Web Technologies

Client Side Validation

Validation

Sending data is not enough — we also need to make sure that the data users fill out in forms is in the correct format we need to process it successfully, and that it won't break our applications.

We also want to help our users to fill out our forms correctly and not get frustrated when trying to use our apps.

Form validation helps us achieve these goals

Validation

- "This field is required" (you can't leave this field blank)
- "Please enter your phone number in the format xxx-xxxx" (it wants three numbers followed by a dash, followed by four numbers)
- "Please enter a valid e-mail address" (the thing you've entered doesn't look like a valid e-mail address)
- "Your password needs to be between 8 and 30 characters long, and contain one uppercase letter, one symbol, and a number" (seriously?)

Validation

- **We want to get the right data, in the right format** — our applications won't work properly if our user's data is stored in any old format they like, or if they don't enter the correct information in the correct places, or omit it altogether.
- **We want to protect our users** — if they entered really easy passwords, or no password at all, then malicious users could easily get into their accounts and steal their data.
- **We want to protect ourselves** — there are many ways that malicious users can misuse unprotected forms to damage the application they are part of

Client Side Validation

We’ve looked at validation on the server side

We’ll now look at validation on the client side

What does client side validation mean ?

Client Side Validation

Client-side validation is validation that occurs in the browser, **before** the data has been submitted to the server.

This is more user-friendly than server-side validation as it gives an instant response. This can be further subdivided:

- JavaScript validation is coded using JavaScript. It is completely customizable.

- Built in form validation is done with HTML5 form validation features, and generally doesn't require JavaScript. This has better performance, but it is not as customizable. (this is what we'll look at today)

Server Side Validation

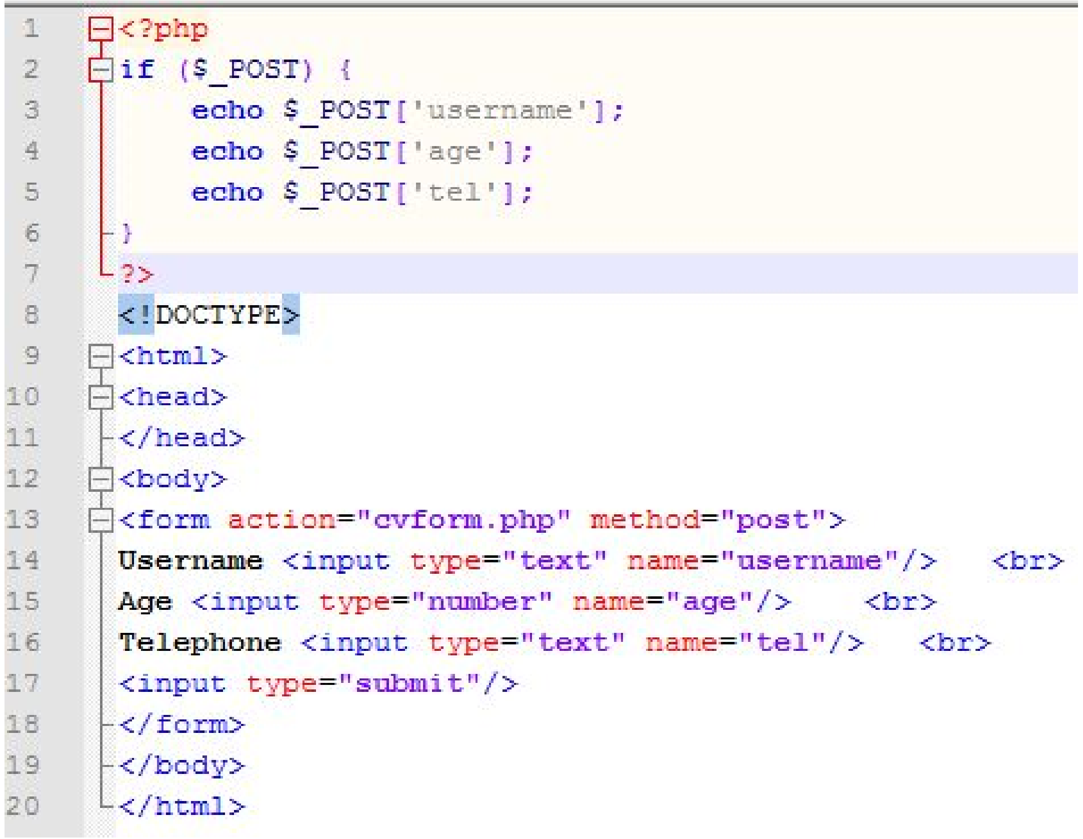
Server-side validation is validation that occurs on the server, after the data has been submitted e.g. by a form ‘post’

Server-side code is used to validate the data before it is put into the database, and if it is wrong a response is sent back to the client to tell the user what went wrong.

