Web Development

COMP 431 / COMP 531

Lecture 2: HTML and Forms

Instructor: Mack Joyner

Department of Computer Science, Rice University

mjoyner@rice.edu

http://www.clear.rice.edu/comp431

Recap

- Administration
- Office Hours
- HTML

Homework Assignment 1
(Simple Page)
Due Thu 8/31 by 11:59 PM

Editors: I use Atom

app.js — ~/comp431/webdev/F17

Project
- ansi-styles
- anymatch
  - index.js
  - LICENSE
  - package.json
  - README.md
- archive-type
- archy
- argparse
- arr-diff
- arr-flatten
- array-differ
- array-each
- array-find-index
- array-slice
- array-union
- array-uniq
- array-unique
- arraybuffer.slice
- asn1

ex1.html | app.js

```
                    })
                        .reduce(function(l,r) { return l.concat(r) })
                    )
            else
                return 0;
        }

        console.log('loaded up with ', vm.params)

        function setAssignment(id) {
            if (!id) {
                vm.assignmentName = 'General Info';
                vm.assignmentDue = undefined;
                vm.assignmentId = undefined;
                vm.url = undefined;
                vm.rubric = undefined;
                vm.showRubric = (vm.assignmentName != 'General Info')
                return;
            }
            var a = vm.srv.getAssignment(id);
            if (a == undefined) {
                id = 'simple';
                a = vm.srv.getAssignment(id);
```

~/comp431/webdev/F17/app.js 82:1 LF UTF-8 JavaScript

inclass-1: hello.html

Good practice to include "lang" and "charset"

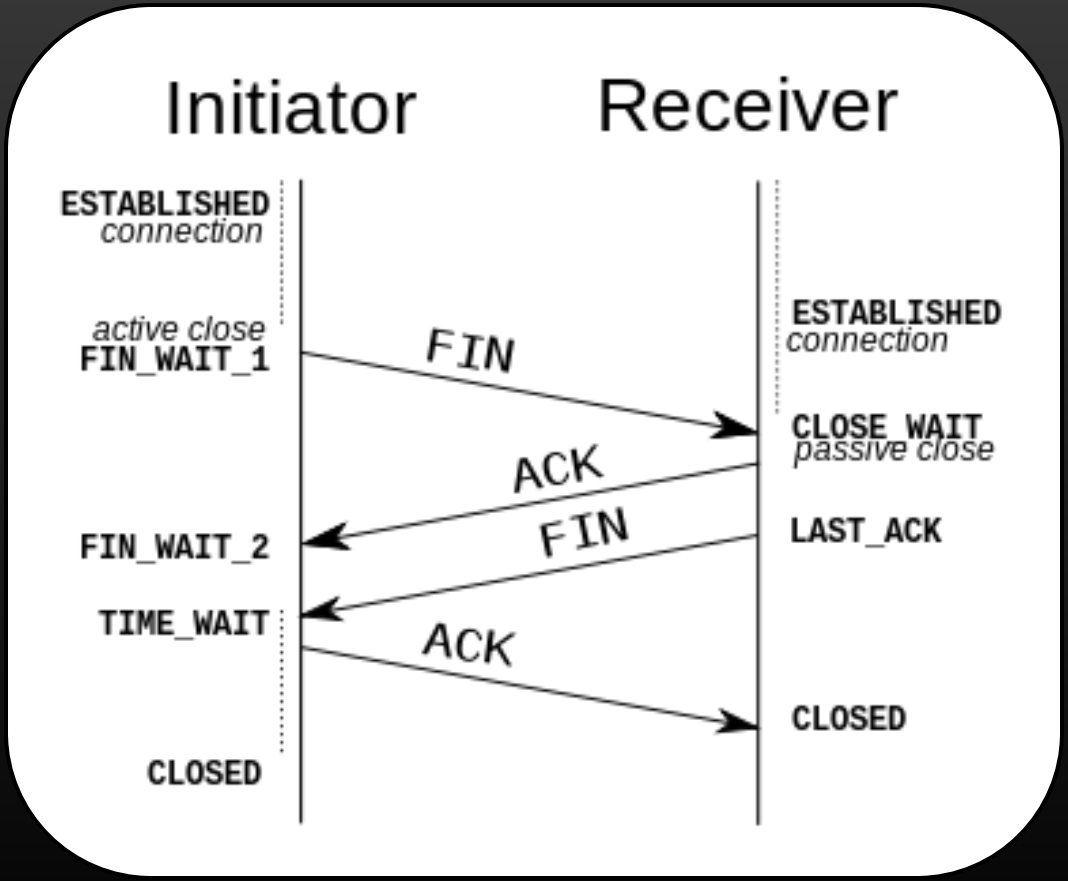
Mandatory to include DOCTYPE !

```
<!DOCTYPE html>
<html lang="en-us">
  <head>
    <meta charset="utf-8">
    <meta name="author" content="Mack Joyner">
```

