ISSN NO: 2770-4491
Date of Publication:20-06-2022

PHP is one of the main tools for creating a Web page in computer science lessons.

Shoykulova Dilorom Kudratovna, teacher of school №339, Tashkent
Sh.Q. Shoyqulov, Karshi State University

Annotation. The article discusses the main characteristics and advantages of one of the most popular web programming languages PHP.

Keywords: Web technology, browser, object model, programming, Web-site, system, technology.

Introduction

PHP (Hypertext Preprocessor) is a widely used open source general purpose programming language. PHP is specially designed for web development and its code can be embedded directly into HTML.

PHP, as everyone knows, is actually the successor to a product called PHP/FI. Created in 1994 by Rasmus Lerdorf, the very first incarnation of PHP was a simple set of CGI scripts written in the C programming language. Initially using them to track visits to his web resume, he called this set of scripts "Personal Homepages Tools", but more commonly referred to as "PHP Tools". As time went on, more and more functionality improvements were needed, and Rasmus rewrote PHP Tools to create a larger and richer implementation. This new implementation was capable of interacting with databases and more, creating a framework with which users could create simple dynamic web applications such as guest books. In June 1995, Rasmus "opened the PHP Tools source code to the public, allowing developers to use it as they saw fit. It also gave users the opportunity to fix bugs in the code and improve it.

In September of that year, Rasmus expanded PHP and dropped PHP from the name for a short time. Now in the form of an FI tool (short for "Form Interpreter"), the new implementation included some of the core functionality of the PHP we know today. It had Perl-like variables, automatic form interpretation, and HTML-embedded syntax. The syntax of the language was similar to Perl, although much more limited, simpler, and somewhat inconsistent. In order to embed code into an HTML file, developers had to use HTML comments. Although this method was not well received, FI was still gaining popularity as a CGI tool, but still not as a language. However, the change began the following month, when in October 1995 Rasmus released a complete rewrite of the code. With the return of the name PHP, but shortened from "Personal Home Page Construction Kit", it was the first release to boast an extended scripting interface. The language intentionally resembled C in structure, making it easy to understand for developers familiar with C, Perl, and similar languages. While still limited to UNIX and POSIX compliant systems, the issue was explored for implementing the language on Windows NT.

The code received another major overhaul in April 1996. Combining the names of previous versions, Rasmus introduced PHP/FI. The second generation implementations began to truly evolve PHP from a toolbox into a programming language in its own right. PHP included built-in support for DBM, mSQL and Postgres95 databases, cookies, support for user-defined functions, and more. PHP/FI was upgraded to version 2.0 in June. Interestingly, however, there was only one version of PHP 2.0. When it finally passed beta status in November 1997, the language parser had already been completely rewritten.

Results And Discussions

Despite its short development history, PHP/FI has continued to gain popularity in the young world of web development. In 1997 and 1998, PHP/FI became a cult favorite for several thousand users around the world. Research by Netcraft in May 1998 showed that nearly 60,000 domains were transmitting headers containing "PHP". This number was approximately 1% of all domains on the Internet at the time. Despite these impressive numbers, development of PHP/FI was limited: despite a few minor contributors, it was still generally developed by one person.

Here is a simple PHP code example:

ISSN NO: 2770-4491
Date of Publication:20-06-2022

```
<html>
   <head>
      <title>Example</title>
   </head>
   <body>
      <?php
      echo "Study programming in PHP!";
      ?>
   </body>
</html>
```

To output HTML code with language commands, the PHP script contains HTML with inline code. PHP code is separated by special <?php and ?> start and end tags that allow you to "switch" into and out of "PHP mode". PHP differs from JavaScript in that PHP scripts run on the server and generate HTML that is sent to the client. If you had a script like the one above hosted on your server, the client would only get the result of executing it, but would not be able to figure out exactly what code produced it. You can even set up your server so that regular HTML files are processed by the PHP processor, so that clients don't even know if they're getting a regular HTML file or the result of a script. PHP is easy to learn, but at the same time able to satisfy the needs of professional programmers. The features of PHP consist of a long list. You can get started quickly and within hours you can create simple PHP scripts. Although PHP is primarily designed to run in a web server environment, its scope is not limited to just that.

