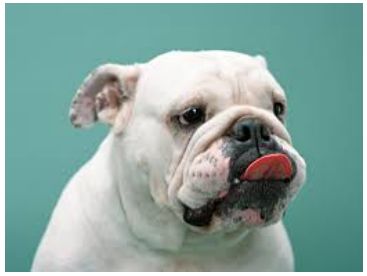
A PHP Form

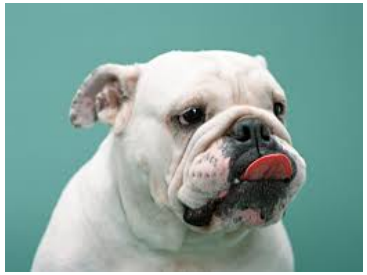
Class 9

Owen's Problem



- this is Owen
- and this is Owen's dog Fang
- Fang is lost
- Owen could search the neighborhood for Fang on foot

Owen's Problem



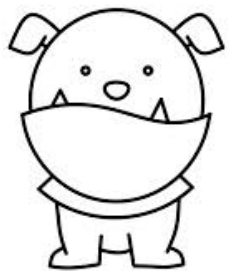
- this is Owen
- and this is Owen's dog Fang
- Fang is lost
- Owen could search the neighborhood for Fang on foot

- but actually, Fang was abducted by aliens
- so Owen has to search the entire galaxy

Owen's Idea

- Owen decides to create a web form
- he can collect information from people who might help him find Fang

REWARD



Have U Seen My Dog?

Form

Owen uses an editor to create an HTML file containing a form

- the file is pure HTML
- a web page may contain zero or more form elements
- each is just an HTML element denoted with `<form>` tags
- forms contain "normal" HTML

```
<p>
  Share your story of alien abduction
  by filling out the fields below!
</p>
```

- plus the form element

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

Form Contents

- inside the form tags there are input elements denoted by self-closing tags
- input is inline

```
<input type="text" id="whattheydid" name="whattheydid" size="32" />

<input id="spottedyes" name="spotted" type="radio" value="yes" />
```

Input Tags

there are many different kinds of input tags

text	defines a single-line string field
checkbox	defines a checkbox (multiple selections)
color	defines a color picker
date	defines a date control (year, month, and day)
datetime	defines a date and time control (year, month, day, hour, minute, second, and fraction of a second)
email	defines a field for an e-mail address
file	defines a file-select field for file uploads (“Browse...” button)
number	defines a field for entering a numeric value
password	defines a password field (characters are masked)
radio	defines a radio button (single selection)

More Input Tags

range	defines a control for entering a number whose exact value is not important (like a slider control)
search	defines a text field for entering a search string
submit	defines a submit button
tel	defines a field for entering a telephone number
time	defines a control for entering a time (no time zone)
url	defines a field for entering a URL
hidden	used to communicate from one php program to another

also in forms, but not inputs

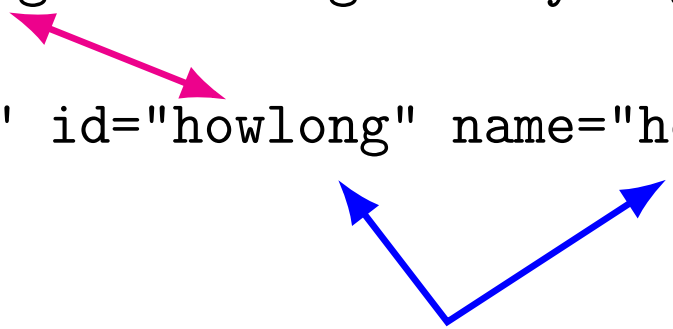
select	defines a drop-down menu, single or multiple
textarea	defines multi-line string field

Form Contents

- usually an input has a label element
- the for attribute must match an input's id

```
<label for="howlong">How long were you gone?</label>

<input type="text" id="howlong" name="howlong" />
```



