• Passing Variables:

• GET: `<a href="name.php?a=xxx&b=xxx&..."> ... </a>`

at name.php:

```
$x = $_GET['a'];
$y = $_GET['b'];
```

• POST:

```
< form action = "name.php" method="post" >
    <input name="a" ... />
    <input name="b" ... />
    <input type="submit" ... >
< /form >
```

← to have the submit button re-route to the same page, replace with: `action="<?php print $_SERVER['PHP_SELF'] ?>"`

at name.php:

```
$x = $_POST['a'];
$y = $_POST['b'];
```

Notes:

```
<input type="checkbox" name="choice[]">
<  ''   ''   -- name="choice[]">
```

access them with

```
$_Post["choice"][0]
$_POST["choice"][1]
```

• External Files:

`include("file.php");` ← As if the content of this file are now all present wherever we called it.

Note: file not found => continues

`require("file.php");` stops the entire php script if file is not found.

• Dynamic Function:

```
function name(){...}
$hello = "name";
$hello(); <- works like a normal function
```

• Upload Files (Forms):

```
< form ... enctype="multipart/form-data" method="POST">
    <input type="hidden" name="MAX_FILE_SIZE" value="51200">
    <input type="file" name="upload3">
    <  ''   ''  ="submit" value="finish">    </form>
```

information about the uploaded file becomes available in `$_FILES['upload3']['...']`

type (images/jpg) size (int) name (ng.jpg)

Browsing Directories (folders):

```
$x = opendir("C:\\ ...\ch3"); <-false if path is invalid
$file=scandir("C:\\ ...\"); <- file is now an array
don't forget to closedir($x);
```

Note: file will have `'.'` `'..'` as the first 2 entries => must be removed.

Note: we can also use `while(false !== ($file = readdir($x)));` to read one-by-one.

Note: scandir needs no opening & closing

Text Files:

touch("url") file exists: only changes "last modified"
else: creates the file

unlink("url") delete

return false if failed <- fopen("url", mode);
- -> read
- -> write
- -> append

Note: usually we use the combination:

fopen("url", m) or die("rip");
- equivalent to exit
- runs only if fopen fails

Don't forget to fclose("url");

reading ($fp=fopen("url");)

- fgets($fp, 1024); <- reads 1024 bits or till next '\n'
- while(!feof($fp)); <- while not at end of file
- fread($fp, 1024); <- same as fgets but won't stop at \n
- fgetc($fp); for one char at a time

writing

- fwrite($fp, "string");
- fputs($fp, "string");

- readfile($fp); returns entire file followed by number of chars.
- file($fp); returns array of lines.

Printing Arrays: print_r($array);

ex) $a = array('hi' => 'hello',
'no' => 'die',
'nu' => array('hey', 'test'));

prints:

```
Array ( [hi] => hello
        [no] => die
        [nu] => Array
                (
                  [0] => hey
                  [1] => test
                ),
)
```