Introduction to JavaScript

INFO/CS 2300:
Intermediate Web Design and Programming

Handout on half wall but not the server

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

Section Friday

There is section on Friday. The break does not officially start until Saturday.

There is a quiz and an activity that will be useful for Project 2.

review lecture on monday

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

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>
```

Mini Crash Course

CSS

Tonight 7PM Gates 114

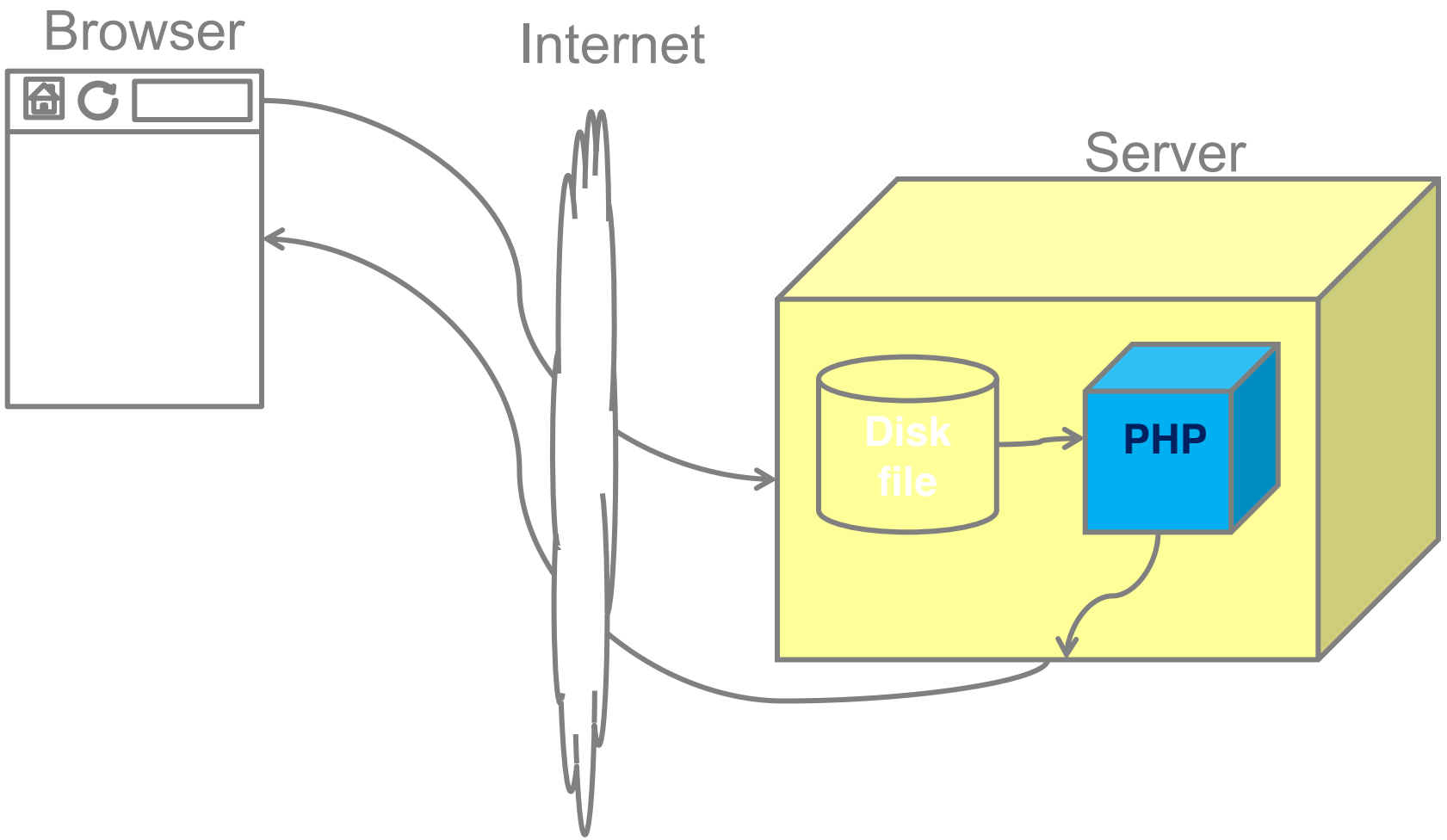
Details on Piazza

JavaScript

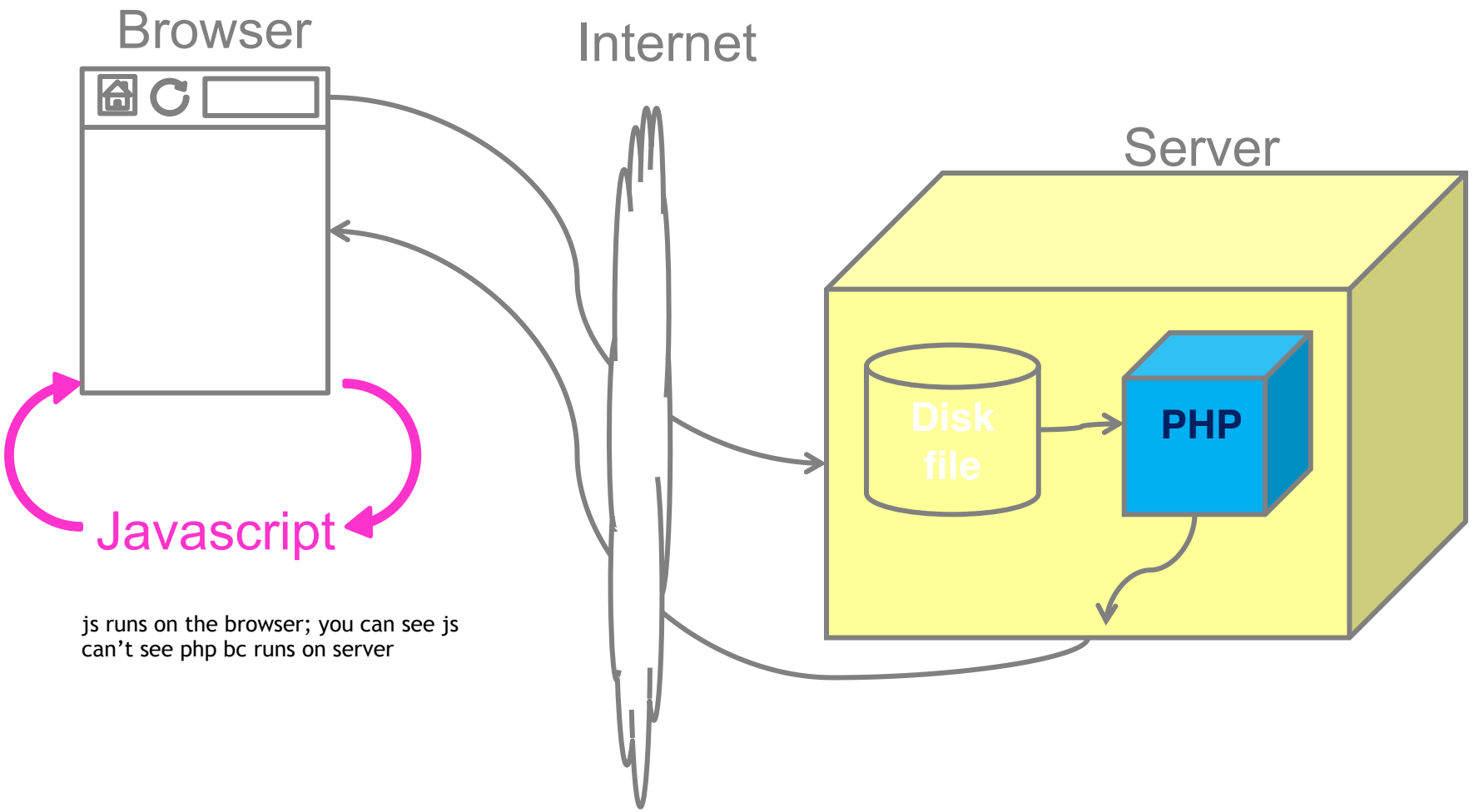
JavaScript is a programming language used in web programming, like PHP, but with some major differences.

