

Announcements

- Submit Server and grades
 - Check you can access them
- Language reference
<http://www.php.net/manual/en/langref.php>
- Listing of PHP Functions available at:
<http://www.php.net/manual/en/indexes.functions.php>

Arrays

- Three approaches for creating arrays
 - Implicit creation → By just assigning a value to a variable representing an array
Example: `$data[1] = 10;`
 - By calling a function that returns an array
 - By using the `array()` construct
 - **Example:** `$languages = array("C++", "Java", "Fortran");`
 - **Example:** `$dictionary = array("House" => "Casa", "Friend" => "Amigo");`
- When no explicit indices are specified, array starts at 0
- Indexing of elements possible via `[]`
- Iteration of elements possible via **foreach** statement that has the following syntax:

`foreach ($array as $arrayElement)`

- **Example:** `arrayCreation.php`
- Empty array: `$emptyArray = array();`

Arrays

- **Helpful functions**
 - `count()/sizeof()` → returns the size of the array
 - `array_keys()` → returns array with the keys
 - `array_values()` → returns array with the values
 - **Example:** `arrayFunctions.php`
- **Iteration Alternative for Associative Arrays**

`foreach ($array as $key => $value)`

- **Example:** `arrayFunctions.php`
- You can use `print_r` to print information about the array
- To delete an element from an array use `unset` (e.g. `unset($data[0])`)
- Multidimensional Arrays
 - Arrays that have other arrays stored in them
 - **Example:** `multidimensionalArrays.php`

String Functions

- `trim()` → returns a string where white spaces from the start and end of the string have been removed
- `rtrim()` → returns string where white spaces have been removed from the end of the string
- `ltrim()` → returns string where white spaces have been removed from the start of the string
- `strtoupper` → returns string in upper case
- `strtolower` → returns string in lower case
- `ucwords` → returns string where the first letter of every word is in uppercase
- `ucfirst` → returns string where only the first letter in the string is capitalized
- `nl2br` → (newline to br) - converts new lines in a string into an HTML breaks
- `wordwrap` → wraps lines and uses `\n` as its line break character
- `explode()` → breaks up a string into an array. Requires two arguments: a delimiter string and the source string
- **Example:** `stringFunctions.php`

HTML Forms

- Forms - means by which information passes from the user to the server
- Tags
 - `<form>` → defines the form. It has two attributes:
 - `action` → indicates where the form contents will be sent when the form is submitted
 - `method` → defines how the contents will be sent
 - `post` → contents sent using the HTTP POST method . Content is “hidden”
 - `get` → contents sent using the HTTP GET method. Contents included in the URL
 - `<input>` → appears inside of the `<form>` tag and defines several input data alternatives
The general format is:

`<input type="ALTERNATIVE" />`

where ALTERNATIVE can be `text`, `password`, `checkbox`, `radio`, `file`, `submit`, `image`, `button`, `reset`, `hidden`

- We can retrieve form information in the target script through `$_POST` or `$_GET`
- **Example:** `formPost.php/postProcessing.php`
- **Example:** `formGet.php/getProcessing.php`

HTML Forms

- Advantages/Disadvantages of POST and GET
- Let's go over an example that summarizes several of the input elements we have in forms
- **Examples:**
 - formsSummary.html, formsSummary.php
 - HTML5/CSS3 Form elements example
 - <http://www.cs.umd.edu/~nelson/classes/utilities/HTML5CSS3/FormElements.html>
- We can create our own customized search using forms, get and
 - *http://search.yahoo.com/search?p=*
 - *http://www.google.com/search?q=*