

PHP Hypertext Preprocessor

CTEC1433/2025W

Internet of Things Applications

Overview

- PHP encompasses a programming language, libraries, and interpreter software.
- With Apache, PHP is built and installed as a plug-in module that interprets PHP code in files that typically have a **.php** extension.

Preprocessing

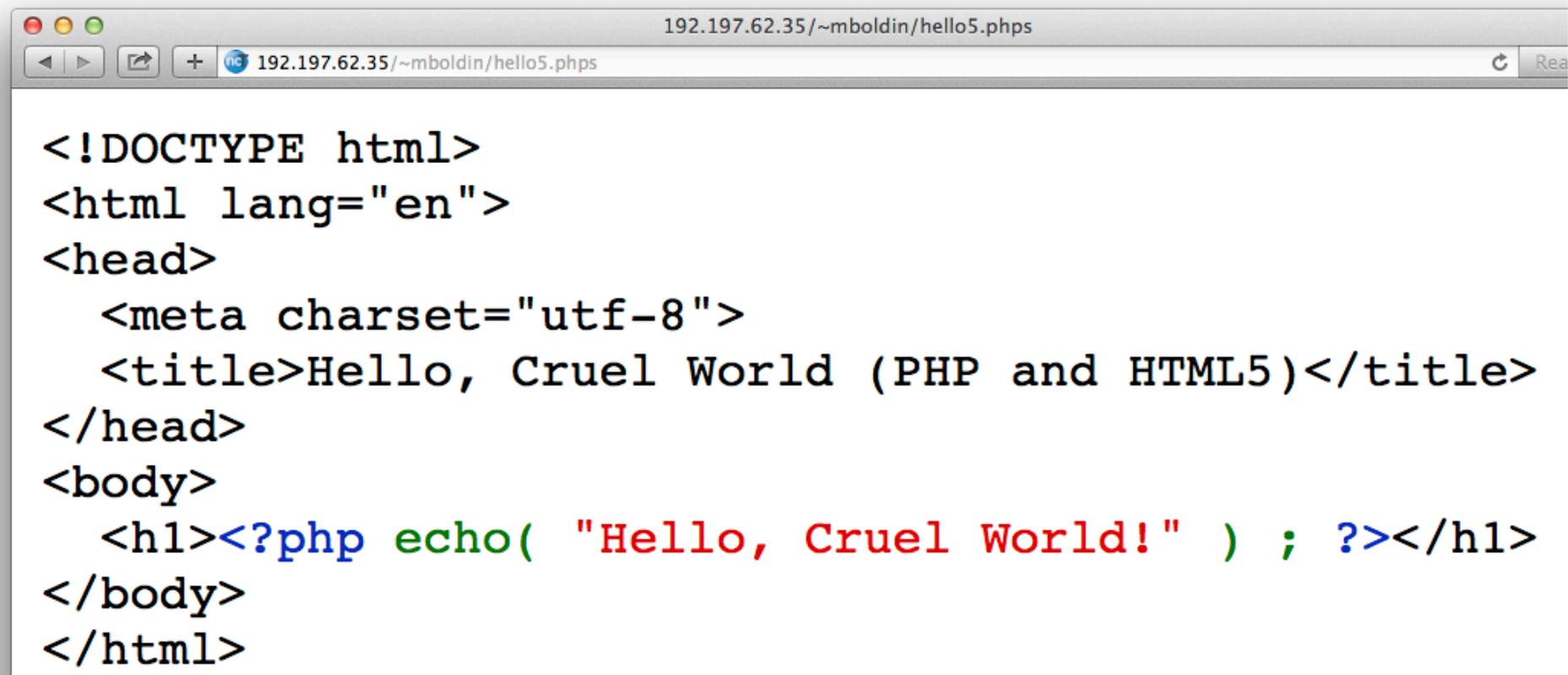
- All the PHP code is executed and replaced by its output, if any.
- Only web content is sent back to the browser.

Hello, Cruel World!

```
<?php  
echo( "Hello, Cruel World!\n" );  
?>
```

Hello, Valid HTML5 Cruel World!

