

CNT2500 - ICA01 - PHP Intro

This ica will incorporate php code in the most common forms.

> An inline form, where php code is interpreted and content replaced within the body of the html document.

> A pre-fix form, where php code sits as a sentinel at the top of the page, allowing the page to act as both client and server page in divisive sections.

> A completely separate php library/include section, performing unified and common setup, initialization and/or construction.

Using grid, a 2 column site and a handful of css formatting rules :

ica01_php

Part I : Server Info

Your IP Address is : 192.197.128.19
\$_GET Evaluation : Found : 4 entry in the \$_GET
\$_POST Evaluation : Found : 0 entry in the \$_POST

Part II : Form Processing

\$_GET Contents :

- [Name] = Shane
- [Hobby] = Stored Procedures
- [HowMuch] = 5
- [submit] = Go Now !

Part III : Array Generation :

Array Generated :

- 10
- 5
- 3
- 1
- 6
- 2
- 9
- 7
- 8
- 4

Part IV : Form Processing

Name :
Hobby :
How Much I like it :
Go Now !

Shane really really really really really likes Stored Procedures

Status : +ProcessForm+ServerInfo+GETData+GenerateNumbers+MakeList+ShowArray

© Copyright 2019 by Λαησεζω
Last Modified:01/02/2019 10:10:11

Inline PHP :

Part I : Server Info, output the information as shown :

This uses the 'REMOTE_ADDR' property of the global \$_SERVER collection and the count() function for arrays. Two column output is evident.

Inline PHP :

Part II : Form Processing, output the information as shown :

Use the foreach() construct to create an unordered list of the available \$_GET entries. Two column output is evident.

Prefix PHP : At the top of your page in a php block :

Declare a global variable called \$status, initialized to an empty string. This will be a string and we will append step completion as "+Step" as they are completed.

In a separate utility file : util.php, in an appropriate PHP block, add some global functions as shown below :

In your main page, as the first thing, use the require_once directive to add util.php

Add a function called GenerateNumbers(), declare an array and using a for() loop. Add the numbers 1 to 10 into the array. Now use shuffle() to mix'em up. Return the array.

Add a function called MakeList(), accepting an array. Declare an output string variable. Then use a for() loop, construct a HTML ordered list from the array values, incrementally assigning to your output variable. Return this output "list" variable.

Part III : Array Generation, appropriately invoke your 2 functions, then add a single echo() statement to inject your ordered list into your html. Two column output is evident.

Part IV : Form Processing, similar to CMPE2000:ica10, make a Name/Hobby/HowMuch block, wrap it in a form, altering the elements to ensure they are all named. Set the form type to "get", set the action to "this" file, everything will be processed within this page. . Change the button to an input/submit. Now the page will Submit to itself... Now process your self. The result of the form processing follows below as shown

At the top of your page, interrogate the \$_GET incoming data, to see if it contains a property called (what-ever-you-called-your-form-element), then verify that it has a strlen() greater than 0. Use the && operator to chain at least a check for the "Hobby" property as well. If the conditions are met, we assume this is a form submission, so in the if() body, extract the 3 form fields into variables using strip_tags() to remove malicious HTML. (Should do more checking, but good enough). Now use a variable to build the string like ica10 did. As this code block lives in global scope, it will be available for use down further in our page.

In another container after your form, put your output <label>. Put a php block inside the label body, and echo out the constructed form string. It may be empty if this wasn't a form submission - that is fine.

Finally, for each of the steps/blocks/functions, Concatenate a description to the global status variable, finally, echo out the status string in our container as shown.