Web Technologies

PHP introduction

David Jelenc

April 2017

Index

Index

PHP programming

- PHP: Hypertext Preprocessor

PHP programming

- PHP: Hypertext Preprocessor
- Procedural and object-oriented (some functional concepts, too)

PHP programming

- PHP: Hypertext Preprocessor
- Procedural and object-oriented (some functional concepts, too)
- Weakly and dynamically typed

PHP programming

- PHP: Hypertext Preprocessor
- Procedural and object-oriented (some functional concepts, too)
- Weakly and dynamically typed
- Scripted, interpreted (no compiling required)

PHP programming

- PHP: Hypertext Preprocessor
- Procedural and object-oriented (some functional concepts, too)
- Weakly and dynamically typed
- Scripted, interpreted (no compiling required)
- Designed to dynamically generate web pages

PHP programming

- PHP: Hypertext Preprocessor
- Procedural and object-oriented (some functional concepts, too)
- Weakly and dynamically typed
- Scripted, interpreted (no compiling required)
- Designed to dynamically generate web pages
- Transparent to clients: runs entirely on the server

PHP programming

- PHP: Hypertext Preprocessor
- Procedural and object-oriented (some functional concepts, too)
- Weakly and dynamically typed
- Scripted, interpreted (no compiling required)
- Designed to dynamically generate web pages
- Transparent to clients: runs entirely on the server
- `http://www.php.net`

Beginning with PHP

A PHP script is a series of commands, each terminated with a semicolon. The interpreter executes only those commands that are within PHP tags. All other text is displayed in verbatim.

Beginning with PHP

A PHP script is a series of commands, each terminated with a semicolon. The interpreter executes only those commands that are within PHP tags. All other text is displayed in verbatim.

PHP tag

```
<?php helloWorld(); ?>
```

Beginning with PHP

A PHP script is a series of commands, each terminated with a semicolon. The interpreter executes only those commands that are within PHP tags. All other text is displayed in verbatim.

PHP tag

```
<?php helloWorld(); ?>
```

A few PHP peculiarities

Beginning with PHP

A PHP script is a series of commands, each terminated with a semicolon. The interpreter executes only those commands that are within PHP tags. All other text is displayed in verbatim.

PHP tag

```
<?php helloWorld(); ?>
```

A few PHP peculiarities

- PHP ignores all spaces within the PHP code,

Beginning with PHP

A PHP script is a series of commands, each terminated with a semicolon. The interpreter executes only those commands that are within PHP tags. All other text is displayed in verbatim.

PHP tag

```
<?php helloWorld(); ?>
```

A few PHP peculiarities

- PHP ignores all spaces within the PHP code,
- function names (e.g. `echo()` or `ECHO()`) and other language constructs (e.g. `if` or `IF`) ar **case-insensitive**

Beginning with PHP

A PHP script is a series of commands, each terminated with a semicolon. The interpreter executes only those commands that are within PHP tags. All other text is displayed in verbatim.

PHP tag

```
<?php helloWorld(); ?>
```

A few PHP peculiarities

- PHP ignores all spaces within the PHP code,
- function names (e.g. `echo()` or `ECHO()`) and other language constructs (e.g. `if` or `IF`) ar **case-insensitive**
- variable names are **case sensitive**: `$var` and `$Var` are two distinct variables.

A mandatory Hello World example

```php
<!doctype html>
<title>Hello world!</title>
<meta charset="UTF-8">

<?php
# Inline comment type 1
// Inline comment type 2

echo ("Hello ");  # Using echo with parentheses
echo "World!";   // Using echo without parentheses

/* Multi-line
comment */
?>
```

Listing 1: PHP - Hello world in PHP

Running PHP scripts

- Directly by using the command prompt (command-line interface, CLI):

Running PHP scripts

- Directly by using the command prompt (command-line interface, CLI):
  - Open the terminal and write

    ```
    $ php script-name.php
    ```
  - Directly from an IDE (by clicking the *play* button, MS Visual Studio Code plugin: *code-runner*)

Running PHP scripts

- Directly by using the command prompt (command-line interface, CLI):
  - Open the terminal and write

