



Introduction to JavaScript

INFO/CS 2300:
Intermediate Web Design and
Programming

Project 1 grading

- Logistical issues –Xiying xw282@cornell.edu
- Regrades:
 - Fill out form from Piazza
 - Meet with me
 - Entire project may be regraded (e.g. grade could go up or down).
 - Regrade request must be within a week from day grades released.
- “Best of”: Forum posts with examples of good projects.
 - Will be contacting students to ask permission before posting

Project 2 – Online Catalog

- More PHP practice
- Form entry and validating user input
- Reading from and writing to a file
- Searching an array
- CSS in separate file - not inline
- Full project description is posted on Piazza
- This is a significantly bigger project

Planning your time

Generally you won't need Monday's lecture topic for the assignment due the next day

Test your critical work on the server early and then do your fine tuning – have something working by Sunday then make it better

Validating helps you catch errors

Debugging Tip: Manage Caching

Versioning your CSS and JS files

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>No Cache Style</title>
```

```
<?php
```

```
    $style_path = 'css/style.css';
```


```
    $version = filemtime( $style_path);
```

```
    echo "<link rel='stylesheet'
```

```
        href='$style_path?ver=$version'>";
```

```
    ?>
```

```
</head>
```



version automatically
updates when you
modify style.css

Mini Crash Courses

This Friday on XAMPP

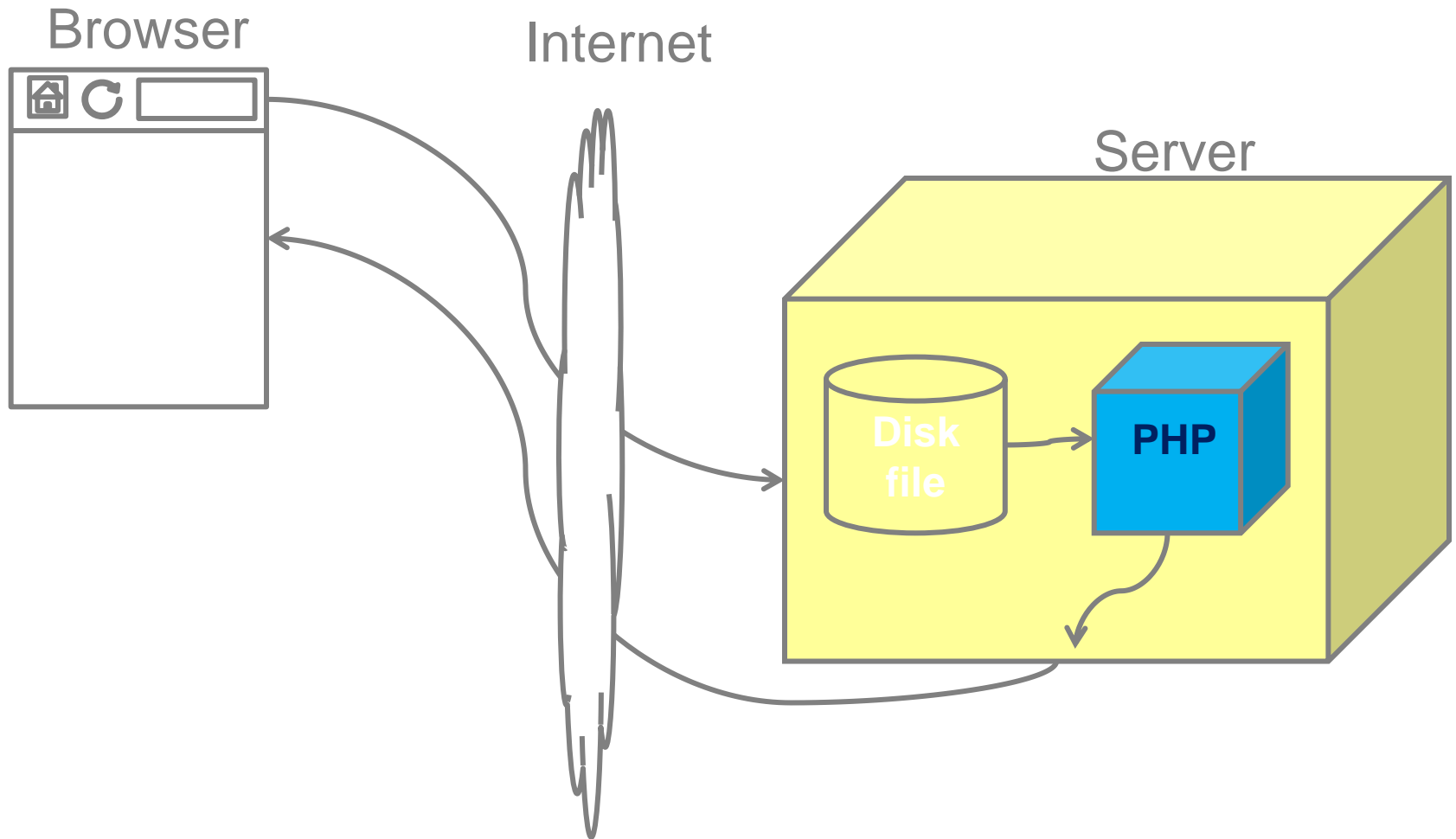
Next Wednesday: Debugging

Details on Piazza

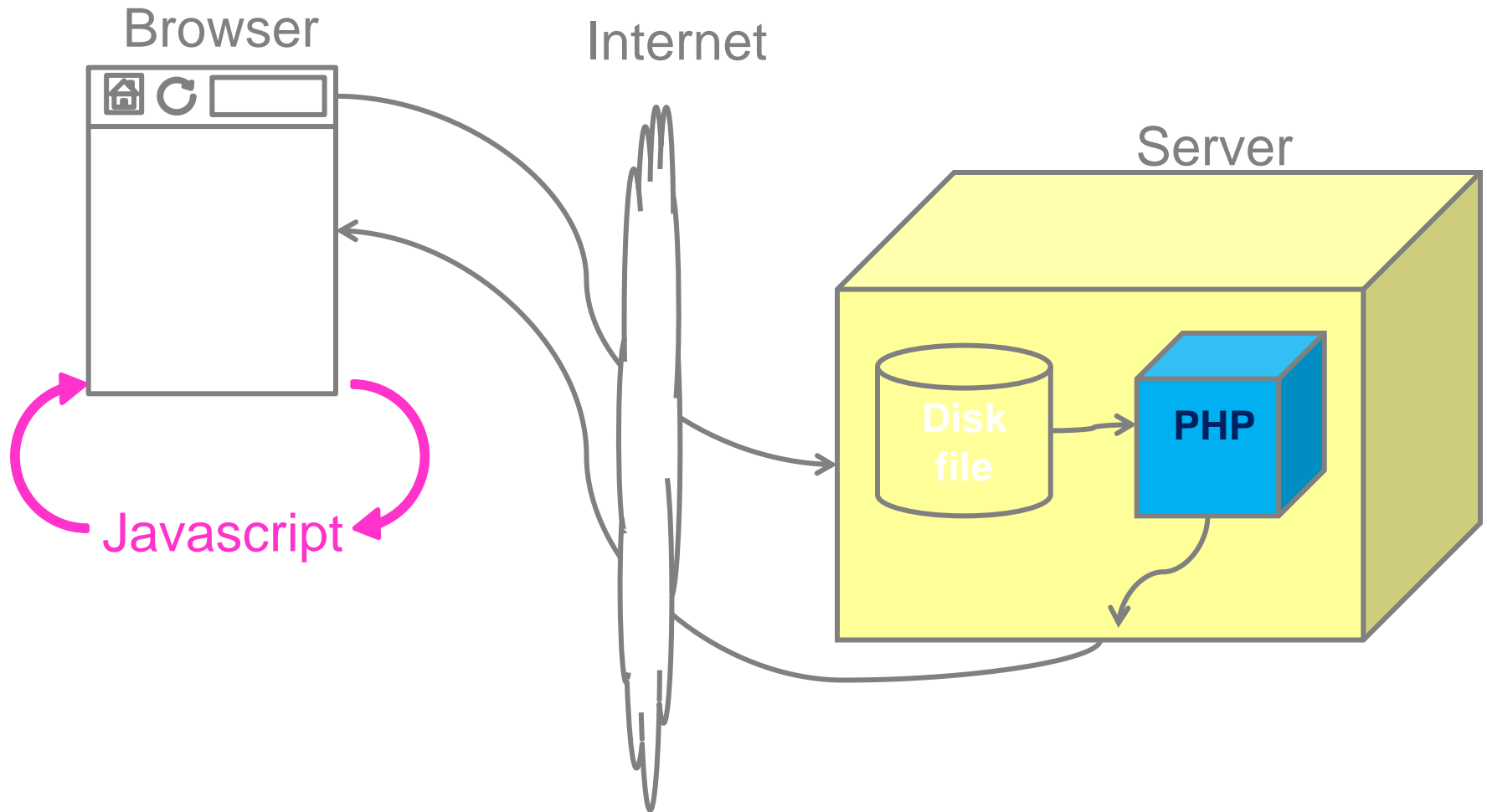
JavaScript

JavaScript is a programming language used in web programming, like PHP, but with some major differences.

The web with PHP



and Javascript



PHP vs. JavaScript

PHP: Runs on the server
 User can't see the code



For this course

JavaScript: Runs on the browser
 User might disable JS
 Code is visible to user
 Good for user interface
 Sometimes can reduce server load

JavaScript document.write

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="UTF-8">
```

```
    <title>JavaScript Document Write</title>
```

```
    <script type="text/javascript">
```

```
      document.write( "Hello world!" );
```

```
    </script>
```

```
  </head>
```

```
  <body>
```

```
  </body>
```

```
</html>
```

