МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

НАБЕРЕЖНОЧЕЛНИНСКИЙ ИНСТИТУТ

СИНТАКСИС ЯЗЫКА PHP

Учебно-методическое пособие
по дисциплине
«ВЕБ-ПРОГРАММИРОВАНИЕ»

Набережные Челны
2018

Галиуллин Л.А. Синтаксис языка PHP: учебно-методическое пособие по дисциплине «Веб-программирование» [Электронный ресурс] / Казанский федеральный университет, Электронный архив, 2018.

Рассматривается серверный язык создания сценариев PHP. Представлены некоторые преимущества PHP. Приведены контрольные вопросы. Для студентов направлений подготовки «Информатика и вычислительная техника», «Программная инженерия».

Введение

PHP — это серверный язык создания сценариев (или стороны сервера), разработанный специально для Web. В HTML-страницу можно внедрить код PHP, который будет выполняться при каждом ее посещении. Код PHP интерпретируется Web-сервером и генерирует HTML или иной вывод, наблюдаемый посетителем страницы.

Разработка PHP была начата в 1994 г. и вначале выполнялась одним человеком, Расмусом Лердорфом (Rasmus Lerdorf). Этот язык был принят рядом талантливых людей и претерпел три основных редакции, пока не стал широко используемым и зрелым продуктом, с которым мы имеем дело сегодня. К январю 2001 г. он использовался почти в пяти миллионах доменов во всем мире и их число продолжает быстро расти. Количество доменов, в которых в настоящее время используется PHP, можно выяснить на странице http://www.php.net/ usage.php. PHP — это продукт с открытым исходным кодом (Open Source). У пользователя имеется доступ к исходному коду. Его можно использовать, изменять и свободно распространять другим пользователям или организациям.

Первоначально PHP являлось сокращением от Personal Home Page (Персональная начальная страница), но затем это название было изменено в соответствии с рекурсивным соглашением по наименованию GNU (GNU = Gnu's Not Unix) и теперь означает PHP Hypertext Preprocessor (Препроцессор гипертекста PHP).

В настоящее время основной версией PHP является четвертая. Адрес начальной страницы для PHP — http://www.php.net.

Что нового в PHP

Имеется ряд важных усовершенствований 4 версии:

- PHP работает значительно быстрее предшествующих версий, поскольку в нем используется новый механизм Zend Engine. Если требуется еще более высокая производительность, по адресу http://www.zend.com можно

получить модули Zend Optimizer, Zend Cache или Zend Compiler.

- PHP всегда можно было использовать как эффективный модуль сервера Apache. С появлением новой версии PHP можно устанавливать и в виде модуля ISAPI для Internet Information Server компании Microsoft.
- Теперь поддержка сеансов является встроенной. В предшествующих версиях для управления сеансом или создания собственного сеанса требовалось устанавливать дополнительный модуль PHPLib.

Некоторые преимущества PHP

К числу конкурентов PHP относятся Perl, Active Server Pages (ASP) от Microsoft, Java Server Pages (JSP) и Allaire Cold Fusion.

PHP обладает множеством преимуществ по сравнению с этими продуктами:

- Высокая производительность
- Наличие интерфейсов ко многим различным системам баз данных
- Встроенные библиотеки для выполнения многих общих задач, связанных с Web
- Низкая стоимость
- Простота изучения и использования
- Переместимость
- Доступность исходного кода

Производительность

PHP исключительно эффективен. Используя единственный недорогой сервер, можно обслуживать миллионы обращений в день. Результаты тестирования, опубликованные компанией Zend Technologies (http://www.zend.com), подтверждают более высокую производительность PHP по сравнению с конкурирующими продуктами.

Интеграция с базами данных

PHP обладает встроенной связностью со многими системами баз данных. В дополнение к MySQL, в числе прочих можно непосредственно подключаться к базам данных PostgreSQL, mSQL, Oracle, dbm, Hyperware, Informix, InterBase и Sybase.

Используя Open Database Connectivity Standard (Стандарт открытого интерфейса связи с базами данных, ODBC), можно подключаться к любой базе данных, для которых существует ODBC-драйвер. Это распространяется на продукты Microsoft и многих других компаний.