http://all-free-download.com/free-photos/download/paddleboarder_catches_a_glass_wave_185295_download.html

The web with PHP



and Javascript



js runs on the browser; you can see js
can't see php bc runs on server

PHP vs. JavaScript

PHP:

- Runs on the server
- User can't see the code

JavaScript:

- Runs on the browser
- Code is visible to user

bc runs on browser

- Good for user interface
- User might disable JS
- Sometimes can reduce server load

JavaScript document.write

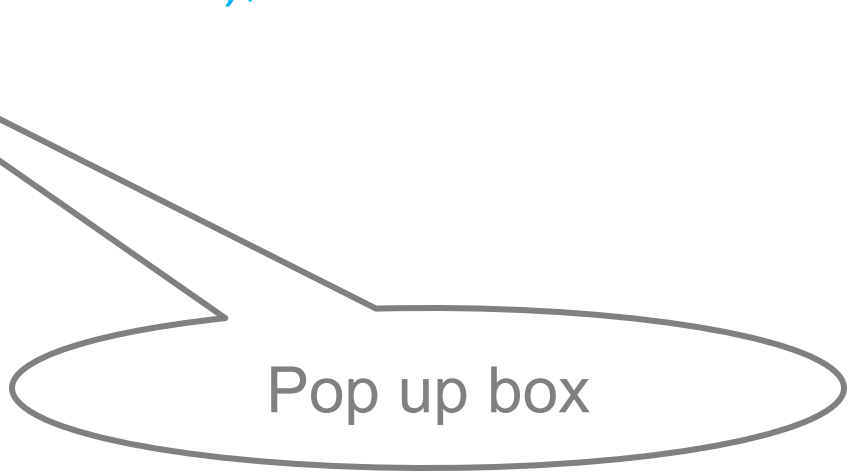
```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>JavaScript Document Write</title>
        <script type="text/javascript">
            document.write( "Hello world!" );
        </script>
    </head>
    <body>
    </body>
</html>
```

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>
```