Server-side validation is not as user-friendly as client-side validation, as it requires a round trip to the server, but it is essential — it is your application's last line of defense against bad (meaning incorrect, or even malicious) data.

Client Side Validation

Lets start with a simple form. Download form **cvform.php** from moodle and place under htdocs in a new folder say **lecture9**



```
<?php
if ($_POST) {
    echo $_POST['username'];
    echo $_POST['age'];
    echo $_POST['tel'];
}
?>
<!DOCTYPE>
<html>
<head>
</head>
<body>
<form action="cvform.php" method="post">
Username <input type="text" name="username"/>   <br>
Age <input type="number" name="age"/>    <br>
Telephone <input type="text" name="tel"/>   <br>
<input type="submit"/>
</form>
</body>
</html>
```

Client Side Validation

We can add **attributes** to an <input> field to add some validation .

The first we will look at is the **required** attribute. This essentially checks if the field is not blank (similar to the empty() function in php).

```
<input type="text" name="username" required/>
```

Add the attribute to all the fields and press submit without entering any data

Note the error message. Also note that the form is **not** posted until the form is valid

Client Side Validation

<input type='text'>

We can add other attributes to a "text" input field

minlength	-	minimum number of characters
maxlength	-	maximum number of characters

```
<input type="text" name="username" required minlength='3' maxlength='40'/>
```

Client Side Validation

<input type='number'>

We can add other attributes to a “number” input field

min	-	minimum value
max	-	maximum value

```
<input type="number" name="age" required min='18' max='35'/>
```

Client Side Validation

<input type='email'>

We can add other input types such as ‘email’

```
Email <input type="email"  name="email" required>
```

https://www.w3schools.com/tags/att_input_type.asp

Add a type='date' input field and see what happens

Client Side Validation

Why is client side validation only not sufficient ?

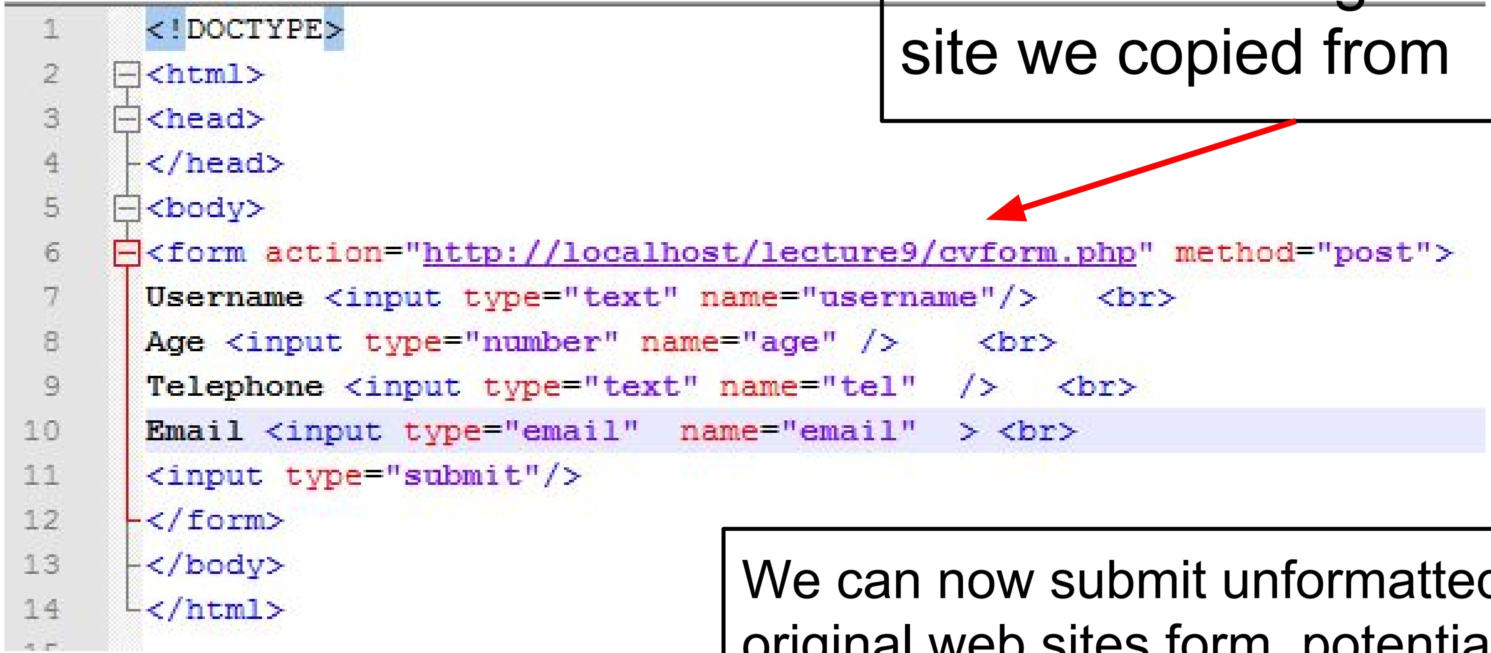
Let’s look at our web page and do a view source

Why can’t we see the php code ? (we’ll discuss this)

We can take a copy of this web page and remove all the validation

We can then submit the original web page form without any validation.