In an HTML file, when PHP processes the file, it looks for opening and closing tags such as <?php and ?> that tell PHP when to start and stop processing the code in between. This kind of processing allows PHP to be embedded in all sorts of different documents, since anything outside of a pair of start and end tags will be ignored by the PHP parser.

PHP includes the short <?= echo tag, which is shorthand for the more verbose <?php echo. Short tags are available by default, but they can be disabled using the short_open_tag directive in the php.ini configuration file, or disabled by default if PHP was compiled with the --disable-short-tags option.

If the file contains only PHP code, it is preferable to omit the end tag at the end of the file. This helps to avoid adding random whitespace or newline characters after the PHP closing tag, which can cause unwanted effects since PHP starts outputting data to the buffer without the programmer's intention to output any data at that point in the script.

Anything outside of a pair of opening and closing tags is ignored by the PHP interpreter, which has the ability to handle files with mixed content. This allows PHP code to be embedded in HTML documents, for example to create templates.

```
<p>This will be ignored by PHP and displayed by the browser.</p>
<?php echo 'This will be processed.'; ?>
<p>This will also be ignored by PHP and displayed by the browser.</p>
```

This works as expected, because when the PHP interpreter encounters the ?> end tags, it simply starts outputting whatever it finds (except the immediately following newline character - see section splitting instructions) until it encounters another start tag, except with a conditional statement contained within the code, in which the interpreter determines the result of the condition before deciding what to skip.

Like C or Perl, PHP requires a semicolon to end statements at the end of each statement. The closing tag of a PHP code block automatically applies a semicolon; those. there is no need to put a semicolon at the end of the last line of the PHP code block. The end tag of a block will "absorb" the immediately following newline, if one is encountered.

PHP supports ten simple types.

4 scalar types:

- bool
- int
- float (floating point number, also known as double)

• string

4 mixed types:

• array
• object
• callable
• iterable

2 special types:

• resource
• NULL

Typically, the programmer does not set the type of a variable; PHP usually does this at runtime, depending on the context in which the variable is used. Variables in PHP are represented by a dollar sign followed by the variable name. The variable name is case sensitive. Variable names follow the same rules as other naming conventions in PHP. A valid variable name must begin with a letter or underscore and consist of any number of letters, numbers, and underscores. This can be displayed with a regular expression: ^[a-zA-Z_\x80-\xff][a-zA-Z0-9_\x80-\xff]*$

```
<?php
        $var = 'ali';
        $Var = 'vali';
        echo "$var, $var"; // print "Ali, Vali"
        $4site = 'error'; // nocorrect; starts with a number
        $_4site = 'correct'; // correct; starts with an underscore
?>
```

Expressions are the most important building blocks of PHP. Almost everything you write in PHP is an expression. The simplest and most accurate definition of an expression is "anything that matters".

The main forms of expressions are constants and variables. If you write "$a = 5", you are assigning '5' to $a. '5' obviously has the value 5, or in other words, '5' is an expression with the value 5 (in this case, '5' is an integer constant). After this assignment, you expect the value of $a to also be 5, so if you wrote $b = $a, you assume that it will work the same as if you wrote $b = 5. In other words, $a it is also an expression with a value of 5. If everything works correctly, then this is exactly what will happen.

An operator is something that takes one or more values (or expressions in programming jargon) and evaluates to a new value (thus the whole construct can be treated as an expression). Operators can be grouped by the number of values they take. Unary operators take only one value, such as ! (logical negation operator) or ++ (increment). Binary operators take two values; these are, for example, the familiar arithmetic operators + (plus) and - (minus), most of the operators supported in PHP fall into this category. And finally, there is only one ternary operator, ? :, which takes three values, is usually referred to as "ternary operator" (although "conditional operator" might be a more accurate name).