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>
```

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

working in browser vs another;
may be variable conflict (firefox has own special variable ('fullScreen'))

Conditionals

Mostly the same, but no ‘elseif’.

```
if (mystring === “happy”) {
    …
} else if (mystring === “sad”) {
    …
}
```

Equal and Identical

Same as PHP

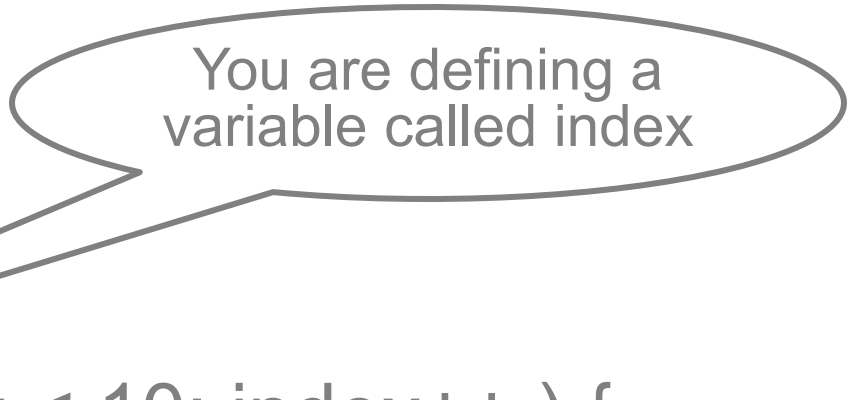
== does a type conversion before comparison

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

use triple equals unless you need to check type

Loops

Same as PHP



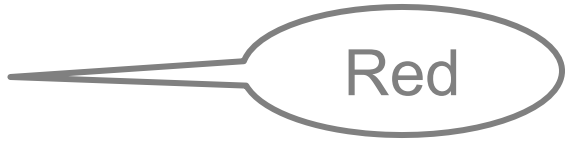

```
for ( index = 0; index < 10; index++ ) {
    …
}
while ( some_variable < 10 ) {
    …
    //some_variable increases inside the loop
}
```

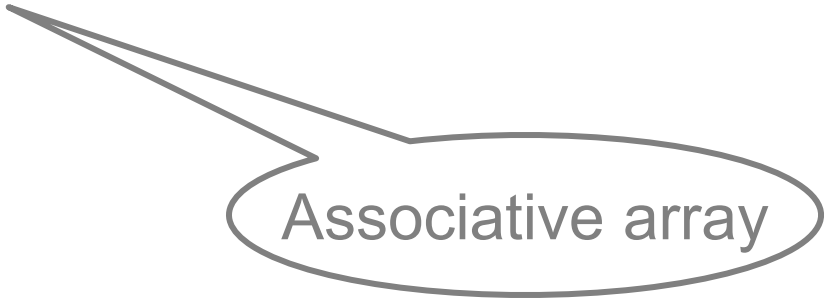
Arrays

Capitalize “Array”

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

document.write( menu[2] );
menu[ ‘Home’ ] = “index.html”;
```





Enumerating arrays

Significant syntax differences:

PHP

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

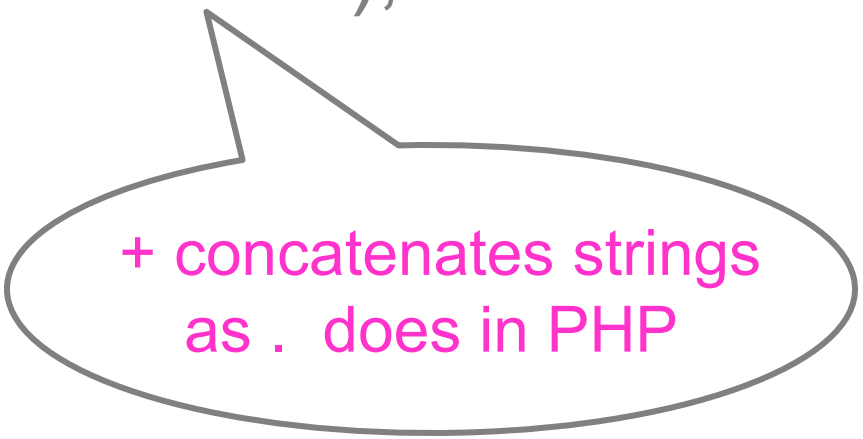
JavaScript

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

Functions

Same as PHP.