No spaces between attribute and value

Transmission Control Protocol

- Clients (initiator) and Servers (receiver)
- Resilient
- http (https) is *generally* routed on port 80 (443)
- https is "HTTP over TLS" (transport layer security)
  - successor to SSL (secure sockets layer)
  - No good reason *not* to use it…



Hypertext Transfer Protocol (HTTP)

- 1965 – hypertext
- 1989 – WWW
- 1991 – HTTP V0.9
- 1996 – HTTP V1.0
- Mid-1996 – HTTP/1.1
- May 2015 – HTTP/2
  (0.4% websites support 7/2015)
  (Likely TLS only)
  (server push)
  (parallel loading)

- Request-response protocol
- Client sends request
- Server replies with response
  - Typically returning a resource

Status Codes

- Informational 1XX
- Successful 2XX
  - 200 = OK
  - 201 = Created
  - 202 = Accepted
  - 204 No Content
- Redirection 3XX
  - 301 = Moved Permanently
  - 302 = Found
- Client Error 4XX
  - 400 = Bad Request
  - 401 = Unauthorized
  - 403 = Forbidden
  - 404 = Not Found
- Server Error 5XX
  - 500 Internal Server Error
  - 501 Not Implemented

Where is JavaScript?

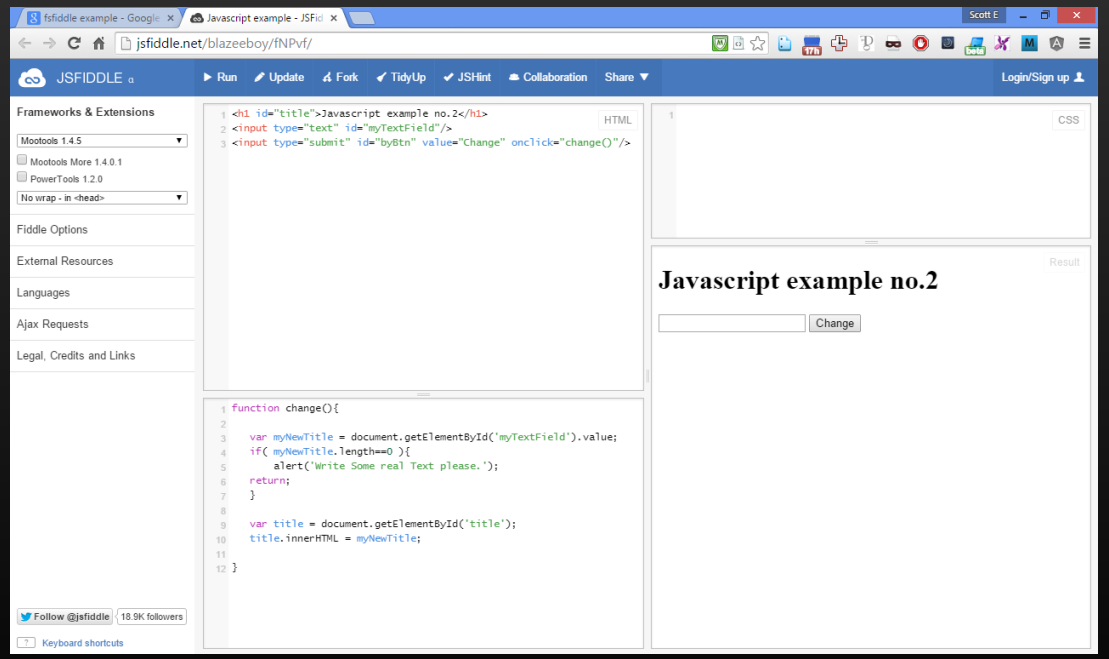
- In your browser
- Add a script to a HTML page and view in browser