- usually an input element's name and id are the same, to avoid confusion

The Form

Let's try it

- `http://borax.truman.edu/315/c09/reportform.html`

Test Drive

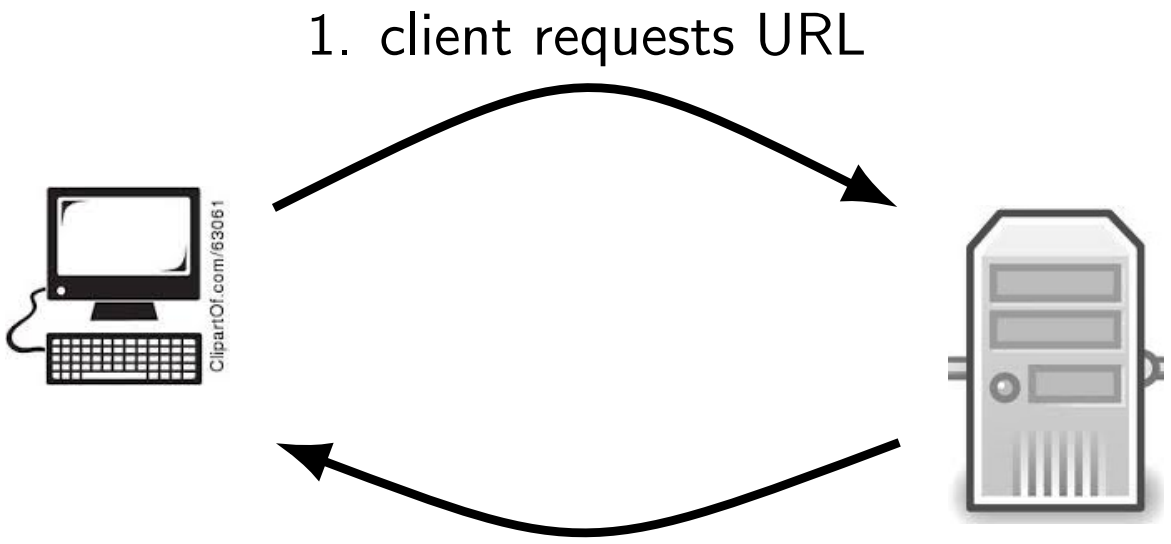
- filling out the form works great

Test Drive

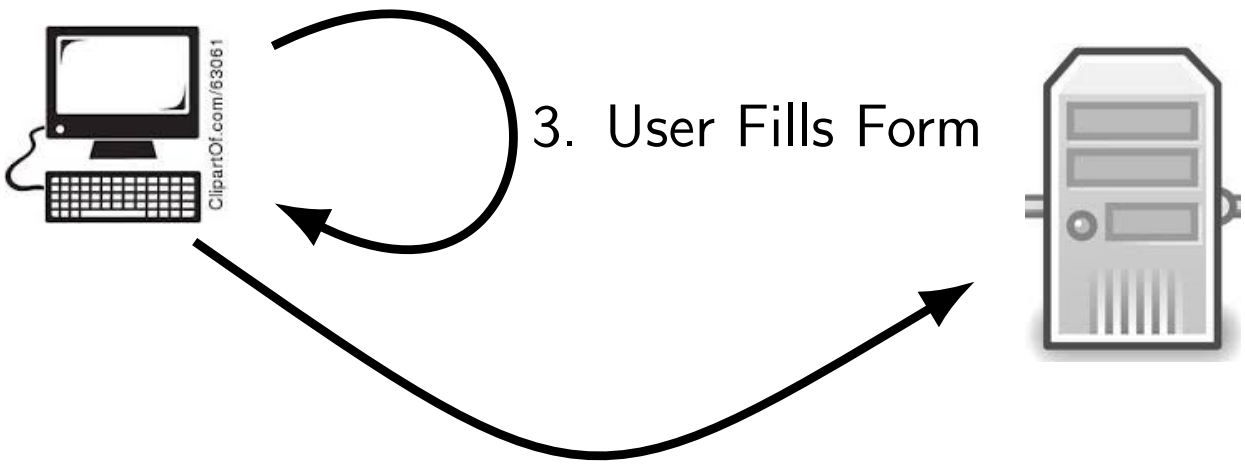
- filling out the form works great
- but what happens when we submit it?
- nothing seems to happen
- we haven't defined what "submit" means

