drag events

- **dragover**
  - *This event is fired as the mouse is moving over an element when a drag is occurring.*
- **drop**
  - *The drop event is fired on the element where the drop occurred at the end of the drag operation*

Exercise

Make it possible to drag cards to other box

Canvas

Drawing rectangles

- **fillRect(x, y, width, height)**
  - *Draws a filled rectangle.*
- **strokeRect(x, y, width, height)**
  - *Draws a rectangular outline.*
- **clearRect(x, y, width, height)**
  - *Clears the specified rectangular area, making it fully transparent.*

Drawing lines

- **moveTo(x, y)**
  - *Moves the pen to the coordinates specified by x and y.*
- **lineTo(x, y)**
  - *Draws a line from the current drawing position to the position specified by x and y.*

Painting paths

- **stroke()**
  - *Draws the shape by stroking its outline.*
- **fill()**
  - *Draws a solid shape by filling the path's content area.*

Exercise

Draw a squared face

Drawing arcs

- **arc(x, y, radius, startAngle, endAngle, anticlockwise)**
  - *Draws an arc which is centered at* ***(x, y)*** *position with* ***radius r*** *starting at* ***startAngle*** *and ending at* ***endAngle*** *going in the given direction indicated by* ***anticlockwise*** *(defaulting to* ***clockwise****).*

Exercise

Draw a rounded face