Встроенные библиотеки

Поскольку PHP был разработан для использования в Web, он имеет множество строенных функций для выполнения широкого разнообразия полезных, связанных с Web, задач. С его помощью можно "на лету" генерировать GIF-изображения, подключаться к другим сетевым службам, отправлять сообщения электронной почты, работать с cookie-наборам и генерировать PDF-документы посредством нескольких строк кода.

Стоимость

Пакет PHP является бесплатным. Наиболее новую версию можно в любой момент совершенно бесплатно выгрузить из http://www.php.net

Изучение PHP

Синтаксис PHP основывается на других языках программирования, в первую очередь на C и Perl, Если вы уже знакомы с C, Perl или C-подобным языком, таким как C++ или Java, то почти сразу сможете эффективно использовать PHP.

Переносимость

Пакет PHP можно использовать под управлением многих

различных операционных систем. Код PHP можно создавать в среде таких бесплатных Unix-подобных операционных систем, как Linux и FreeBSD, коммерческих версий Unix типа Solaris и IRIX или различных версий Microsoft Windows. Как правило, программы будут работать без каких-либо изменений в различных средах с установленным PHP.

Исходный код

Пользователь имеет доступ к исходному коду PHP. В отличие от коммерческих закрытых программных продуктов, если нужно что-либо изменить или добавить в этом языке, то это всегда можно сделать. Не следует дожидаться, пока фирма-изготовитель выпустит правки (патчи). Нет необходимости беспокоиться о том, что изготовитель собирается покинуть рынок или перестанет поддерживать продукт.

Синтаксис и грамматика

Синтаксис PHP заимствован непосредственно из C. Java и Perl также повлияли на синтаксис данного языка.

Переход из HTML

Есть три способа выхода из HTML и перехода в "режим PHP кода":

1. <? echo("простейший способ, инструкция обработки SGML\n"); ?>
2. <?php echo("при работе с XML документами делайте так\n"); ?>
3. <script language="php">
   echo ("некоторые редакторы (подобные FrontPage) не любят обрабатывающие инструкции");
   </script>;
4. <% echo("От PHP 3.0.4 можно факультативно применять ASP-тэги"); %>

Разделение инструкций

Инструкции (утверждения) разделяются также как в C или Perl - точкой с запятой. Закрывающий тэг (?>) тоже подразумевает конец утверждения, поэтому следующие записи эквивалентны:

```
<php
   echo "Это тест";
?>
<php echo "Это тест" ?>
```

Типы переменных

PHP поддерживает переменные следующих типов:

- integer - целое
- double - число с дробной частью
- string - строковая переменная
- array - массив
- object - объектная переменная
- pdfdoc - PDF-документ (только при наличии поддержки PDF)
- pdfinfo - PDF-инфо (только при наличии поддержки PDF)

Тип переменной обычно не устанавливается программистом; вместо этого, он определяется PHP во время выполнения программы, в зависимости от контекста, в котором она используется. Если вам нравится указывать тип переменной непосредственно, используйте для этого инструкцию cast либо функцию settype(), но учтите, что переменная может вести себя по-разному, в зависимости от того, какой тип определен для нее в данное время.

Инициализация переменной

Для инициализации переменной в PHP вы просто присваиваете ей значение. Для большинства переменных это именно так; для массивов и объектных переменных, однако, может использоваться несколько иной механизм.

Инициализация массивов

Массив может инициализироваться одним из двух способов: последовательным присвоением значений, или посредством конструкции array().

При последовательном добавлении значений в массив вы просто записываете значения элементов массива, используя пустой индекс. Каждое последующее значение будет добавляться в качестве последнего элемента массива.

```
$names[] = "Jill";   // $names[0] = "Jill"
$names[] = "Jack";   // $names[1] = "Jack"
```

Как в C и Perl, элементы массива нумеруются, начиная с 0, а не с 1.