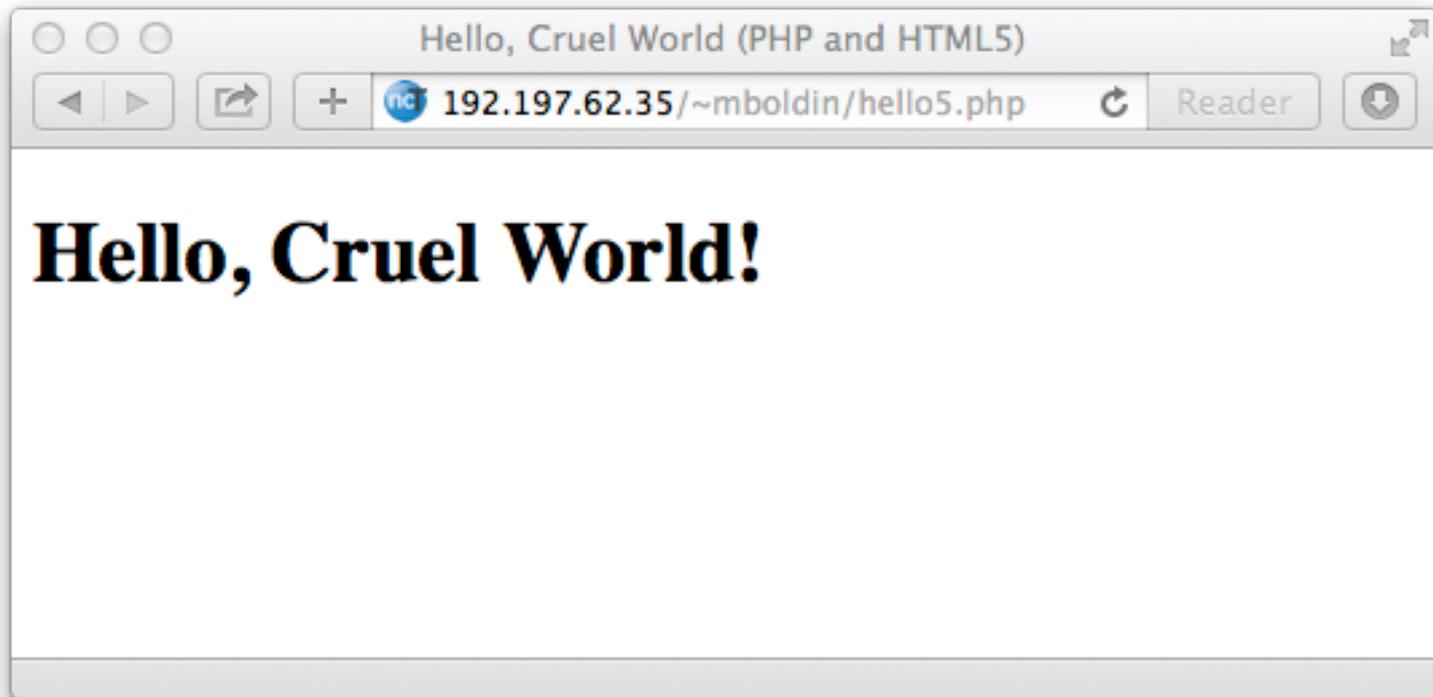
Source Code



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Hello, Cruel World (PHP and HTML5)</title>
</head>
<body>
  <h1><?php echo( "Hello, Cruel World!" ) ; ?></h1>
</body>
</html>
```

Hello, Valid HTML5 Cruel World!

Output

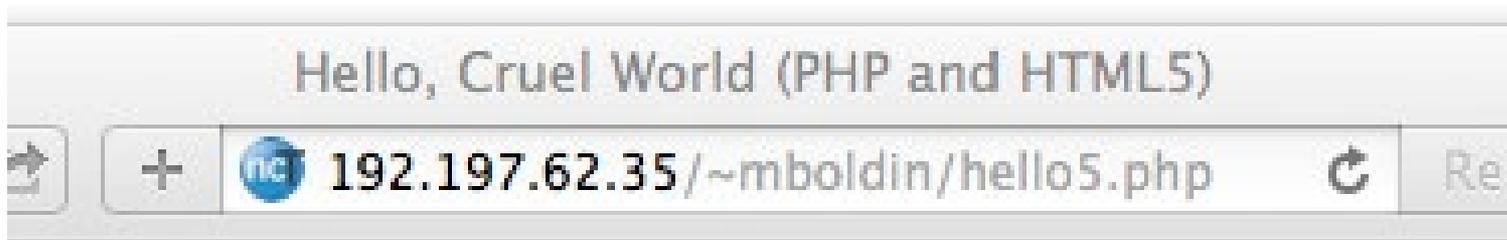


Hello, Valid HTML5 Cruel World!

Output -- View HTML Source

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <title>Hello, Cruel World (PHP and HTML5)</title>
6 </head>
7 <body>
8   <h1>Hello, Cruel World!</h1>
9 </body>
10 </html>
11
```

