An overview of PHP

Cristian Bogdan

What is PHP

- PHP: Hypertext Processor (or Personal Home Page?)
- This lecture goes through the doc: http://www.php.net/manual/en/
- Interpreted language
  - Easy to get something done fast
  - Problems when size and complexity grow
  - Performance penalty (though addressed in later Zend engine)
- A number of PHP organisations and hosting services around the net.
- A huge catalogue of features developped by volunteers
- A wide user community definately happy that they not only can program, but they also can program for the internet…
- Language in evolution… PHP 5 adds exception handling. OOP is also an add-on since PHP4, PHP5 perfects it
  - most OOP concepts and keywords are like in java: class, extends, interface, private, protected, etc
  - exceptions are like in Java (try/catch, etc)

What can PHP do

- CGI scripting of course
  - Based on escapes within HTML pages <?php > or <% … %>
- Command line scripting
- GUI applications (PHP-GTK) http://gtk.php.net/

Installation

- Install the PHP interpreter separately. There is good support for this on Linux
- In the webserver configuration, associate the php extension with the PHP interpreter
- For serious applications you will need a database engine.
- Apache is the typical choice for the web server.
  - Integrated as a module, so no supplementary processes are created (in CGI, typically there is one process per access, which is very expensive)
- Mysql is the typical db engine

PHP in HTML

- The most often case where PHP is used
- Escaping
  - <?php …. ?>
  - <? …. ?>
  - <script language="php" > ….</script>
  - <% .... %>, <%=. … %> like ASP and JSP
- Like in JSP, escaping can be interrupted to write some HTML

```
<?php
    if ($expression) {
    ?>
 <b>This is true.</b>
<?php }  ?>
```

- You can see the evolution under community pressure, here and in other areas

Types

- a variable name begins with $, no type declaration (type declaration can be required by passing some settings to the interpreter)
- can have one type first, another one later… (BASIC??)
- boolean, integer (similar to C), float (similar to C)
- string
  - single-quoted, no character escapes
  - double-quoted, like C character escapes
  - <<< EOD notation to write longer multiline strings
  - $a.$b appends the string b and at the end of the string a (+ in Java)
  - $a(index1, index2) gives a substring
  - string functions in the function library (strlen() like in C)
  - expression intergration "some text { expression $var } blabla"
- arrays are mappings between keys and values (Dictionary/Hashtable/Map in java)
  - $arr = array("foo" => "bar", 12 => true)
- There are automactic type conversions between types (very dangerous…). Explicit type conversions exist too

Other types

- Classes and objects, OOP
- Resources, a kind of reference
- Pseudo-types, a kind of #typedef ?

Variables

- See Types
- Assignment by value (not by reference)
- Programming for the lazy
- Lots of predefined variables, especially related to HTTP/CGI
  - _SERVER, _GET, _POST, _COOKIE, _FILES, _REQUEST, _SESSION
- Scope of variables, globals
- Variable variables ☺
- External varialbes, useful for forms
- Functions as variables

Other procedural stuff

- Operators similar to C
- Statements similar to C, plus:
  - <? if(…) : ?> ….<? endif; >
  - foreach () through arrays, just values, or also keys
    - foreach (array_expression as $value) statement
    - foreach (array_expression as $key => $value) statement
- Code inclusion with require() and include()
- Conditional function definition (similar to C #ifdef)

Features

- HTTP authentication, cookies, file uploads
- Remote files (like java.net.URLConnection)
- Connections (like java.net.Socket)
- Persistent db connections
  - Normally db connections are defined with engine specific functions as external resources. Each access would open its connection, that is expensive

Functions

- Array functions, calendar fuctions, date functions, character functions, IO functions, printer, file/directory, etc
- Functions for protocols/standards, e.g. FTP, HTTP, URL, LDAP, IRC, Mail, NSAPI, popmail, XML/XSL, Bzip2/Zip/Zlib
- Functions for various databases (Mysql, Oracle, MS SQL server, mSQL, PostgreSQL, SQLite, dBase), dbx is general
- Functions for other systems/tools: e.g. Apache, COM, Cyrus, PDF, dBase, DBM, DOM, .NET, Lotus Notes, GNU readline, Hyperware

Conclusions

- Easy to learn from Java or C
- CGI parameters, HTTP sessions, etc are easy to recognize
- Good language/system for doing something small fast (but then JSP/ASP do most of the same)
- Not a wise choice for a serious/large project due to the lack of type safety, lack of OOP in the libraries, etc.
  - Experienced PHP people confirm that larger projects tend to become a big mess due to the freedoms that seemed so good in the begining, which make programmers lazy
- The array concept is nice, but its name is misleading (array means something very different in all the rest of Computer Science)
- Still, a very good choice for pragmatists who want to get the job done.