PHP - Form validation

<form action =" file.php" method =" get" >

File.php

<? php echo $_GET["name"]; ?

GET

· create arr (key1 => val, ...)

POST

↙ Supervariables

$_GET

↳ passed var via URL param

↳ VISIBLE to EVERYONE

↳ limits info sent 2000 char

$_POST

↳ passed var via HTTP post method

↳ INVISIBLE to EVERYONE

↳ no limit on info sent

Form Validation

· can have required/opt fields

· can set allowed chars

<form method =" post" action =" <? php echo

Sanitizes variables → htmlspecialchars($_SERVER["PHP_SELF"]); ?> " >
(stops cross-site scripting)

↑ superglobal that returns currently executing script
- sends data to page itself
↳ gets error on same page

use · trim($data) - removes ^unnecessary whitespace

· stripslashes ($data);

· htmlspecialchars($data);

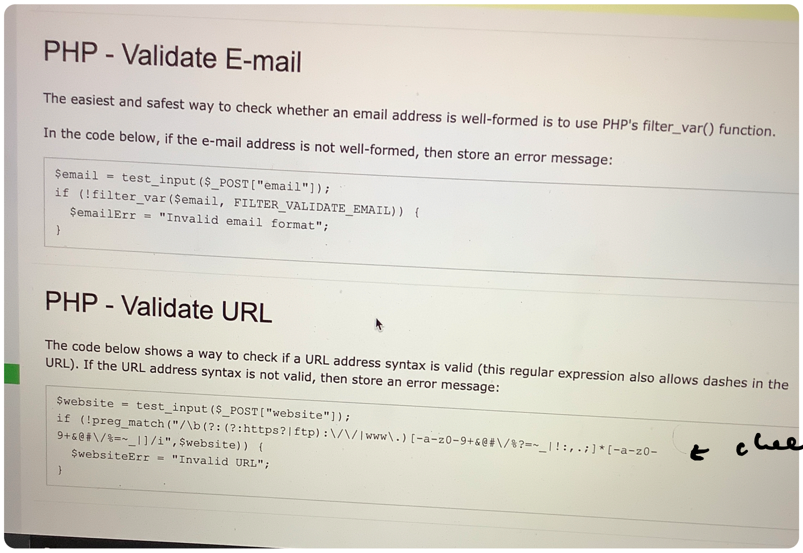
if ($_SERVER["REQUEST_METHOD"] == "POST") {...}
↳ checks if anything posted

Required inputs

· if (empty($_POST["name"])) {
$err = "thing is empty"; }

· <span class="error">* <?php echo $err ; ?> </span>
↳ displays error

Valid Email



PHP - Validate E-mail

The easiest and safest way to check whether an email address is well-formed is to use PHP's filter_var() function.

In the code below, if the e-mail address is not well-formed, then store an error message:

```
$email = test_input($_POST["email"]);
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
  $emailErr = "Invalid email format";
}
```

PHP - Validate URL

The code below shows a way to check if a URL address syntax is valid (this regular expression also allows dashes in the URL). If the URL address syntax is not valid, then store an error message:

```
$website = test_input($_POST["website"]);
if (!preg_match("/\b(?:(?:https?|ftp):\/\/|www\.)[-a-z0-9+&@#\/%?=~_|!:,.;]*[-a-z0-
9+&@#\/%=~_|]/i",$website)) {
  $websiteErr = "Invalid URL";
}
```

← checks chars

valid name

if (! preg_match ("/^[a-zA-Z]*$/", $name))

$\underbrace{\texttt{[a-zA-Z]*}}_{\text{only a-z or A-Z}}$

Keeping values in form

- value = "<?php echo $name;?>">