IMPORTANT! To run a PHP app...



If you are doing web apps, your PHP script must run from a web server – they are **server-side scripts**.

The web browser (i.e., *client side*) is used to view the **output** of each PHP script.

IMPORTANT! To run a PHP app... (2)

If you are doing web apps, your PHP script must run from a web server – they are **server-side scripts**.

- On your local computer (Windows PC, Mac or Linux PC), the easiest way to do this is to install **XAMPP**.
- Save or copy your PHP scripts to the **htdocs** folder (e.g., **C:\xampp\htdocs** on Windows).
- Use <http://localhost/> or <http://127.0.0.1/> to start your URLs.

To run a PHP app ... (3)

Sometimes, XAMPP's Apache server listens on ports other than the default port 80... see the "Using XAMPP" guide for details.

Drop In and Out

- You can switch from HTML5 and/or CSS to PHP and back *whenever you want . . .*
- This includes *inside* HTML5 tags!

```

```



PHP code
begins



PHP code
ends

Drop In and Out (2)

```
• • •  
<table>  
  <caption>Times tables</caption>  
  <tr>  
    <?php  
    for ( $i = 1 ; $i < 10 ; $i++ )  
    {  
      ?>  
      <th><?php echo( $i ) ; ?></th>  
      <?php  
      } // end for  
      ?>  
    </tr>  
  • • •
```

Drop In and Out (3)

- You can switch from HTML5 and/or CSS to PHP and back ***whenever you want . . .***

- `<?php // enter PHP mode`

```
/* write some PHP code -- on the same  
line, on a different line, or on  
multiple lines */
```

```
?> <!-- leave PHP mode -->
```

Environment Variables

- Use the `$_SERVER` array, e.g,

```
$docroot = $_SERVER[ 'DOCUMENT_ROOT' ] ;
```

- To get the name of the PHP script itself use

```
$_SERVER[ 'PHP_SELF' ]
```

Form Data

- To process form data use the `$_POST` array.

- HTML5:

```
<input type="text" name="address">
```

PHP:

```
$address = $_POST[ "address" ] ;
```

Connecting to MySQL

```
$dbp = new mysqli( "localhost", $user, $password, $database ) ;  
if ( $dbp->connect_errno )  
{  
    cgidie( "Unable to connect",  
           $dbp->connect_errno, $dbp->connect_error ) ;  
}
```

```
if ( ! $dbp->begin_transaction( ) )  
{  
    cgidie( "Can't begin SQL transaction",  
           $dbp->errno, $dbp->error ) ;  
}  
  
$res = $dbp->query( "SELECT " .  
                   "  bin, wine, producer, vintage, bottles " .  
                   "FROM " .  
                   "  cellar " ) ;
```

- <https://www.php.net/manual/en/book.mysqli.php>

Connecting to MySQL (2)

```
$nrows = $res->num_rows ;
```

```
for ( $i = 0 ; $i < $nrows ; $i++ )  
{  
    $row = $res->fetch_assoc( ) ;
```

```
$res->free( ) ;  
  
// close connection  
  
$dbp->close( ) ;
```

```
        $query = sprintf( "UPDATE cellar " .  
                        " SET bottles = %d " .  
                        " WHERE bin = %d ",  
                        $bottles, $bin ) ;  
    }  
    if ( ! $dbp->query( $query ) )  
    {  
        if ( ! $dbp->rollback( ) )  
        {  
            cgidie( "Can't rollback " .  
                  "SQL transaction",  
                  $dbp->errno, $dbp->error ) ;  
        }  
    }  
}
```