Инициализация объектов

Для инициализации объектной переменной используйте новое предписание для сопоставления данного объекта объектной переменной.

```
class foo {
   function do_foo () { echo "Doing foo."; }
}
$bar = new foo;
$bar -> do_foo ();
```

Область переменной

Областью переменной является контекст, внутри которого она определена. Обычно все переменные PHP имеют одну область. Однако, внутри функций определенных пользователем, представлена локальная область функции. Любая переменная, определенная внутри функции, по умолчанию ограничена локальной областью функции. Например:

```
$a = 1; /* глобальная область */
Function Test () {
   echo $a; /* ссылка на переменную локальной области */
```

```
}
Test ();
```

Этот скрипт не выдаст что-либо на выходе, поскольку инструкция echo относится к локальной версии переменной $a, значение которой присваивается не внутри этой области. Вы можете заметить, что здесь имеется некоторое отличие от языка C, в том, что глобальные переменные в C автоматически действуют и внутри функций, если только они не переписываются локальными определениями. Это может вызвать некоторые проблемы, т.к. по неосторожности можно изменить глобальную переменную. В PHP глобальные переменные должны быть продекларированы глобально внутри функции, если предполагается их использование в данной функции. Например:

```
$a = 1;
$b = 2;
Function Sum () {
   global $a, $b;
$b = $a + $b;
}
Sum ();
echo $b;
```

Этот скрипт выдаст значение "3". Поскольку $a и $b декларируются глобально внутри функции, ссылки на данные переменные трактуются как ссылки на их глобальные версии.

Вторым способом доступа к переменным из глобальной области является использование специального определяемого PHP массива $GLOBALS. При этом предыдущий пример может быть записан как:

```
$a = 1; $b = 2;
Function Sum () { $GLOBALS["b"] = $GLOBALS["a"] + $GLOBALS["b"]; }
Sum ();
echo $b;
```

Массив $GLOBALS является ассоциативным, в котором имя глобальной переменной является ключом, а значение этой переменной является значением элемента массива.

Другой важной характеристикой от области определения

переменной является статическая переменная. Статическая переменная существует только в локальной области функции, но она не теряет своего значения, когда программа, при исполнении, покидает эту область. Рассмотрим следующий пример:

```
Function Test () {
   $a = 0;
   echo $a;
   $a++;
}
```

Эта функция совершенно бесполезна практически, поскольку каждый раз при ее вызове она устанавливает $a в 0 и выводит "0". Выражение $a++ так же бесполезно, поскольку при выходе из функции переменная $a исчезает. Для придания дееспособности функции подсчета переменная $a декларируется как статическая:

```
Function Test () {
   static $a = 0;
   echo $a;
   $a++;
}
```

Теперь при вызове функции Test() она будет выводить значение $a и увеличивать его.

Статические переменные также весьма существенны, когда функции вызываются рекурсивно. Рекурсивные функции - это те, которые вызывают сами себя. Составлять рекурсивную функцию нужно внимательно, т.к. при неправильном написании можно сделать рекурсию неопределенной. Вы должны быть уверены в адекватности способа прекращения рекурсии. Следующая простая функция рекурсивно считает до 10:

```
Function Test () {
   static $count = 0;
   $count++;
   echo $count;
   if ($count < 10) {
      Test ();
   }
   $count--;}
```

Изменяемые переменные

Иногда бывает удобно давать переменным изменяемые имена. Такие имена могут изменяться динамически. Обычная переменная устанавливается так:

$a = "hello";

Изменяемая переменная берет некое значение и обрабатывает его как имя переменной. В приведенном выше примере значение hello может быть использовано как имя переменной, посредством применения двух записанных подряд знаков доллара, т.е.:

$$a = "world";

С этой точки зрения, две переменных определены и сохранены в символьном дереве PHP: $a с содержимым "hello" и $hello с содержимым "world". Так, инструкция:

echo "$a ${$a}";

осуществляет то же самое, что и инструкция:

echo "$a $hello";

а именно, обе они выводят: hello world.

Чтобы использовать изменяемые переменные с массивами, решите проблему неоднозначности. Это означает, что если вы пишете $$a[1], то синтаксическому анализатору необходимо знать, имеете ли вы в виду использовать $a[1] как переменную, или вы предполагаете $$a как переменную, а [1] как индекс этой переменной. Синтаксис для разрешения неоднозначности такой: ${$a[1]} для первого случая и ${$a}[1] для второго.