JavaScript does not necessarily play well through file://

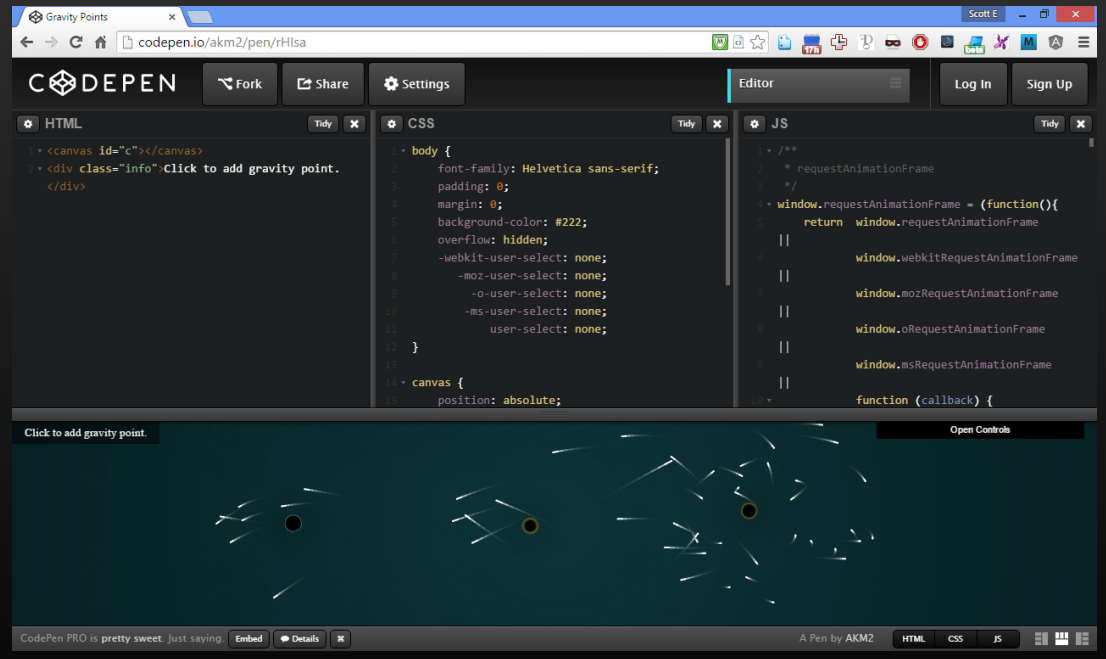
…

Online Frontend Playgrounds

- JavaScript can evaluate JavaScript
- Therefore we can easily create an online JavaScript environment
  (or use someone else's)



jsfiddle.net



codepen.io

Chrome Developer Tools

Developer Tools - http://www.rice.edu/

Elements Network Sources Timeline Profiles Resources Audits Console Page Speed AngularJS

Sources Content scripts Snippets

www.rice.edu
- _css
- _fonts/open_sans/regular
- _scripts
  - jquery-ui
  - media-element
  - carousels2.js
  - owl-carousel.min.js
  - tools5.js
  - slick
  - (index)
- (no domain)
- aadgmnobpdmgmigaicncghmmc
- ajax.googleapis.com
- api.filepicker.io
- api.keen.io
- cdn.segment.io
- d26b395fwzu5fz.cloudfront.net
- d36hc0p18k1aoc.cloudfront.net
- d36hc0p18k1aoc.cloudfront.net
- d3l7tj34e9fc43.cloudfront.net
- dialog.filepicker.io

carousels2.js × owl-carousel.min.js tools5.js

```
1  // JavaScript Document
2      $(function(){
3          // $('.owl-prev, .owl-next').css('display','none');
4          $('.owl-prev').addClass('fa fa-chevron-left').html('');
5          $('.owl-next').addClass('fa fa-chevron-right').html('');
6          // $('.owl-carousel,.fa-chevron-right,.fa-chevron-left').mouseen
7          //   $('.fa-chevron-left, .fa-chevron-right').fadeIn(1000);
8          // });
9          // $('.owl-carousel').mouseleave(function(){
10         //   $('.fa-chevron-left, .fa-chevron-right').fadeOut(2000);
11         // });
12     // $('.owl-carousel,.fa-chevron-right,.fa-chevron-left').hoverIntent
13     //   function(){
14     //   $('.fa-chevron-left,.fa-chevron-right').fadeIn(800);
15     // },
16     //   function(){
17     //   $('.fa-chevron-left,.fa-chevron-right').fadeOut(1200);
18     //   });
19
20     });
21
22 // Slick Carousel for Top Stories
23 $(document).ready(function(){
24   $('.owl-carouselTop').slick({
25   lazyLoad: 'ondemand',
26   dots: true,
27   infinite: true,
28   speed: 300
29
```

