Including Files

Intro PHP

Download sample code from Nexus

What should be included?

Include code that is likely to be utilized in multiple places within the program, or code that is common to all files.

```
<?php include './inc/header.php'; ?>

<div id="container">
<!-- page content here -->
</div>

<?php include './inc/footer.php'; ?>
```

Rule of thumb: remove to an external file any code that might be common to every page, or used more than once.

It is also common to include utility files like function libraries and configuration options.

In this example, the HTML header and footer are being included into a page.

The header and footer might be included on every page in the site. If something in the header or footer is changed, that change would be reflected everywhere that the header and footer are used.

01_include_common.php

PHP files can include other files

To import external files into the current file, use simple include or require statements.

```
$title = 'Include Common Files.';

include './inc/functions.php';
```

This statement instructs PHP to include a file named functions.php into the current file at the location where the statement is executed.

PHP will attempt to find the file at the path specified in the statement.

Once included, PHP behaves as if the code in the included file had actually been present in the current file.

Rules of scope will be applied to the included file as if the code in it had been present in the current file.

In the above example, the variable $title can be access by any code within functions.php. Likewise, any functions in functions .php will be accessble in the current file after the point where functions.php is included.

01_include_common.php

Traverse the file system

Depending on your php.ini configuration, PHP can traverse your entire file system to find files to import.

Given this folder structure:

/var/www/site/public/index.php

/var/www/site/inc/header.php

To include header.php into index.php, PHP will need to reach out of the public folder to find the file.

```
include __DIR__ . '/../inc/header.php';
```

02_traversal.php

The __DIR__ magic constant refers to the absolute path of the directory in which the constant is used. In this case, the public folder in which index.php resides.

/var/www/site/public

The include statement then reaches up one folder from public, into the site folder: /..

And finally, into the inc folder to find the requested file: /inc/header.php

The final constructed absolute path will look like this:

/var/www/site/public/../inc/header.php

Include vs Require

There are two language keywords that can be used to import files: include and require

```php
include __DIR__ . '/inc/sidebar.php';
```

Use include for non-critical files. If PHP cannot find these files, the site can still be displayed, although perhaps not perfectly. If the file cannot be found, PHP will issue a warning and will continue loading the rest of the page.

```php
require __DIR__ . '/../../database_credentials';
```

Use require for critical files. If PHP cannot find these files, the site would break irreparably. Files like database connections, credentials, or security settings are essential, and must always be included. If the file cannot be found, PHP will issue a fatal error and stop loading the page.

03_include_vs_require.php

Sometimes once is enough

Sometimes it's important to only load a file once. To do otherwise might cause fatal errors.

To load your functions file more than once, for example, would cause a fatal error, as functions would be declared more than once. In that case, use the include_once or require_once keywords to insure that PHP does not load the file a second time.

```
require(__DIR__ . '/inc/functions.md');
```

Instead of this

```
require_once(__DIR__ . '/inc/functions.md');
```

Do this

04_once_is_enough.php

Where to store files

For most sites and hosting situations, you should store partial files (includes) outside the public folder.

As a rule of thumb, only store files in the public folder that are meant to be loaded directly into the browser. These are files that can be reached by a standard URL. That would include css file, javascript files, images, and any php files the user might need to reached directly by the browsser.

http://example.com/about.php

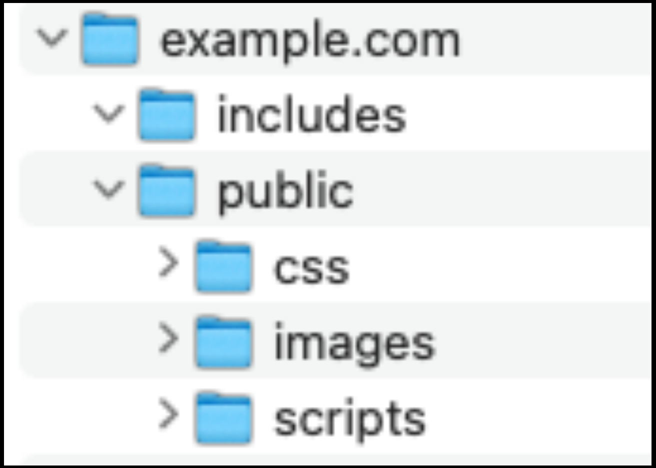
In this case, about.php must be in the public folder, as it must be reachable by web browsers.

However, any files included from the file system by PHP using include of require, should be stored outside the public folder.

04_once_is_enough.php

A common folder structure

For most simple PHP sites, the following folder structure should be adequate



```
example.com
    includes
    public
        css
        images
        scripts
```

In this example, the public folder is the DocumentRoot as you would define in in your vhosts file. It is the root folder that can be reached by the browser.

Next:

PHP and MySQL