- <https://www.php.net/manual/en/book.mysql.php>

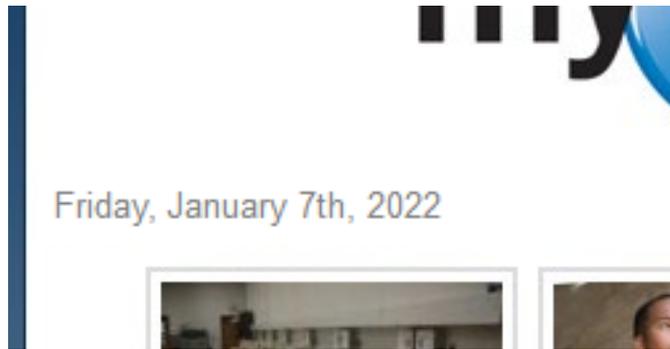
technology@niagarac.on.ca



January 30, 1997 to January 30, 2022

**“[SOME] THINGS I DO (IN PHP) ON THE
technology.niagaracollege.ca
WEB SITE”**

Today's Date (on the server)



```
p#date {  
    font-size: 10pt ;  
    color: gray ;  
}
```

```
<?php  
    date_default_timezone_set( "America/Toronto" ) ;  
    printf( "%s", date( "l, F jS, Y" ) );  
?>
```

The PHP code is contained inside a paragraph (<p>) element, whose CSS is also shown above.

- <https://www.php.net/manual/en/function.date-default-timezone-set.php>
- <https://www.php.net/manual/en/timezones.php>
- <https://www.php.net/manual/en/function.date.php>
- <https://www.php.net/manual/en/datetime.formats.php>
- <https://www.php.net/manual/en/function.printf.php>

When a page was last modified

```
function last_modified( $file, $suppress_heading = 0 )
{
    date_default_timezone_set( "America/Toronto" ) ;
    $modified_date = filemtime( $file ) ;
    /* Sat Jan  7 15:11:48 EST 2006 */
    if ( ( ! isset( $suppress_heading ) ) or
        ( $suppress_heading === 0 ) )
    {
        echo( "<em>Last modified:</em>&nbsp;" ) ;
    }
    echo( date( "D M d H:i:s T Y", $modified_date ) ) ;
}
```

```
last_modified( $_SERVER[ 'SCRIPT_FILENAME' ] ) ;
```

When a page was last modified

- This is a two-step process...
 - The first part is a function which, given a filename, retrieves its modification time from the O/S.

```
$modified_date = filemtime( $file ) ;
```

- The second part uses PHP environment variable, which contains the absolute pathname of the current PHP script (i.e., the web page), to pass to the function.

```
last_modified( $_SERVER[ 'SCRIPT_FILENAME' ] ) ;
```

When a page was last modified

- <https://www.php.net/manual/en/function.filemtime.php>
- <https://www.php.net/manual/en/reserved.variables.server.php>

Tracking user agents

```
$ua = $_SERVER[ 'HTTP_USER_AGENT' ] ;

$fp = fopen("/tmp/browser.log", "a");

if ( flock( $fp, LOCK_EX ) ) { // do an exclusive lock
    fprintf($fp, "%s\n", $ua ) ;
    flock( $fp, LOCK_UN ) ; // release the lock
}

fclose( $fp ) ;
```

The “User Agent” field is sent (or should be sent) by a web client/browser and provided to the PHP script by the web server.

Since many users can access a web page simultaneously, file locking is used. The file is opened for append and the user agent string is logged.

Tracking user agents

- <https://www.php.net/manual/en/function.fopen.php>
- <https://www.php.net/manual/en/function.flock.php>
- <https://www.php.net/manual/en/function.fprintf.php>
- <https://www.php.net/manual/en/function.fclose.php>

You Find More ...

- <https://www.php.net/manual/en/>
PHP Manual
- <https://www.w3schools.com/php/default.asp>
W3Schools PHP Tutorial
- <https://phptherightway.com>
PHP: The Right Way