    ```
    $ php script-name.php
    ```
  - Directly from an IDE (by clicking the *play* button, MS Visual Studio Code plugin: *code-runner*)
- Indirectly by sending a an HTTP request to the server (common gateway interface, CGI):

Running PHP scripts

- Directly by using the command prompt (command-line interface, CLI):
  - Open the terminal and write

    ```
    $ php script-name.php
    ```
  - Directly from an IDE (by clicking the *play* button, MS Visual Studio Code plugin: *code-runner*)
- Indirectly by sending a an HTTP request to the server (common gateway interface, CGI):
  1. We send an HTTP request to a PHP script that is accessible to the web server,
  2. The server will parse the request and run the script.
  3. Finally, the server will return an HTTP response containing the script output.

Running PHP scripts

- Directly by using the command prompt (command-line interface, CLI):
  - Open the terminal and write

    ```
    $ php script-name.php
    ```
  - Directly from an IDE (by clicking the *play* button, MS Visual Studio Code plugin: *code-runner*)
- Indirectly by sending a an HTTP request to the server (common gateway interface, CGI):
  1. We send an HTTP request to a PHP script that is accessible to the web server,
  2. The server will parse the request and run the script.
  3. Finally, the server will return an HTTP response containing the script output.
- Using the interactive console:

  ```
  $ php -a
  ```