Client Side Validation



```
<!DOCTYPE>
<html>
<head>
</head>
<body>
<form action="http://localhost/lecture9/cvform.php" method="post">
Username <input type="text" name="username"/>   <br>
Age <input type="number" name="age" />    <br>
Telephone <input type="text" name="tel"  />   <br>
Email <input type="email"  name="email"  > <br>
<input type="submit"/>
</form>
</body>
</html>
```

Form on the original web site we copied from

We can now submit unformatted data to the original web sites form, potentially causing problems. This is why we need server side code as a last defence.

Client Side Validation

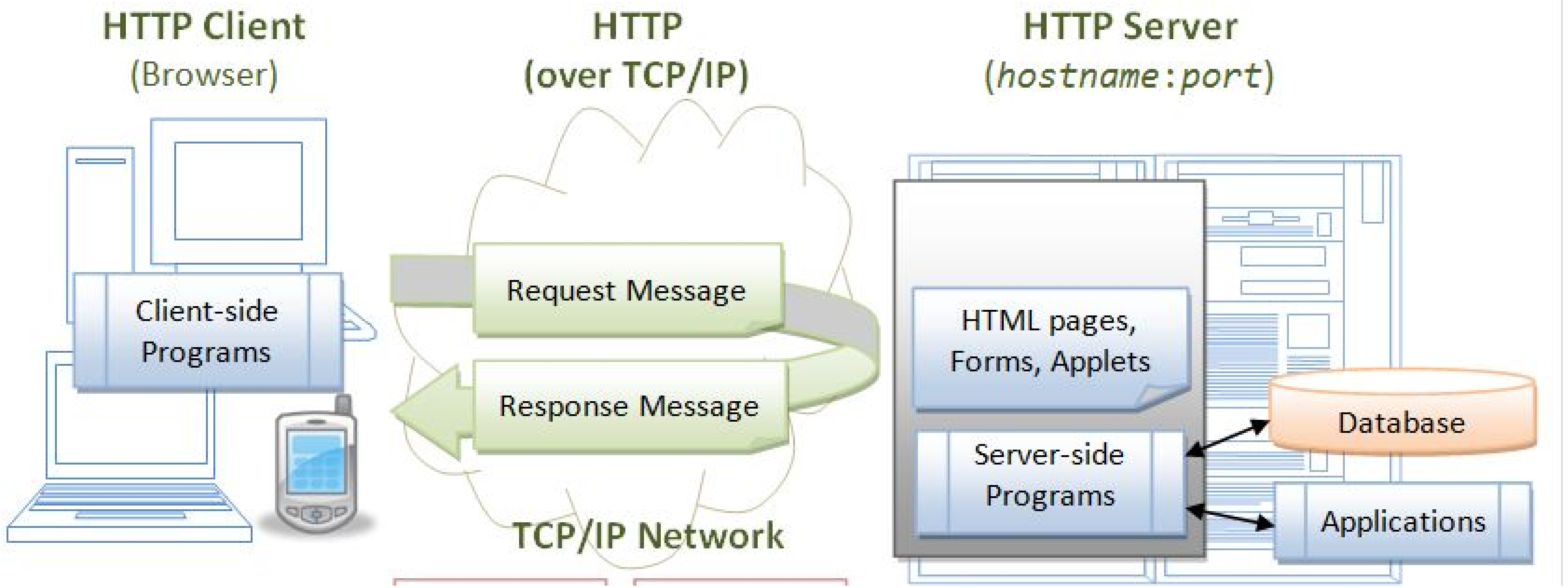
Remember what we are seeing in our browser is the ‘executed’ code from our php file.