Any PHP script consists of a sequence of instructions. An instruction can be an assignment, a function call, code repetition (a loop), a comparison, or even an instruction that does nothing (an empty statement). The statement is usually followed by a semicolon. In addition, statements can be combined into blocks by enclosing them in curly braces. An instruction block is also an instruction in itself. This section describes the different types of instructions.

PHP is a powerful programming language and interpreter that interacts with a web server either as a module or as a standalone binary CGI application. PHP is capable of accessing files, executing various commands on the server, and opening network connections. That is why all scripts executed on the server are potentially dangerous. PHP was originally developed as a more secure (relative to Perl, C) language for

ISSN NO: 2770-4491
Date of Publication:20-06-2022

writing CGI applications. With a range of compile-time and dynamic application settings, you can always find the right combination of freedom and security.

Since there are many different ways to use PHP, there are also many options that control its behavior. The wide range of options ensures that you can use PHP for a variety of purposes, but it also means that certain combinations of options make the server insecure.

The flexibility of PHP configuration can be compared to the flexibility of the language itself. PHP can be used to create full-fledged server applications that use the operating system's capabilities available to a specified user, or as simple include files stored on a server with minimal risk in a tightly controlled environment. How this environment is set up, as well as the quality of its security, largely depends on the PHP developer.

One of PHP's biggest strengths is how it works with HTML forms. The key here is that each form element is automatically made available to your PHP programs. Here is an example of an HTML form:

Example #1. The simplest form of HTML

```
<form action="action.php" method="post">
    <p>First Name: <input type="text" name="name" /></p>
    <p>Last Name: <input type="text" name="age" /></p>
    <p><input type="submit" /></p>
</form>
```

There is nothing special about this form. This is a normal HTML form without any special tags. When the user fills out the form and clicks the submit button, the action.php page will be called. This file might contain something like:

Example #2. Outputting form data

```
Hello, <?php echo htmlspecialchars($_POST['name']); ?>.
    your is <?php echo (int)$_POST['age']; ?> years old.
```

result:

```
Hello Ali. yours 30 years old.
```

Conclusions

If you do not take into account the pieces of code with htmlspecialchars () and (int), the principle of operation of this code should be simple and clear. htmlspecialchars() ensures that "special" HTML characters are properly encoded so that malicious HTML or Javascript is not inserted into your page. The field age, which we know must be a number, we can simply convert to int, which will automatically get rid of unwanted characters. PHP can also do this automatically with the filter module. The $_POST['name'] and $_POST['age'] variables are automatically set for you by PHP. We previously used the $_SERVER superglobal, but here we also use the $_POST superglobal, which contains all the POST data. Note that the submit method of our form is POST. If we were to use the GET method, then our form information would be in the $_GET superglobal variable. Alternatively, you can use the $_REQUEST variable if the data source is irrelevant. This variable contains a mix of GET, POST, COOKIE data.

References

1. Shoyqulov Sh. Q. Adobe Flash dasturida maktabgacha ta’lim tarbiyalanuvchilari va boshlang‘ich sinf o‘quvchilari uchun interfaol o‘quv materiallari tayyorlash. O‘quv- uslubiy qo‘llanma, Nasaf, QarshiDU: 2012 (Shoykulov Sh. K. Create interactive learning materials for preschoolers and elementary school pupils in Adobe Flash. Educational-methodical manual, Nasaf, Karshi SU: 2012).