One way to include
JavaScript: Inline

body starts empty but
document.write fills it in



JavaScript alert

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>JavaScript Alert</title>
```

```
    <script type="text/javascript">
```

```
      alert("Hello world!");
```

```
    </script>
```

```
  </head>
```

```
  <body>
```

```
  </body>
```

```
</html>
```




Pop up box

JavaScript window.open

```
<!DOCTYPE html >
<html>
  <head>
    <meta charset="UTF-8">
    <title>Fun with JavaScript</title>
  </head>

  <body>
    <script type="text/javascript">
      window.open("helloworld.html","", "");
    </script>
  </body>
</html>
```



opens a new window



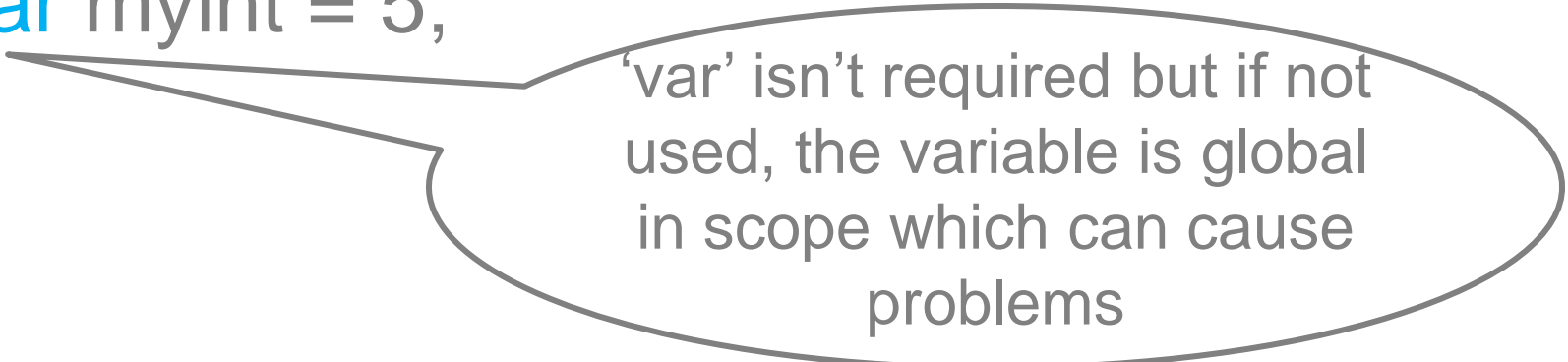
Comparing with PHP

Variables

Not prefixed by '\$' in JavaScript. Usually declare by 'var' when variable is first created.

```
var mystring='cat';
```

```
var myint = 5;
```



'var' isn't required but if not used, the variable is global in scope which can cause problems

Conditionals

Mostly the same, but no 'elseif'.

```
if (mystring === "happy") {  
    ...  
} else if (mystring === "sad") {  
    ...  
}
```


Equal and Identical

Same as PHP

== does a type conversion before
comparison

=== type must be the same to evaluate to
true

Loops

Same as PHP

```
for ( i = 0; i < 10; i++ ) {
```

```
    ...
```

```
}
```

```
While ( i < 10 ) {
```

```
    ...
```

```
}
```

Arrays

Capitalize “Array”

```
var menu = Array( “Blue”, “Green”, “Red” );
```

```
document.write( menu[2] );
```



Red

```
menu[ ‘Home’ ] = “index.html”;
```



Associative array

Enumerating arrays

Significant syntax differences:

PHP

```
foreach ($myarray as $index => $item) {  
    print( $item );  
}
```

In JavaScript we write:

```
for ( index in myarray ) {  
    document.write( myarray[ index ] );  
}
```

Functions

Same as PHP.