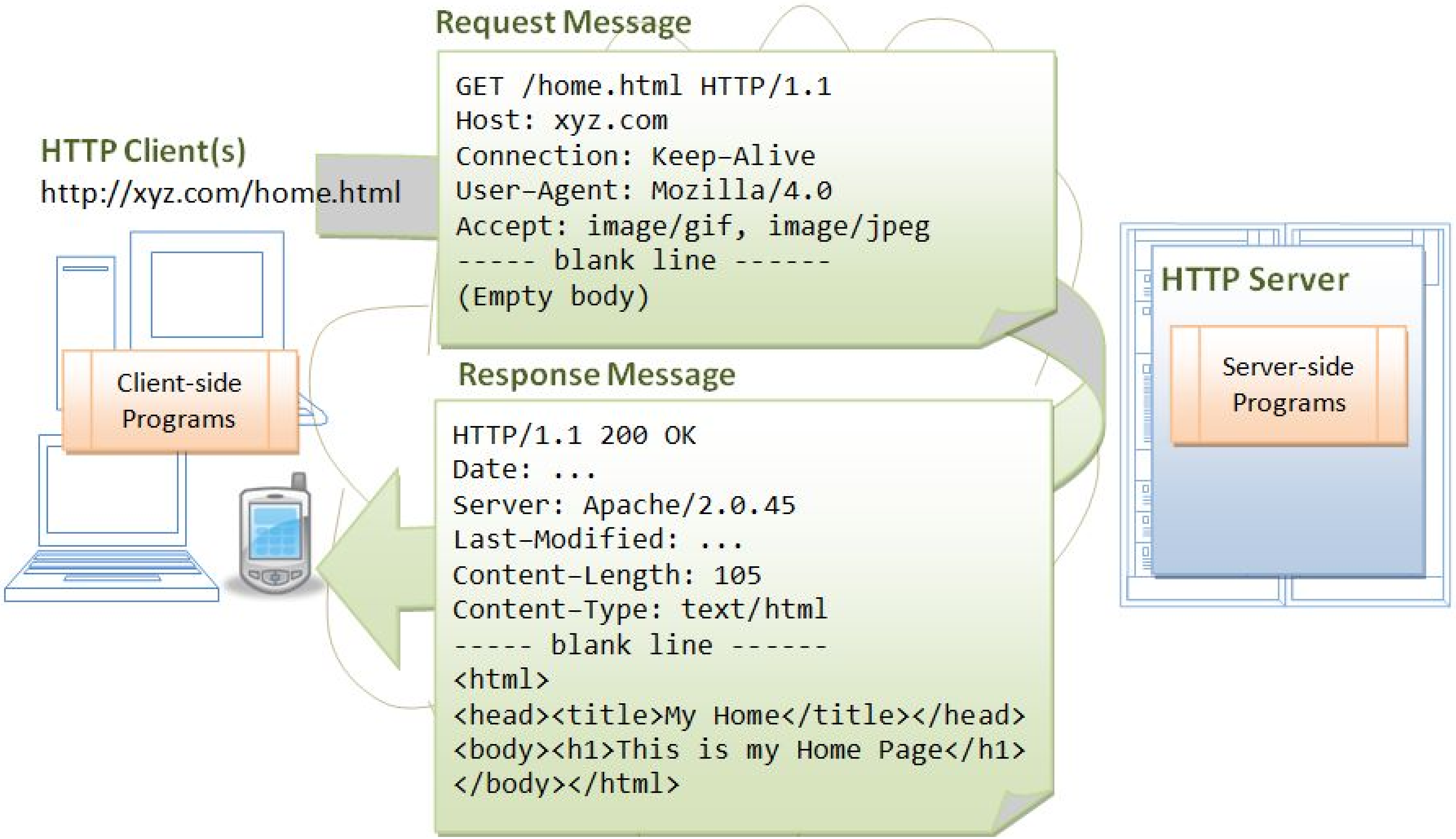
In our case when we request the php file from the web server we are seeing a combination of the

- html text
- **text** output from the ‘executed’ php code

This is why we don’t see any php code when we do a view source in the browser

Production Environment





```
GET /home.html HTTP/1.1
Host: xyz.com
Connection: Keep-Alive
User-Agent: Mozilla/4.0
Accept: image/gif, image/jpeg
----- blank line ------
(Empty body)
```

```
HTTP/1.1 200 OK
Date: ...
Server: Apache/2.0.45
Last-Modified: ...
Content-Length: 105
Content-Type: text/html
----- blank line ------
<html>
<head><title>My Home</title></head>
<body><h1>This is my Home Page</h1>
</body></html>
```

Server Side Validation

In other words if we’ve set up our web server security properly it’s difficult for anyone to ‘**Get At**’ the php code and manipulate the server side validation

Client Side Validation

Conclusion

- client side validation is useful from a ‘user experience’ point of view
- it can enhance the user experience by informing the user as to formatting errors and what is expected
- it’s more efficient than server side validation as we don’t send the data to the server until the form is valid
- we still need server side validation as a last line of defence