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



Regular expression checking

A little different.

No quotes needed

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

‘/‘ is delimiter

variables are objects
using function test on regex
(myregexp) on input string

object

method

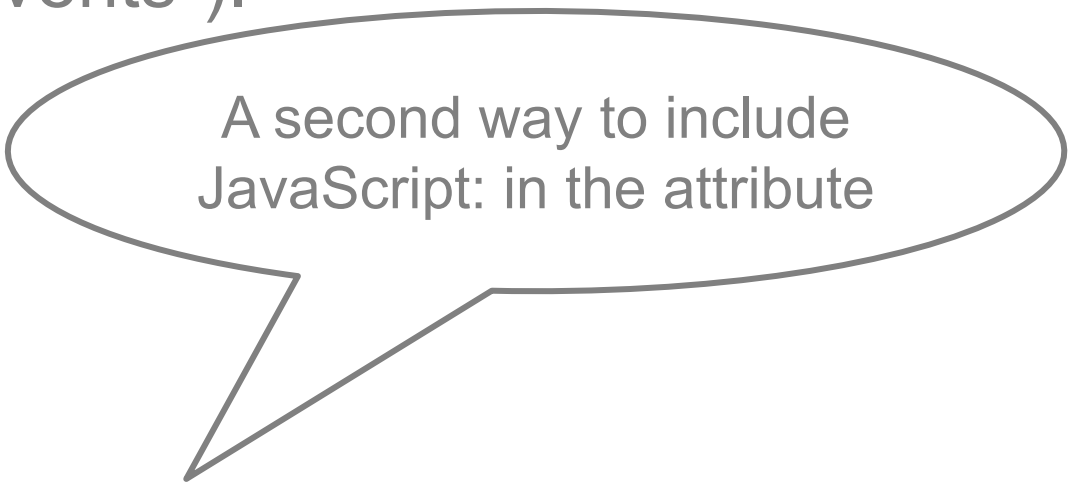
check to see if the variable
inputstring matches the
reg exp myregexp

Event handlers

js particular- interactive

JavaScript events

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



can include js in tag - do not do it this way

```
<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

```
<img src="bos.gif" width="146"
  onmouseover="this.src='nyc.gif';"
  onmouseout="this.src='bos.gif';">
```

Return values from events

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

For instance,

Click here

will do nothing if we click on the link.

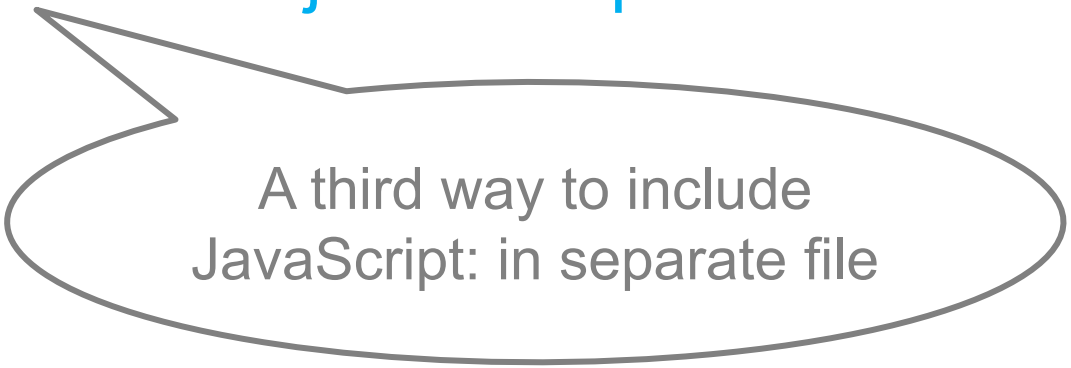
Click in

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>
```



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();">
<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"> </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> </td>
    <td id="submitmsg"> </td></tr>
</table>
</form>
```

'this' refers to the input

```
function validZip(zip) {
        if (zip.length() == 5) {return true};
}
```

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) {
    var check = /^[0-9]{5}$/;                    //^,$ start and end exp
    if (zip == “”) return false;
    else if (check.test(zip)) return true;
    else return false;









}
```

```
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?

check on server in case someone edits js

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!