2. Shoyqulov Sh. Q. Adobe Flash dasturida maktabgacha ta’lim tarbiyalanuvchilari va boshlang‘ich sinf o‘quvchilari uchun animatsion- rivojlantiruvchi o‘yin loyihalari. O‘quv- uslubiy qo‘llanma, Nasaf, QarshiDU: 2014 (Shoykulov Sh. K. Create animated and educational game projects for preschoolers and elementary school pupils in Adobe Flash. Educational-methodical manual, Nasaf, KarshiSU: 2014).
3. The text is of the main components of multimedia technologies, Academicia Globe: Inderscience Research, 3(04), 573–580. URL. https://agir.academiascience.org/index.php/agir/article/view/684 , ISSN: 2776-1010, SJIF: 5.653
4. The graphics- is of the main components of multimedia technologies, Web of Scientist: International Scientific Research Journal (WoS), ISSN:2776-0979. URL. https://wos.academiascience.org/index.php/wos/article/view/1427, Vol. 3 No. 4 (2022): wos, Published: 2022-04-04, 1373-1381 p.
5. The Audio- Is of the Main Components of Multimedia Technologies International Journal on Integrated Education, e-ISSN : 26203502, p-ISSN : 26153785, Vol. 5 No. 5 (2022): IJIE, Published: May 3, 2022, p. 263-268. URL. https://journals.researchparks.org/index.php/IJIE/article/view/3092
6. Yormatov I.T., YuldashevaN.A., Toshpulatov I.A. (2020). Issues of electronic trade development in Uzbekistan. ISJ Theoretical & Applied Science, 12 (92), 211-215. Soi: http://s-o-i.org/1.1/TAS-12-92-40
7. Doi: https://dx.doi.org/10.15863/TAS.2020.12.92.40
8. М. А. Максудов, И. Т. Ёрматов, & Ж. А. Максудов (2019). К вопросам эффективного использования финансовых ресурсов на предприятии. Экономика и бизнес: теория и практика, (6-2), 57-61. doi: 10.24411/2411-0450-2019-10860
9. URL: https://cyberleninka.ru/article/n/k-voprosam-effektivnogo-ispolzovaniya-finansovyh-resursov-na-predpriyatii/viewer
10. Ерматов, И. Т. (2017). Актуальные проблемы повышения инновационного потенциала Узбекистана. *Актуальные проблемы гуманитарных и социально-экономических наук*, *5*(11), 70.
11. URL: https://www.elibrary.ru/item.asp?id=28938038
12. Қамбаров, Жамолиддин, & Ёрматов, Илмидин (2022). ОЛИЙ ТАЪЛИМ СИФАТНИ ОШИРИЩДА АВТОМАТЛАШТИРИЛГАН ВА МОНИТОРИНГ ТИЗИМИНИНГ ИМКОНИЯТЛАРИДАН ФОЙДАЛАНИШ. Nazariy va amaliy tadqiqotlar xalqaro jurnali, 2 (1), 41-50. doi: 10.5281/zenodo.6090134
13. URL:https://cyberleninka.ru/article/n/oliy-talim-sifatni-oshirishda-avtomatlashtirilgan-va-monitoring-tizimining-imkoniyatlaridan-foydalanish/viewer
14. Исманов Иброҳим, & Ёрматов Илмидин (2021). ФАРҒОНА ПОЛИТЕХНИКА ИНСТИТУТИДА СИФАТЛИ ТАЪЛИМНИ РИВОЖЛАНТИРИШ МАСАЛАЛАРИ. Nazariy va amaliy tadqiqotlar xalqaro jurnali, 1 (2), 39-50. doi: 10.5281/zenodo.5752006
15. URL: https://cyberleninka.ru/article/n/far-ona-politehnika-institutida-sifatli-talimni-rivozhlantirish-masalalari/viewer
16. Ерматов, И. Т. Некоторые проблемы модернизации промышленных предприятий / И. Т. Ерматов // Научно-технический прогресс: социальные, технические и общественные факторы : Сборник статей. – Москва : ООО "ИМПУЛЬС", 2018. – С. 133-137. – EDN XTALBJ.
17. URL: https://www.elibrary.ru/item.asp?id=35245208
18. Tolibov, I. Sh. U. The concept of building management by influence of innovation on change of organizational structure and production structure of the enterprise / I. Sh. U. Tolibov, I. T. Yormatov // Theoretical & Applied Science. – 2018. – No 9(65). – P. 256-263. – DOI 10.15863/TAS.2018.09.65.43. – EDN YKWXXV.
19. URL: https://www.elibrary.ru/item.asp?id=36265703