{} Line 1, Column 1

Watch
Call Stack — Async
Not Paused
Scope
Not Paused
Breakpoints
No Breakpoints
DOM Breakpoints
XHR Breakpoints
Event Listener Breakpoints

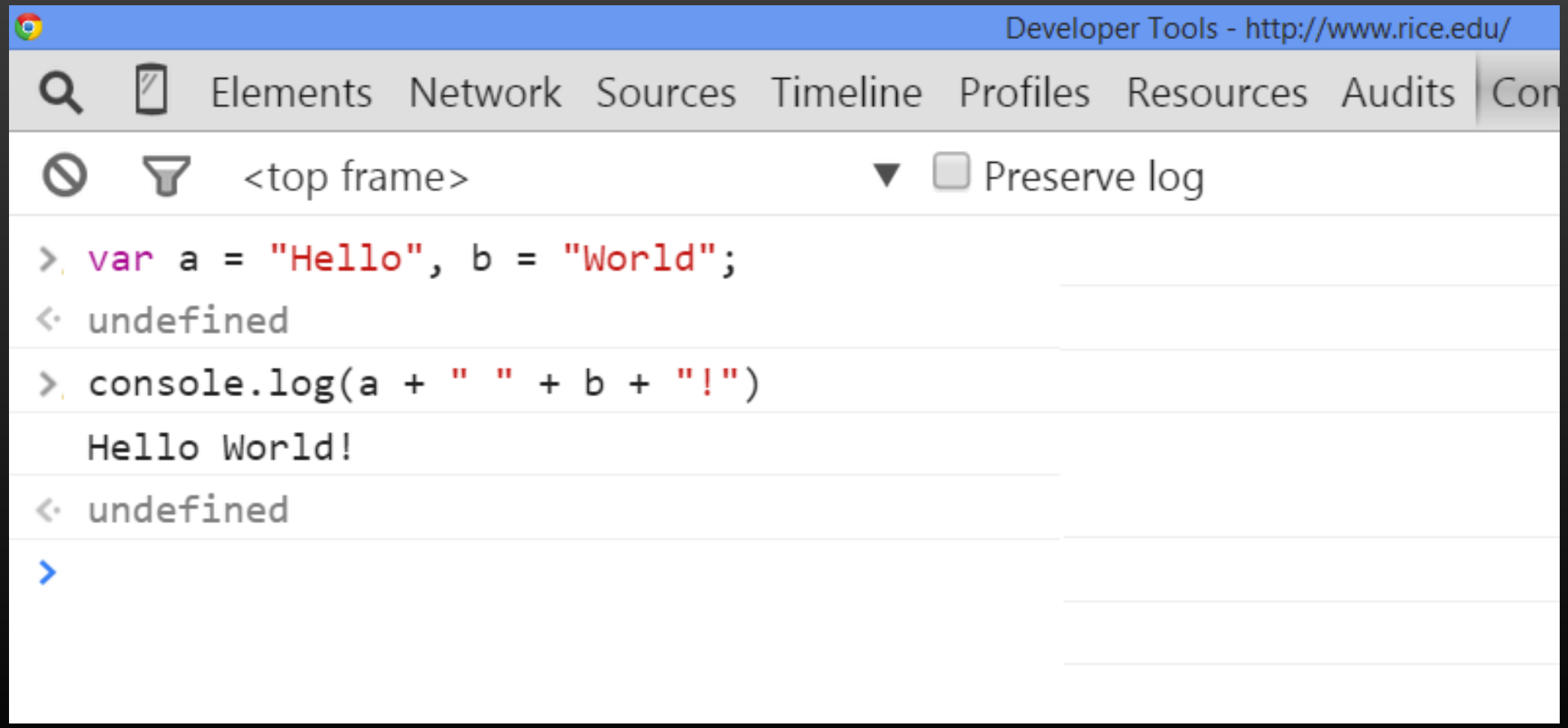
Console Search Emulation Rendering

<top frame> Preserve log

```
GET chrome-extension://fmfcbgogabcbclcofgocippekhfcmgfj/cast_sender.js net::ERR_FAILED   www-embed-player-new.js:151
GET chrome-extension://enhhojjnijigcajfphajepfemndkmdlo/cast_sender.js net::ERR_FAILED   www-embed-player-new.js:151
GET chrome-extension://eojlgccfgnjlphjnlopmadngcgmmdgpk/cast_sender.js net::ERR_FAILED   www-embed-player-new.js:151
```

