college CD

Server-side technologies (TCS)

Class 1

PHP language

PHP is the acronym of Hypertext Preprocessor. Inspired by C, Java and Perl languages, it is an open source script language to be used within HTML document in order to develop applications and make them dynamic.

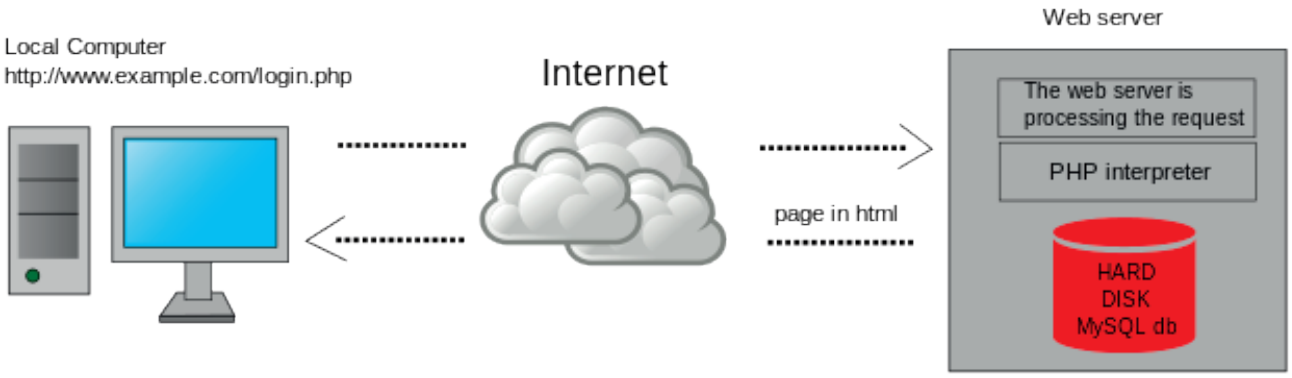
One important specificity of this script language is that it isn't executed on client-side, but on server side and it is used in different circumstances : messaging, blogs, e-commerce, online banking and publications, search engines, etc. Everything you want, actually, but gaming.

Static vs dynamic documents

A static document is one directly written using HTML language. JavaScript can be added to the document so it would make it look dynamic, but this can only be executed on client-side using local data. Server-side languages (Microsoft/IIS, Perl, PHP, etc.) must be used on server-side for any heavier treatments requesting database access, complex searches, etc.

On top of offering more possibilities, server-side languages such as PHP never have to rely on the user's computer or software's speed, making the documents much more efficient. The scripts are interpreted and executed on the server so it only sends client the results in the form of a simple HTML document.

Using PHP will make it possible for you to create rapidly and quite easily dynamic web documents, especially in conjunction with a relational database web server such as MySql.



Technical aspects

Firstly, in order for PHP codes to be interpreted and executed, the web server must be able to understand it. The hosting server must then have PHP module activated (or installed).

Secondly, you wont be able to see the result of your PHP coding while working offline using only a web browser since PHP needs to be interpreted. A virtual PHP web server (devserver) will have to be installed to your computer such as EasyPHP (Windows), wampserver (Windows), mamp (Apple), BigApache, etc.

PHP syntax

As a script language different from other languages in used such as Perl or C, instead of writing a program made of several program lines to generate an HTML document, you here have to write a HTML document which includes PHP coded in order to produce specific actions.

PHP is an embedded language, meaning that it can be placed anywhere in a HTMNL document, so coding must be placed into tags making it possible to the server to recognize and interpret it.

```
<p>
        <h1> My title</h1>
        <?php some PHP coding ?>
        some more HTML...
</p>
```

As soon as some PHP coding has beed implemented into an HTML document, this document's extension must be changed to php. For instance, from *index.html* to *index.php*.

Commands : echo and print

Echo and Print commands allow to show content on screen. In the following example, both PHP commands would output the sentence « Hello World » on screen.

```
<html>
<head>
<title>Titre</title>
</head>
<body>
<?php
echo "Hello World !";      // each statement ends with a semicolon
?>

<?php
print ("Hello World");     // print can take only one parameter
?>

<?php
echo "Hello"," World!";   // echo can take multiple parameters
?>
</body>
</html>
```

Special characters

Just like it is done with JavaScript, special characters sometimes need to be used in strings.

Code	Result
\n	New line
\r	Return
\t	Tab
\\	Backslash
\$	$
\"	Double quotes

The following code would produce an error message:

```
<?php
echo 'statement's comment';
// The apostrophe would be confused with the end of the string
?>
```

The statement should use a special character (\):

```
<?php
echo 'statement\'s comment';
?>
```

Commenting your code

Here again, comments are written just like it is done in Javascript:

```
<?php
// This is a single line comment

/* This is
a multiple lines
comment */
?>
```

Working with variables

Variables are names used to store data. They start with the symbol «$» followed by a name which can be a mix of letters, numbers and/or some symbols. Variable can be of many types (string, number, matrix, boolean, object...)