A request-response cycle in HTTP

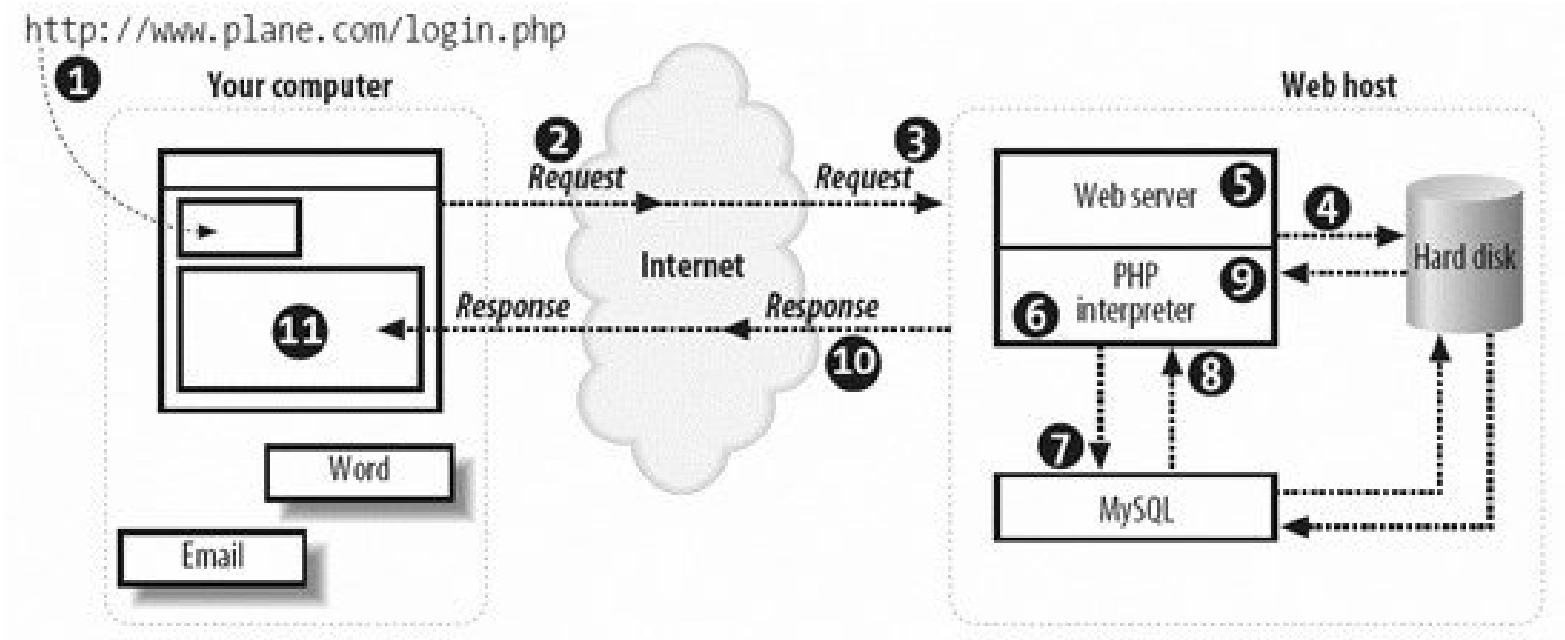

Figure: Processing HTTP requests with PHP

Working with functions

```php
<?php
function printHello($name="World") {
  echo "Hello " . $name;
}
?><!doctype html>
<title>Working with functions</title>
<meta charset="UTF-8">
<p>
  We can alternate between PHP and HTML seamlessly:
  <?php printHello("Donald"); ?>
</p>
```

Listing 2: PHP - Functions in PHP

Commonly-used functions: `http://www.php.net/quickref.php`

Primitive and complex data types

Type	Meaning
Integer	Positive, 0 and negative numbers
Double	Real numbers
Boolean	Boolean values: either **true** or **false**
NULL	Dedicated data type whose value can only be **NULL**
String	Used to represent char and text sequences
Array	An associative array, a dictionary-like structure
Object	User-defined class
Resource	(Deprecated) A reference to an external resource (e.g. data-base connection or a file handle)

Data types and operators

```php
<?php
$var1 = 3;
$var1 = "Three"; // Dynamic types
$var1 = 3 + "7"; // Weak types, $var1 = 10

$var1 = "Web " . "technologies";
// $var1 = "Web technologies"

$var1 = 'PHP';
$var2 = 5;
$var3 = $var2 + 1;
$var2 = $var1;

echo($var1); // "PHP"
echo($var2); // "PHP"
echo($var3); // 6
?>
```

Listing 3: PHP - Variables and operators

Data types and operators

```php
<?php

echo(' Learning ' . $var1);
// Concatenation

echo(" Learning $var1");
// String interpolation when using double quotes
// Output:" Learning PHP"

echo(' Learning $var1');
// No string interpolation when using single quotes
// Output:" Learning $var1".

?>
```

Listing 4: PHP - Variables and operators

Complex data types: Associative array

```php
<?php
$data = ['test', 1, 'two']; # Array creation
# Alternative, using the old syntax
$data = array('test', 1, 'two');

echo($data[0]);   # "test"
echo($data[1]);   # 1

$data[1] = 'one'; # Replace the value at index 1
$data[3] = 3.14;  # Add another element
$data[] = 'WT';   # Append another element at the end
echo($polje[4]);  # "WT"
?>
```

Listing 5: PHP - An ordinary array

Complex data types: Associative array

```php
<?php
$capitals = [];
$capitals["Slovenia"] = "Ljubljana";
$capitals["Croatia"] = "Zagreb";
$capitals["Germany"] = "Berlin";

echo('Capital of Slovenia is " . $capitals["Slovenia"]);

foreach ($capitals as $country => $capital) {
  echo("The capital of $country is $capital\n");
}
?>
```