FYI: You may need to enable Developer mode to get access to the console in Safari

Hello World in console



Script tag

- The `<script>` tag defines a client-side script
- Previously needed a type attribute

```
type="text/javascript"
```

- Either has a source attribute

```
src="https://ajax.googleapis.com/ajax/libs/
jquery/2.1.3/jquery.min.js"
```

That's a line wrap from powerpoint

or content

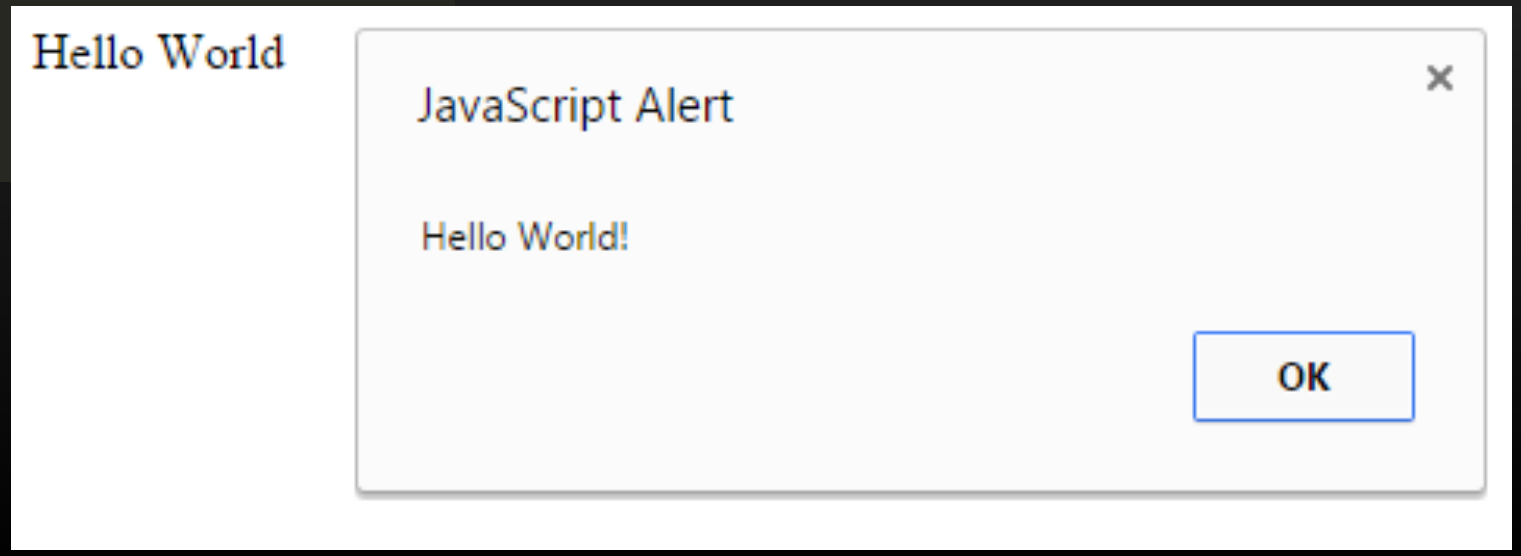
```
<script>
  window.alert("Hello World!");
</script>
```

Hello world in a page

```
1  <html>
2  <body>
3  Hello World
4  </body>
5  <script>
6    window.alert("Hello World!");
7  </script>
8  </html>
```

The "window" object is provided by the engine

Other built-ins include "navigator" and "document"



Query Parameters

- Can be attached to any URL

http://somesite.com:8080/example/page.php?q=5&b=8#someAnchor

- Most often used with GET requests

```
> location
<· ▶Location {ancestorOrigins: DOMStringList, origin: "file://", hash: "",
  search: "?arg=value&second=string&number=15", pathname:
> location.search
<· "?arg=value&second=string&number=15"
>
```

Forms

- The work horse of many a web page
- Typically used with POST requests
  - POST allows for a larger payload
- Contains `<input>` and `<button>` children