```
function greeting( name ) {  
    alert( "Hi " + name + "!" );  
}
```



+ concatenates strings
as . does in PHP

Regular expression checking

A little different.

```
var myregexp = /^[A-Za-z]+$/  
if ( myregexp.test( inputstring ) ) {
```

```
...  
}
```

object

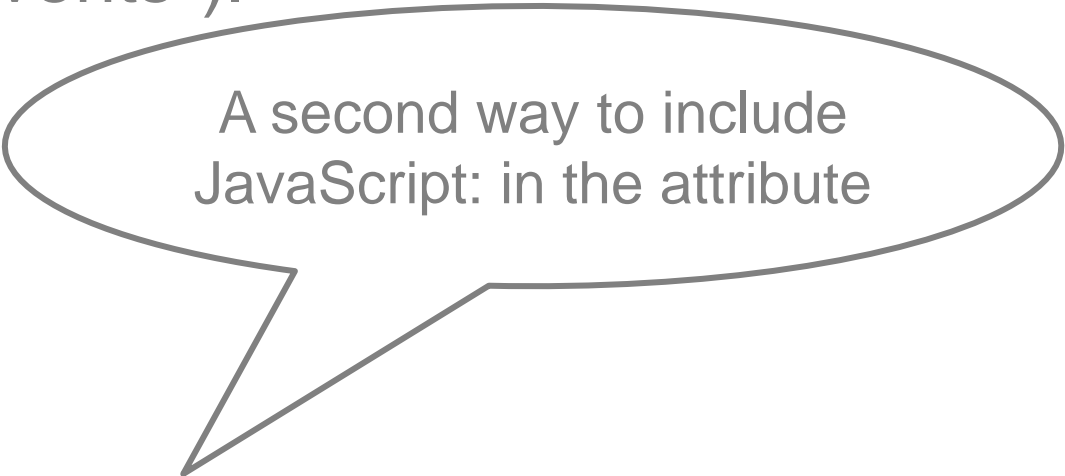
method

check to see if the variable
inputstring matches the
reg exp myregexp

Event handlers

JavaScript events

JavaScript can respond to user actions in the browser (e.g. “events”).



A second way to include
JavaScript: in the attribute

```
<a href="#" onmouseover="alert('I said, don't click  
here!');">Don't Click Here!</a>
```



Some events

onclick: User clicks on an area (e.g. button)

onmouseover: User moves mouse over an area
(e.g. link, image)

onmouseout: User moves mouse away from an
area

onchange: User changes the contents of a form
input

onsubmit: User submits a form

Rollover effects

```

```



Return values from events

If the code from our event handler returns “false”, it cancels the action that otherwise would have happened.

For instance,

```
<a href="link.html" onclick="return  
false;">Click here</a>
```

will do nothing if we click on the link.



Click in

JavaScript is executed:

- A. On the server, before the PHP for the page is executed
- B. On the server, after the PHP for the page is executed
- C. In the browser, after the PHP for the page is executed
- D. In the browser, before the PHP for the page is executed

Click in

JavaScript is executed:

- A. On the server, before the PHP for the page is executed
- B. On the server, after the PHP for the page is executed
- C. In the browser, after the PHP for the page is executed
- D. In the browser, before the PHP for the page is executed

If we want to execute some JavaScript when someone clicks on an HTML element (like a button), we use the event handler

- A. onClick
- B. onclick
- C. onmouseover
- D. OnSubmit

If we want to execute some JavaScript when someone clicks on an HTML element (like a button), we use the event handler

A. onClick

B. onclick

C. onmouseover

D. OnSubmit

What would “return false;” in JavaScript do if it were the result of an onsubmit event handler for a form?

- A. Clicking the button will submit the form
- B. Clicking the button won't submit the form
- C. Tell the user that the submission is not valid
- D. The page will crash.

What would “return false;” in JavaScript do if it were the result of an onsubmit event handler for a form?

- A. Clicking the button will submit the form
- B. Clicking the button won't submit the form
- C. Tell the user that the submission is not valid
- D. The page will crash.

An application: form checking

One use of JavaScript: check form input
before submitting a form.



Including a JavaScript file

