Hello World

```html
<html>
<head><title>PHP Hello World</title></head>
<body>
<?php
    echo "<p>Hello World!</p>";
?></p>
</body>
</html>
```

Development Environment

- Text editor + Web Server + PHP
  - On CS3
    - `~/public_html/hello.php`
    - [http://cs3.calstatela.edu/~cs320stu31/hello.php](http://cs3.calstatela.edu/~cs320stu31/hello.php)
- Eclipse PHP IDE
  - [http://www.eclipse.org/pdt/](http://www.eclipse.org/pdt/)
  - Zend Debugger
    - [http://downloads.zend.com/pdt](http://downloads.zend.com/pdt)

PHP as a Programming Language

- Comments
- Literals
- Variables
- Operators
- Expressions
- Statements
- Functions
- Classes

Comments

- Script style: `//` single-line comment
- C++ style: `//` single-line comment
- C style:
  ```
  /* single-line comment */
  /* multiple-line comments */
  ```
Literals

- 123, 0123, 0x123
- 12.3
- "123", '123'
- true, false
- null

Heredoc Syntax for String

- A "Here document" Example:

```php
echo <<<HTML
Username: <input type="text" name="username" /> <br />
Password: <input type="password" name="password" /> <br />
<input type="submit" name="submit" value="Login" />
HTML;
```

Variables and Constants

- Starts with a $
- Variables names are case-sensitive
- Variable reference &
- Constant
define( name, value )
  E.g. define( TITLE, "PHP Test Page")

Example: Variable and Variable Reference

```php
$a = "hello";
$b = "world";
$c = $a . " " . $b;
echo $c;
```

Operators

- Arithmetic
  - +, -, *, /, %
- Assignment
  - =
    - +=, -=, *=, /=, %=, *=
  - .=
- Increment/decrement
  - ++, --
- Concatenation
- Comparison
  - ==, !=
  - >, >=, <, <=
- Logical
  - &&, ||, !
- Conditional
  - ?:

Example: Operators

```php
$a = "hello";
$b = "world";
if( $a < $b ) echo "hello &lt; world <br ";
$c = $a + " " + $b;
echo $c;
$c = $a . " " . $b;
echo $c;
$a .= $b;
```
Control Statements

- **Branch**
  - if
  - if ... else
  - if ... elseif ... else

- **Switch**
  - switch

- **Loop**
  - while
  - do ... while
  - for
  - foreach

- **Break and continue**
  - break
  - continue

Example: Switch Statement

```php
switch( $fruit )
{
    case "apple":
        echo "The fruit is an apple";
        break;
    case "pear":
        echo "The fruit is a pear.";
        break;
    default:
        echo "The fruit is not an apple or a pear.";
}
```

Arrays

- **Associative arrays**, like *Map* in Java
  - <key, value>
  - Key must be either integer or string

Example: Arrays

```php
$courses = array("cs320" => "web", "cs122" => "SQL");
$strings["a"] = "abc";
$strings[1] = "def";
# when key is omitted
$numbers1 = array(1, 2, 3, 4, 5);
$numbers2 = array(1 => 1, 2 => 2, 3, 4, 5 => 5);
$strings[] = "xyz";
$strings[] = "uvw";
```

Example: Access Array Elements

```php
$numbers1[2] = "<br>";
echo $numbers1["cs320"] = "<br>";
foreach( $courses as $key => $value )
echo "$key, $value <br>";
foreach( $numbers as $n )
echo "$n <br>";
for( $i = 0 ; $i < count($numbers) ; ++$i )
echo $numbers[$i];
```

Some Useful Array Functions

- **Remove an element**
  - unset()
- **Number of elements**
  - count(), sizeof()
- **Sort**
  - By value: asort(), arsort()
  - By key: ksort(), krsort()
Example: Array Functions

```php
foreach($strings as $key => $value)
  echo $key, " ", $value, "<br>";
asort($strings);
foreach($strings as $key => $value)
  echo $key, " ", $value, "<br>";
ksort($strings);
foreach($strings as $key => $value)
  echo $key, " ", $value, "<br>";
```

Functions

```php
function sum( $op1, $op2 )
{
  return $op1 + $op2;
}

function say_hello()
{
  echo "hello."
}
```

About Functions

- Functions can be defined pretty much anywhere in the code
  - E.g. in other functions or in an if statement
- Functions have global scope
- No function overloading, and no way to undefine or redefine a function
- Support **variable functions**

Example: Variable Function

```php
function sum ( $a, $b)
{
  return $a + $b;
}
$v = "s"."u"."m";
echo $v(1,3);
```

Basic Class Syntax

```php
class GuestBookEntry {
  private $name;
  public  $comment;
  function __construct( $name, $comment )
  {
    $this->name = $name;
    $this->comment = $comment;
  }
  function getName() { return $this->name; }
  function setName( $name ) { $this->name = $name; }
}
```

Using Objects

```php
$entry = new GuestBookEntry( "cysun", "hello" );
echo $entry->getName(), " says ", $entry->comment;
```
Include Other Files
- `include()`, `include_once()`
- `require()`, `require_once()`

PHP as a Web Programming Language
- Handle HTTP requests
- Generate HTTP response
- Session tracking
- Database access

Example: GuestBook without Database
- Store an array of `GuestBookEntry` in session scope — there is no application scope in PHP

Pre-defined Variables
- `$_GET`
- `$_POST`
- `$_REQUEST`
- `$_COOKIE`
- `$_SESSION`
- `$_SERVER`

Access Request and Session Variables
```
if (! empty($_REQUEST["submit"]))
{
    $name = $_SESSION["name"]; 
    if (empty($name))
    {
        $name = $_REQUEST["name"];  
        $_SESSION["name"] = $name;
    }
}
```

Override Default Settings
```
- php.ini
  - display_errors = Off
  - error_reporting = E_ALL
  - session.auto_start = 0

- code
  - ini_set("display_errors", 1);
  - error_reporting(E_ERROR);
  - session_start(); = 0
```

Example: GuestBook with Database

- Using MySQL database
  ```
  $connection = mysql_connect("localhost", "cs320stu31", "abcd");
  mysql_select_db( "cs320stu31", $connection );
  $result = mysql_query( "select * from guest_book" );
  $entries = array();
  while( $row = mysql_fetch_array($result) )
    $entries[] = new GuestBookEntry( $row["name"], $row["comments"] );
  mysql_close( $connection );
  ```

Handle Query Results

- $result $row
- `mysql_fetch_array()`
- Variations of `mysql_fetch_array()`
  - Fetch a row as an associative array
    - `mysql_fetch_assoc($result)`
  - Fetch a row as an integer-indexed array
    - `mysql_fetch_row($result)`

More MySQL Functions


About PHP

- Advantages
  - Easy to learn
  - Easy to use
  - Dynamic typing
  - Web-friendly language features
- Disadvantages??

Resources

- [www.php.net](http://www.php.net)