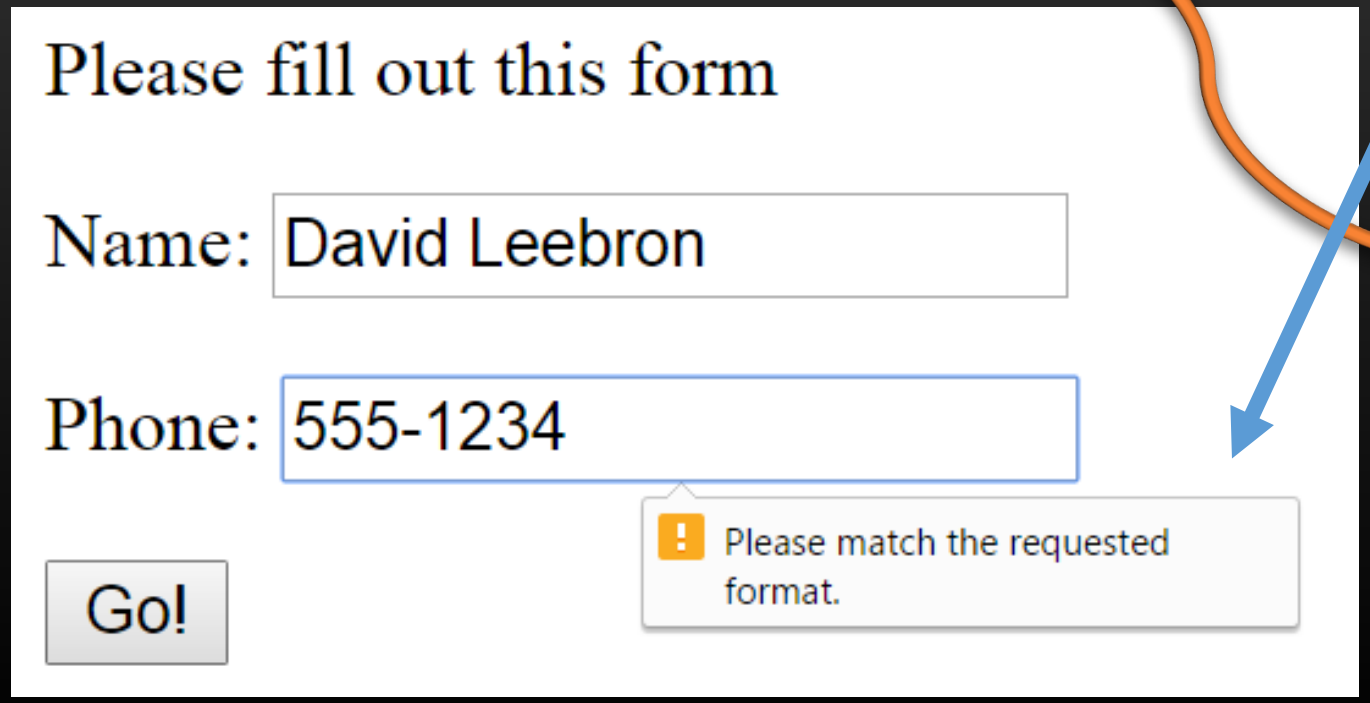
```
<form
  method=("" | "GET" | "POST")
  action=<url>
  id="myForm" >

  <input ... >
</form>
```

`id` used for referencing (`name` is deprecated)

Form contents

```
4  <form id="myForm" method="" action="#">
5      <p>Name: <input type="text" name="name"></p>
6      <p>Phone: <input type="tel" name="phone" pattern='\d\d\d'></p>
7      <input type="submit" value="Go!">
8  </form>
```



button?

Input Type

The Basics

- text
- password
- checkbox
- radio
- file
- submit
- reset

HTML5

- search
- email
- url
- tel
- number
- range
- date
- color
- ...

GET vs POST

GET

- Parameters in URL
- Used to get data
- *Should* have no side-effects
- URL limit of 2083 characters (IE)
- Browser can cache

POST

- Parameters in request body
- Used to update server
- May update the server (watchout for repeated requests)
- In principle no limit to payload size
- Shouldn't be cached

The GET gotcha

GET *should* have **NO** side-effects

- GET is ***so*** easy
- Parameters sent in URL is sweet and simple
- How about a delete button?
  - For example it might GET /delete?id=2

- Browser plugin: Google Web Accelerator
  - Scans a page and executes all GETs to cache
  - This is great because it speeds up our surfing experience
  - This is bad because we just hit all of those delete links… oops!