Listing 6: PHP - An associative array

Complex data types: Associative array

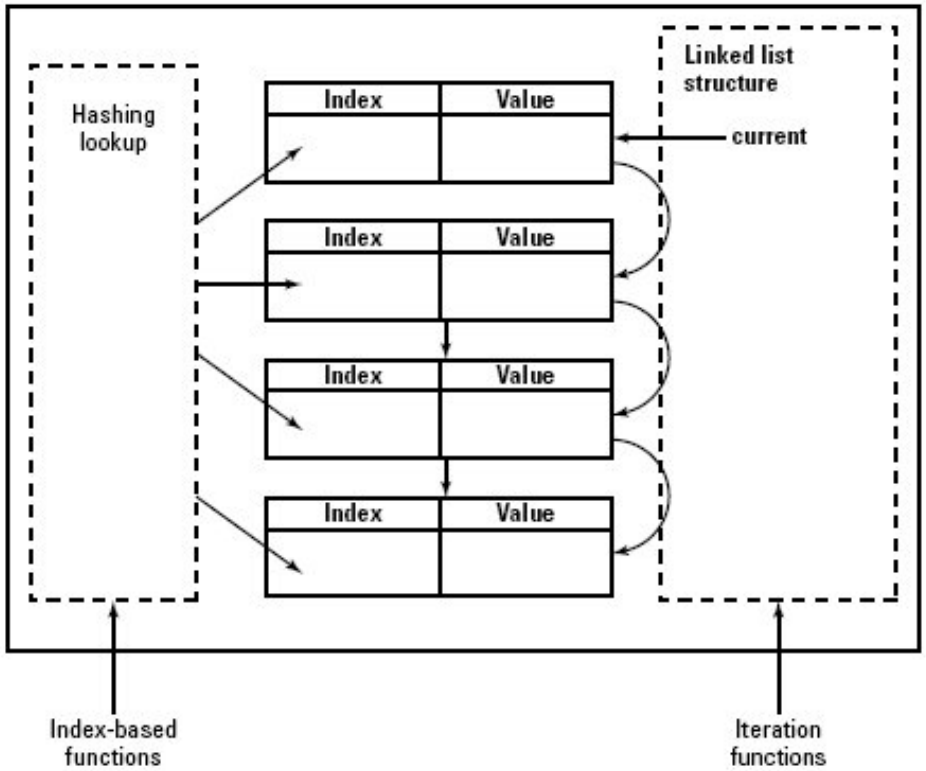

Figure: Associative array is both a map and a link-list

Complex data types: Classes

```php
class BaseClass {
    function __construct() {
        echo "The basic constructor.";
    }
}

class SubClass extends BaseClass {
    public $var = "A default value";

    function __construct() {
        parent::__construct();
        echo "Constructor inside the extended class";
    }

    public function displayVar() {
        echo $this->var;
    }
}
```

Listing 7: PHP - Defining a class

Complex data types: Classes

```php
<?php
$obj = new SubClass();
$obj->displayVar();
?>
```

Listing 8: PHP - Using a class

Branching

```php
<?php
if ($number % 2 == 0) {
  echo "The number is even.";
} else {
  echo "The number is odd.";
}
?>
```

Listing 9: PHP - Branching with if–statements

Loops: WHILE and FOR

```php
$count = 1;
while ($count <= 10) {
  echo("$count \n");
  $count++;
}

for ($i = 0; $i < 10; $i++) {
  echo("$i \n");
}
```

Listing 10: PHP - Looping with WHILE and FOR

Index

Osnovno o protokolu

- Hypertext Transfer Protocol

Osnovno o protokolu

- Hypertext Transfer Protocol
- The workhorse of the Worlde Wide Web – WWW:

Osnovno o protokolu

- Hypertext Transfer Protocol
- The workhorse of the Worlde Wide Web – WWW:
  - Typically over TCP and port 80

Osnovno o protokolu

- Hypertext Transfer Protocol
- The workhorse of the Worlde Wide Web – WWW:
  - Typically over TCP and port 80
  - Text-based, but also allows binary contents

Osnovno o protokolu

- Hypertext Transfer Protocol
- The workhorse of the Worlde Wide Web – WWW:
  - Typically over TCP and port 80
  - Text-based, but also allows binary contents
- Client-server model:

Osnovno o protokolu

- Hypertext Transfer Protocol
- The workhorse of the Worlde Wide Web – WWW:
  - Typically over TCP and port 80
  - Text-based, but also allows binary contents
- Client-server model:
  - A client is typically an Internet browser (Firefox)

Osnovno o protokolu

- Hypertext Transfer Protocol
- The workhorse of the Worlde Wide Web – WWW:
  - Typically over TCP and port 80
  - Text-based, but also allows binary contents
- Client-server model:
  - A client is typically an Internet browser (Firefox)
  - A server is typically a dedicated web server (Apache2)

Osnovno o protokolu

- Hypertext Transfer Protocol
- The workhorse of the Worlde Wide Web – WWW:
  - Typically over TCP and port 80
  - Text-based, but also allows binary contents
- Client-server model:
  - A client is typically an Internet browser (Firefox)
  - A server is typically a dedicated web server (Apache2)
- An HTTP session is a a sequence of request-response pairs between the client and the server.