Variables can be named using any letters and numbers as well as some symbols and it may also start with an underscore. Although, it must never start with a number or include spaces, and remember PHP is case sensitive.

Declaring and assigning values to variables

Assigning a value to a variable is done using the «=» operator followed by the value of various natures.

```
<?php
$myVariable = "Hello world!";      // string

$x = 5;                            // integers and floats
$y = 7.5;
?>
```

Output data to screen

Two commands may be used to output data to screen : *echo*, *print and var_dump()*.

```
<?php
$name = "John";
echo "Hello ";              // outputs the word « Hello »
echo $name;                 // outputs the value of the variable (John)
echo " !";                  // outputs the character « ! »
?>

OR

<?php
$name = "John";
print "<p>Hello ";          // both commands can output HTML tags
var_dump($name);            // string(4) "John"
print " !</p>";
?>
```

Concatenation

Concatenation makes it possible to output series of data within a single argument. Instead of using multiple echo or print command lines, like in the preceding example, several data can be output using simple periods.

Concatenating variables

```
<?php
$a = "Hello ";
$b = "world!";
echo $a . $b;
?>
```

Result :

Hello world!

Concatenating variables and strings

```
<?php
$firstName = "John";
$lastName = "Smith";
echo "Hello " . $firstName . " " . $lastName. " !";
?>
```

Result :
Hello John Smith !

Adding text to a string using « .= »

```
<?php
$a = "John";
$a .= " Smith";
echo $a;
?>
```

Result :
John Smith

Using single or double quotes?

Using double quotes with *echo* or *print* allows you to include a variable so its value appears in the string to output.

Example :

```
echo "Hello  $firstName $lastName !";
```

Useful string functions

strtolower : Converts string characters to lower case.
strtoupper : Converts string characters to upper case.

```
<?php
echo strtolower("HELLO WORLD!");
echo strtoupper( "hello world!");
?>
```

strlen : Counts and returns a string's number of characters (including spaces).

```
<?php
echo strlen("This is an example.");
?>
```

Result :
19

str_word_count : Counts and returns the number of words of a string.

```
<?php
echo str_word_count ("This is an example of a long sentence.");
?>
```

Result :
8

substr : Return part of the string using three parameters : the string to be shortened, the position of the starting point, the number of characters to be returned.

```
<?php
$myVariable = "This is an example of a sentence that will be shorten.";
echo substr($myVariable,5,16);
?>
```

Result :
is an example of

str_replace : Replaces a specified string values in a given string using three parameters : the text to be replaced, the replacement text and the text that is analyzed.

```
<?php
echo str_replace ("This", "Here", "This is an example.");
?>
```

Result :
Here is an example.

str_strpos : Locates and return the position of a suite of character(s) within a string.

```
<?php
$myVariable = "This is an example";
echo strpos($myVariable,"ex");
?>
```

Result :
11

ucfirst : Makes the first character of a string an upper case.
lcfirst : Makes the first character of a string an upper case.

```
<?php
$myVariable = "this is an example";
echo ucfirst($myVariable);                    // This is an example
?>
```

OR

```
<?php
echo lcfirst("THIS IS AN EXAMPLE");      // tHIS IS AN EXAMPLE
?>
```

wordwrap : Wraps a string to a given number of characters.

```
<?php
nl2br();
$a = "An example: I wonder how it will look like using word wrap fun
tion on this text...";
echo wordwrap($a,15,"<br>\n");
?>
```

Other functions

chop()	Removes whitespace or other characters from the right end of a string.
chr()	Returns a character from a specified ASCII value.
ltrim()	Removes whitespace or other characters from the left side of a string.
rtrim()	Removes whitespace or other characters from the right side of a string.
str_shuffle()	Randomly shuffles all characters in a string.
strcasecmp()	Compares two strings (case-insensitive).
strcmp()	Compares two strings (case-sensitive).
strrev()	Reverses a string.
trim()	Removes whitespace or other characters from both sides of a string.

Heredoc syntax

The heredoc syntax allows you to, somehow, use preformatted text content including line breaks. It can be used either to echo a string or to associate its content to a variable.

```
<?php
$myVariable = <<<EOD    // any suite of characters you want
<h1>Using heredoc syntax is pretty cool</h1>
<p>
It layouts things the way you want!
</p>
EOD;                    // same suite needs to be used both places
?>
```

Assignment 1 : Your first PHP script

In a HTML document, use PHP to obtain the following :

The user will write a sentence as a value of a predefined variable.

The output should look like this :

The sentence you entered is :
This is my sentence

It is composed of :

- 4 different words
- 19 characters (including spaces)

Note : *Your program should make sure the final output is in lower case with the first letter in upper case.*