Form Submission (GET)

- Properly fill out form and click the button

/form-example.html?name=David+Leebron&phone=123

Form Submission POST

```
<form id="myForm" method="POST" action="#">
```

× Headers Preview Response Timing

▼ **General**

Request URL: /form-example.html?name=David+Leebron&phone=123

▶ **Request Headers (5)**

▼ **Query String Parameters** view source view URL encoded

name: David Leebron

phone: 123

▼ **Form Data** view source view URL encoded

name: Try a POST

phone: 987

Hidden Fields

- Can be useful to store extra data that is sent with form, e.g., session id

```
11  <form id="myForm2" method="" action="#">
12      <p>Name: <input type="text" name="name"></
13      <p>Phone: <input type="tel" name="phone" pattern="\d\d\d"></p>
14      <p><input type="hidden" name="secret" value="message"></p>
15      <input type="submit" value="Go!">
16  </form>
```

Please fill out this form

Name:

Phone:

Go!

Name: This One

Phone: 456

Go!

▼ **Query String Parameters**

name: This One

phone: 456

secret: message

Aside on Date

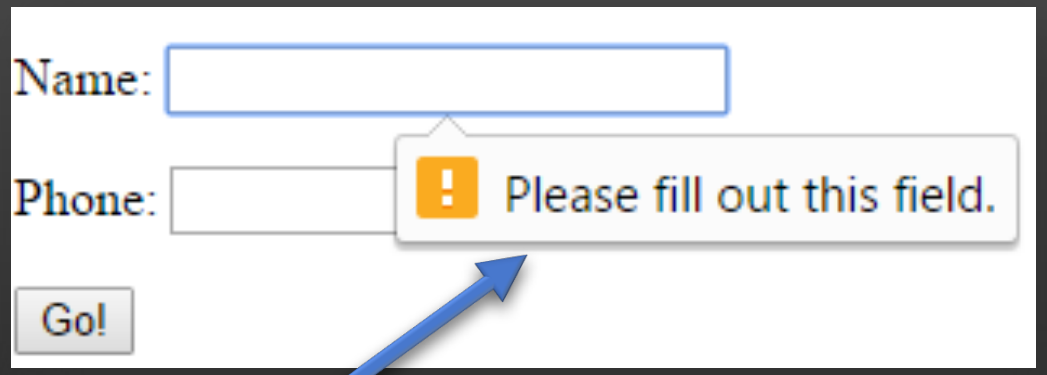
- JavaScript has numerous built-ins
- Date object is modelled after JDK1.0 java.util.Date

```
> Date.now()
<· 1440354508583
> new Date(Date.now())
<· Sun Aug 23 2015 13:28:35 GMT-0500 (Central Daylight Time)
> new Date(1440000000000)
<· Wed Aug 19 2015 11:00:00 GMT-0500 (Central Daylight Time)
```

Form Validation

- … the bane of JavaScript?
- Why do we need it?

Form Validation



- HTML5 has some built-in validation
  - But it's not everything we'd ever want

```
<input type="text" name="name" required></p>
<input type="tel" name="phone" pattern='\d\d\d'>
```

- We pre-validate before the built-in HTML5 validation using an onclick event handler attached to the submit button

```
    <input type="submit" value="Go!" onclick="return doSomething()">
</form>
<script>
function doSomething() {
    window.alert('something');
    return false;
```

onclick truthy says to submit form or not
We'll talk more about events later

Form Validation Example

```
onclick="return doSomethingBetter(this.parent)"
```

```
12        <input type="submit" value="Go!" onclick="return doSomethingBetter()">
13      </form>
14
15      <script>
16        function doSomethingBetter(form) {
17          if (!form) {
18            //find the right one?
19            var allFormsAsArray = document.forms;
20            //better to get by id
21            form = document.getElementById("myFormValidated");
22          }
23          console.log(form);
24
25          return (form.name.value === "Mack"
26                  && form.phone.value == 123);
27        }
28      </script>
```

Encoding…

- How about

Name: This is a test?

▼ General

Request URL: /form-example.html?name=This+is+a+test%3F&phone=123

▶ Request Headers (3)

▼ Query String Parameters view source view URL encoded

name: This is a test?

phone: 123

```
> decodeURIComponent(location.search.substring(1).split('&')[0].split('=')[1])
<· "This+is+a+test?"
> encodeURIComponent('a+b')
<· "a%2Bb"
```