

PHP Includes

</TITLE>

A presentation for CS 230

By Dustin Crossley

Designing web sites is

TOUGH

“...if you get a tough job, one that is hard, ...put a lazy man on it, and after 10 days he will have an easy way to do it...”

- Clarence Bleicher

Building a web page



*is a bit like putting together
a puzzle*

Put some pieces together

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
  </head>
  <body>
    <header>
      <nav></nav>
    </header>
    <div></div>
    <footer></footer>
  </body>
</html>
```

```
div {
  margin: 0;
}

header {
  background-color: blue;
}

footer {
  height: 100px;
}
```

... and you have a website!

What If?

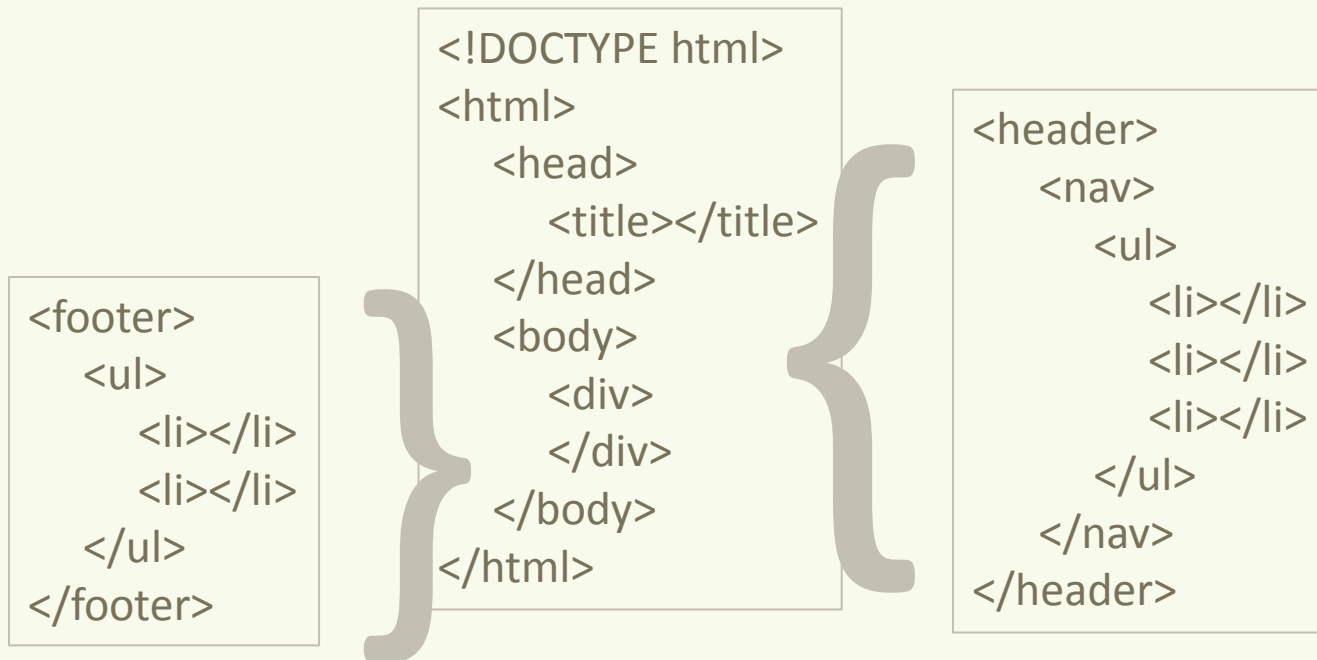
What if we could glue some of the pieces together?

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
  </head>
  <body>
    <div></div>
  </body>
</html>
```

```
<header>
  <nav>
    <ul>
      <li></li>
      <li></li>
      <li></li>
    </ul>
  </nav>
</header>
```

```
<footer>
  <ul>
    <li></li>
    <li></li>
  </ul>
</footer>
```

Creating these little *“modules”*



*would make it easier and faster to
put the puzzle together.*

This is what is known as

Modularization

Definition

modularize

[moj-uh-luh-rahz]

verb (used with object), modularized, modularizing.

