```
    <?php file_put_contents('foo.txt', '.$foo.'); ?>
    var baz = <?php echo 42; ?>;
    alert(baz);
</script>
```

step 1, PHP executes all code between <?php ?> tags. The result is this:

```
<script type="text/javascript">
    var foo = 'bar';
    var baz = 42;
    alert(baz);
</script>
```

The file_put_contents call did not result in anything, it just wrote "."$foo."" into a file. The <?php echo 42; ?> call resulted in the output "42", which is now in the spot where that code used to be.

This resulting HTML/Javascript/CSS code is now sent to the client (browser), where it gets evaluated. The 'alert' call works, while the 'foo' variable is not used anywhere

Date 21/02/2017

Ajax ⇒ Asynchronous Javascript and XML

XML HttpRequest

All modern browsers have a built-in XMLHttpRequest object to request data from a web server.

Benefits of XMLHttpRequest object

The XMLHttpRequest object is a developer's dream, because using it, we can:

1. Update a web page without reloading the page
2. Request data from a server - after the page has loaded.
3. Receive data from a server - after the page has loaded.
4. Send data to a server - in the background.

Reference: https://www.w3schools.com/xml/xml_http.asp

Date
21/02/2017

cURL

PHP supports libcurl, a library created by Daniel Stenberg, that allows you to connect and communicate to many different types of servers with many different types of protocols.

libcurl currently supports the http, https, ftp, gopher, telnet, dict, file, and ldap protocols.

libcurl also supports HTTPS certificates, HTTP POST, HTTP PUT, FTP uploading, HTTP form based upload, proxies, cookies, and user+password authentication.

In other words:

cURL is a library that lets you make HTTP request in PHP.

cURL is a way you can hit a URL from your code to get a html response from it. cURL means client URL which allows you to connect with other URLs and use their response in your code.

Reference: php.net/manual/en/intro.curl.php

Date
21/02/2017

Reference: www.pcds.co.in/java/what-is-pear.php

PEAR

PEAR (PHP Extension and Application Repository) is a framework and distribution system for reusable PHP components.

This project provides a structured library of code, maintain a system for distributing code and for managing code packages & promote a standard coding style.

Date
21/02/2017

php.ini directives (variables)

display_errors : On/Off

error_reporting : 0/E_ALL

E_NOTICE, E_WARNING, etc.

0 - zero to report nothing

E_ALL - To report all level/type of errors

display_errors determines whether errors should be printed to the screen as part of the output or if they should be hidden from the user.

error_reporting determines to which level of errors to report and which not. The parameter is either an integer or a named constant (php.net/manual/en/errorfunc.constants.php).

Imp when we are not on production server i.e. if we are on development server/mode, we must use following 2 lines to get reported & display all type of errors.

```
<?php
error_reporting(E_ALL);
ini_set("display_errors", 1);
?>
```

The same as above is used in zend-framework project/public/index.php

---

Reference : www.pcds.co.in/java/how-to-get-useful-error-messages-in-php.php