```
<!DOCTYPE html>
```

```
<html>
```

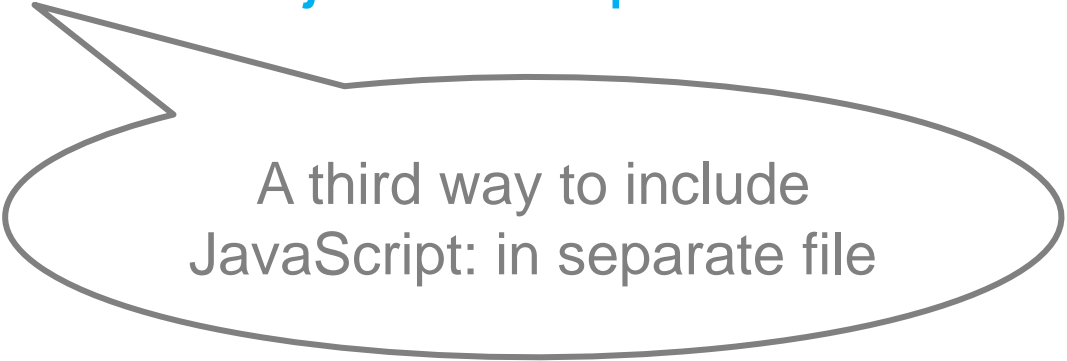
```
  <head>
```

```
    <meta charset="UTF-8">
```

```
    <title>JS Form Input Checking</title>
```

```
    <script src = "valid.js"></script>
```

```
  </head>
```



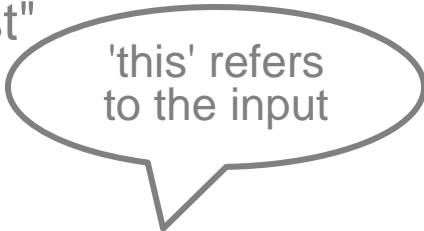
A third way to include
JavaScript: in separate file

How?

For starters, for each form element, we'll add an event handler to check that a correct input was put in.

```
<input type="text" name="zip"  
  onchange="validZip( this.value );">
```

```
<form name="myform" action="goodform.php" method="post"
  onsubmit="return validAll();">
```



'this' refers
to the input

```
<table>
```

```
<tr><td>Your name:</td>
```

```
  <td><input type="text" name="name" onchange="validName(this.value);">
```

```
  <td id="namemsg">Required</td> </tr>
```

```
<tr><td>Your zip code:</td>
```

```
  <td><input type="text" name="zip" onchange="validZip(this.value);">
```

```
  <td id="zipmsg">Required</td></tr>
```

```
<tr><td>Your phone:</td>
```

```
  <td><input type="text" name="phone" onchange="validPhone(this.value);">
```

```
  <td id="phonemsg">&nbsp;</td></tr>
```

```
<tr><td>Your email:</td>
```

```
  <td><input type="text" name="email" onchange="validEmail(this.value);">
```

```
  <td id="emailmsg">Required</td></tr>
```

```
<tr><td><input type="submit" name="submit" value="Submit"></td>
```

```
  <td>&nbsp;</td>
```

```
  <td id="submitmsg">&nbsp;</td></tr>
```

```
</table>
```

```
</form>
```

Message function

For right now, assume we can use a function `msg('idname', message)` that changes the text in the `<td id="idname"> </td>` element to “message”.

Writing the validating function

Now you try...

```
function validZip(zip) {
```

```
}
```



```
function validZip(zip) {  
    var check = /^[0-9]{5}$/;  
  
    if (zip == "") {  
        msg("zipmsg","Required");  
        return false;  
    } else if (check.test(zip)) {  
        msg("zipmsg","");  
        return true;  
    } else {  
        msg("zipmsg","Invalid 5-digit zip");  
        return false;  
    }  
}
```

Validating everything

```
function validAll() {  
    var name = validName(document.forms.myform.name.value);  
    var zip = validZip(document.forms.myform.zip.value);  
    var phone = validPhone(document.forms.myform.phone.value);  
    var email = validEmail(document.forms.myform.email.value);  
  
    if (!(name && zip && phone && email)) {  
  
    }  
}
```

Validating everything

```
function validAll() {  
    var name = validName(document.forms.myform.name.value);  
    var zip = validZip(document.forms.myform.zip.value);  
    var phone = validPhone(document.forms.myform.phone.value);  
    var email = validEmail(document.forms.myform.email.value);  
  
    if (!(name && zip && phone && email)) {  
        msg("submitmsg","Please correct errors before submitting form");  
        return false;  
    } else {  
        return true;  
    }  
}
```

Prevent submit

So far, we've given messages but the form could still be submitted. How do we use the `validAll()` function to prevent that?

```
<form name="myform"  
  action="goodform.php" method="post"  
  onsubmit="return validAll();">
```

Should we still validate on the server?



Review

- JavaScript is another useful programming language for web design; it runs on the browser under user control.
- JS useful for manipulating the webpage without going to the server via event handlers.
- You must still check user input on the server!