- 1. to form or organize into modules, as for flexibility*
- 2. containing or consisting of modules*
- 3. produced in the form of modules*

Basic ideas

- *HTML code and structure doesn't change*
- *Smaller pieces of code saved to separate files*
- *Modules become reusable code snippets*
- *Main html file references the modules needed*

Simple, right?

Putting it together

What the files look like inside

index.php

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
  </head>
  <body>
    ? Header goes here ?
    <div>
      </div>
    </body>
  </html>
```

header.php

```
<header>
  <nav>
    <ul>
      <li></li>
      <li></li>
      <li></li>
    </ul>
  </nav>
</header>
```

Notice the header is missing from the index.php file

Need Glue

Now we have two files

index.php

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
  </head>
  <body>
    ? Header goes here ?
    <div>
      </div>
    </body>
</html>
```

header.php

```
<header>
  <nav>
    <ul>
      <li></li>
      <li></li>
      <li></li>
    </ul>
  </nav>
</header>
```

But how do we stick them together?

PHP

Is the answer

PHP makes it easy

Wait, what's PHP?

PHP – recursive acronym for

***P**HP: **H**ypertext **P**repocessor*

Originally “Personal Home Page”

PHP Details

PHP is a widely-used open source general-purpose scripting language

- *Original concept of PHP written in 1994*
- *Written in C programming language*

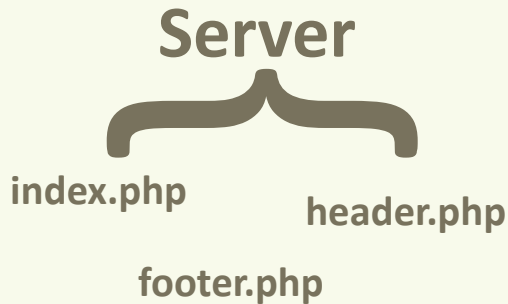
Compare to JavaScript

JavaScript	PHP
Code runs in the browser (client-side)	Code runs on the server (server-side)
Client receives the code and executes it to produce the results	Client receives only the results of running the script, not the code

PHP : How it works

A brief overview

1) PHP files are processed on the server,



(Files must end with the .php extension)

2) mashed together,



3) and a standard html file is sent to the client



The resulting php file contains no references to the php code!

PHP Code Block

statement

```
<?php include $_SERVER['DOCUMENT_ROOT'] . '/modules/header.php'; ?>
```

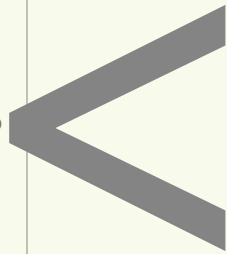
breakdown

<?php ?>	PHP code block
include	PHP function that tells the server to look for the php module to merge in
\$_SERVER['DOCUMENT_ROOT']	PHP “Super Global” object to determine the root of current website
.	Period – PHP concatenation symbol
'/modules/header.php'	Location (path) of the module, from the root of website
;	Semi-colon, tells the server this is the end of the statement, similar to JavaScript

Using PHP

Putting it together

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
  </head>
  <body>
    ? Header goes here ?
    <div>
    </div>
  </body>
</html>
```



```
<body>
  <?php include $_SERVER['DOCUMENT_ROOT'] . '/modules/header.php'; ?>
  <div>
  </div>
</body>
```

*After PHP server
processing it looks like:*

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
  </head>
  <body>
    <header>
      <nav>
        <ul>
          <li></li>
          <li></li>
          <li></li>
        </ul>
      </nav>
    </header>
    <div>
    </div>
  </body>
</html>
```

Modularization

SUMMARY

Making a Small Change

Imagine...

- *a site that has hundreds of pages*
- *making a small change to each one*
- *how fast the change in a single PHP module would be*

Modularization Benefits

- *Saves time – write code in common modules once*
- *Easier maintenance –change code in one place*
- *Saves money – saving time means saving money!*

Simple, right?

THANK YOU

</SLIDEDECK>

References

- <http://babel.hathitrust.org/cgi/pt?id=mdp.35112104266178;view=1up;seq=328>
- <http://www.merriam-webster.com/dictionary/modularized>
- <http://dictionary.reference.com/browse/modularization>
- <http://php.net>
- <http://noteandpoint.com/2010/06/up-to-speed-on-html-5-and-css-3-2/>
- <http://blainerobertson.net/content/?content=478>
- <http://pdfobject.com/>