Osnovno o protokolu

- Hypertext Transfer Protocol
- The workhorse of the Worlde Wide Web – WWW:
  - Typically over TCP and port 80
  - Text-based, but also allows binary contents
- Client-server model:
  - A client is typically an Internet browser (Firefox)
  - A server is typically a dedicated web server (Apache2)
- An HTTP session is a a sequence of request-response pairs between the client and the server.
  - The client builds an HTTP request,

Osnovno o protokolu

- Hypertext Transfer Protocol
- The workhorse of the Worlde Wide Web – WWW:
  - Typically over TCP and port 80
  - Text-based, but also allows binary contents
- Client-server model:
  - A client is typically an Internet browser (Firefox)
  - A server is typically a dedicated web server (Apache2)
- An HTTP session is a a sequence of request-response pairs between the client and the server.
  - The client builds an HTTP request,
  - and sends it to the server over TCP/80.

Osnovno o protokolu

- Hypertext Transfer Protocol
- The workhorse of the Worlde Wide Web – WWW:
  - Typically over TCP and port 80
  - Text-based, but also allows binary contents
- Client-server model:
  - A client is typically an Internet browser (Firefox)
  - A server is typically a dedicated web server (Apache2)
- An HTTP session is a a sequence of request-response pairs between the client and the server.
  - The client builds an HTTP request,
  - and sends it to the server over TCP/80.
  - The server parses the request, constructs the response and returns it to the client over the same socket (TCP/80)

Osnovno o protokolu

- Hypertext Transfer Protocol
- The workhorse of the Worlde Wide Web – WWW:
  - Typically over TCP and port 80
  - Text-based, but also allows binary contents
- Client-server model:
  - A client is typically an Internet browser (Firefox)
  - A server is typically a dedicated web server (Apache2)
- An HTTP session is a a sequence of request-response pairs between the client and the server.
  - The client builds an HTTP request,
  - and sends it to the server over TCP/80.
  - The server parses the request, constructs the response and returns it to the client over the same socket (TCP/80)
  - An HTTP session may consist of several request-response cycles.

An HTTP Request message

An HTTP Request message

- User enters the URL in browser
  `http://www.example.com/index.html`

An HTTP Request message

- User enters the URL in browser
  `http://www.example.com/index.html`
- The browser constructs the HTTP request

An HTTP Request message

- User enters the URL in browser
  `http://www.example.com/index.html`
- The browser constructs the HTTP request

An HTTP request message

```
GET /index.html HTTP/1.1
Host: www.example.com
```

An HTTP Request message

- User enters the URL in browser
  `http://www.example.com/index.html`
- The browser constructs the HTTP request

An HTTP request message

```
GET /index.html HTTP/1.1
Host: www.example.com
```

- and sends it to the server over TCP/80.

An HTTP Response message

An HTTP Response message

The server *parses* the incoming HTTP request,

An HTTP Response message

The server *parses* the incoming HTTP request, *computes and constructs* the HTTP response,

An HTTP Response message

The server *parses* the incoming HTTP request, *computes and constructs* the HTTP response, and returns it over the same socket (TCP/80) to the client.

An HTTP Response message

The server *parses* the incoming HTTP request, *computes and constructs* the HTTP response, and returns it over the same socket (TCP/80) to the client.

HTTP response message

```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Etag: "3f80f-1b6-3e1cb03b"
Accept-Ranges: bytes
Content-Length: 438
Connection: close
Content-Type: text/html; charset=UTF-8

<!-- HTML code -->
```

HTTP methods

HTTP has multiple methods (or types of requests)

HTTP methods

HTTP has multiple methods (or types of requests)

- GET,

HTTP methods

HTTP has multiple methods (or types of requests)

- GET,
- HEAD,

HTTP methods

HTTP has multiple methods (or types of requests)

- GET,
- HEAD,
- POST,

HTTP methods

HTTP has multiple methods (or types of requests)

- GET,
- HEAD,
- POST,
- PUT, DELETE, TRACE, OPTIONS, CONNECT, PATCH

HTTP methods

HTTP has multiple methods (or types of requests)

- GET,
- POST,
- GET and POST are most commonly used.