Conversation

- here's what happens



Conversation



Associative Arrays

- before we can understand the final piece, we need to understand arrays in PHP
- the simple old-fashioned array does not exist in PHP
- instead PHP has associative arrays of key-value pairs
- by default, if keys are not specified, unsigned ints starting at 0 are used as keys

```
$foo = ['madam', 'im', 'adam'];

var_dump($foo);
array(3)
{
  [0] => string(5) "madam"
  [1] => string(2) "im"
  [2] => string(4) "adam"
}
```

Associative Arrays

- but keys can also be strings

```
$foo = ['foo' => 'bar',
        'bim' => 'bam',
        'knock knock' => 'whos there'];

var_dump($foo);
array(3)
{
  ["foo"] => string(3) "bar"
  ["bim"] => string(3) "bam"
  ["knock knock"] => string(10) "whos there"
}
```

Associative Arrays

- keys can also be a mix of ints and strings

```
$foo = [2 => 3.1415,
        'foo' => 'bar',
        5 => 'Annie',
        'Gertrude'];

var_dump($foo);
array(4)
{
  [2] => float(3.1415)
  ["foo"] => string(3) "bar"
  [5] => string(5) "Annie"
  [6] => string(8) "Gertrude"
}
```

Associative Arrays

- keys must be ints or strings; values can be anything

```
$foo = [['ann', 'bob', 'carol'],
        ['deb', 'eve', 'frank', 'gil']];

var_dump($foo);
array(2)
{
  [0] => array(3)
         {
           [0] => string(3) "ann"
           [1] => string(3) "bob"
           [2] => string(5) "carol"
         }
  [1] => array(4)
         {
           [0] => string(3) "deb"
           [1] => string(3) "eve"
           [2] => string(5) "frank"
           [3] => string(3) "gil"
         }
}
```

Associative Arrays

```
$foo = ['foo' => 'bar',
        'bim' => 'bam',
        'knock knock' => 'whos there'];

echo $foo['bim'][1];

$index = 'knock knock';
echo $foo[$index];
```

Associative Arrays

```
$foo = ['foo' => 'bar',
        'bim' => 'bam',
        'knock knock' => 'whos there'];
```

add to an array: `$foo['clockwork'] = 'orange';`

delete an array element: `unset($foo['bim']);`

```
print_r($foo);
Array
(
    [foo] => bar
    [knock knock] => whos there
    [clockwork] => orange
)
```

Errors

- it is a logical error to refer to an array index that does not exist

```
$foo = ['foo' => 'bar',
        'bim' => 'bam',
        'knock knock' => 'whos there'];

echo $foo['quux'];
PHP Notice:  Undefined index: quux in php shell code on line 1
```

- the PHP program does not "crash" — it is forgiving of this type of error
- but it is still an error and almost certainly means something is wrong

Submit to a Program

- to submit we need an action attribute on the form
- currently I don't have an action

```
<form method="post">
```

- the action must be the URL of a program
- upon submitting, the data is sent to that program
- we change the tag to:

```
<form method="post" action="processform.php">
```

- note that the method is "post"
- we will explore that shortly
- for now, just note that data comes via POST

Receiving Data

- we need a program that can receive POST data
- here is a PHP program that looks at POST variables
  `http://borax.truman.edu/315/c09/processform.phps`

Transferring Data

- the receiving PHP program makes use of the `$_POST` associative array
- each named input element (as well as textarea and select) in the `<form>` element of reportform.html creates an element of the `$_POST` array in processform.php