Переменные вне PHP

HTML Формы (GET и POST)

Когда программой-обработчиком формы является PHP-скрипт, переменные этой формы автоматически доступны для данного скрипта PHP. Например, рассмотрим следующую форму:

```
<form action="foo.php" method="post">
   Name: <input type="text" name="name"><br>
   <input type="submit">
```

```
</form>
```

При активизации формы PHP создаст переменную $name, значением которой будет то содержимое, которое было введено в поле Name: данной формы.

PHP также воспринимает массивы в контексте переменных формы, но только одномерные. Вы можете, например, группировать взаимосвязанные переменные вместе или использовать это свойство для определения значений переменных при множественном выборе на входе:

```
<form action="array.php" method="post">
   Name: <input type="text" name="personal[name]"><br>
   Email: <input type="text" name="personal[email]"><br>
   Beer: <br>
   <select multiple name="beer[]">
      <option value="warthog">Warthog
      <option value="guinness">Guinness
      </select>
   <input type="submit">
</form>
```

Если PHP-атрибут track_vars включен, через установку конфигурации track_vars или директивой <?php_track_vars?>, тогда переменные, активизированные посредством методов POST или GET, будут также находиться в глобальных ассоциативных массивах $HTTP_POST_VARS и $HTTP_GET_VARS соответственно.

Имена переменных РИСУНКА АКТИВИЗАЦИИ

При активизации (запуске) формы можно использовать рисунок (изображение) вместо стандартной кнопки запуска:

```
<input type=image src="image.gif" name="sub">
```

Когда пользователь нажимает кнопку мыши где-либо над таким рисунком, сопровождающая форма передается на сервер с двумя дополнительными переменными, sub_x и sub_y. Они содержат координаты места нажатия кнопки мыши пользователем внутри данного рисунка. Можно отметить, что практически, реальные имена переменных, передаваемые браузером, содержат точку вместо символа подчеркивания, но

PHP конвертирует точку в элемент подчеркивания (underscore) автоматически.

Переменные окружения

PHP автоматически создает переменные окружения, как и обычные переменные.

echo $HOME; /* Показывает переменную HOME, если она установлена.*/

Хотя при поступлении информации механизмы GET, POST и Cookie также автоматически создают переменные PHP, иногда лучше явным образом прочитать переменную окружения, для того чтобы быть уверенным в получении ее правильной версии. Для этого может использоваться функция getenv(). Для установки значения переменной окружения пользуйтесь функцией putenv().

Изменение типа

PHP не требует явного определения типа при объявлении переменной, тип переменной определяется по контексту в котором она используется. То есть, если вы присваиваете строковое значение переменной var, var становится строкой. Если затем присвоить переменной var значение целого (числа), то она станет целым.

Примером автоматического преобразования типа в PHP может служить оператор сложения '+'. Если какой-либо из операндов является числом с дробной частью (тип double), то затем все операнды оцениваются как double и результат будет иметь тип double. Иначе, эти операнды будут интерпретированы как целые (integers) и результат будет так же иметь тип integer. Отметим, что при этом НЕ меняются типы операндов, меняется только их оценка.

```
$foo = "0";  // $foo является строкой (ASCII 48)
$foo++;      // $foo является строкой "1" (ASCII 49)
$foo += 1;   // $foo сейчас является целым (2)
$foo = $foo + 1.3;  // $foo сейчас имеет тип double (3.3)
```

```
$foo = 5 + "10 Little Piggies";    // $foo является целым (15)
$foo = 5 + "10 Small Pigs";   // $foo является целым (15)
```

Если вы желаете изменить тип переменной, используйте settype().

Контрольные вопросы

1. Какие серверные языки веб-программирования Вы знаете?
2. Какие СУБД Вы знаете?
3. Как на HTML создать объект «кнопка»?
4. Как на HTML создать объект «текстовое поле»?
5. Как на HTML создать объект «флажок»?
6. Как на HTML создать таблицу 3х3?
7. Что делает SQL-команда «SELECT»?
8. Что делает SQL-команда «INSERT»?
9. Что делает SQL-команда «DELETE»?
10. Что делает SQL-команда «UPDATE»?

Рекомендуемые источники

1. Колдаев В.Д. Основы алгоритмизации и программирования: Учебное пособие / В.Д. Колдаев; Под ред. Л.Г. Гагариной. - М.: ИД ФОРУМ: ИНФРА-М, 2015. - 416 с. [Электронный ресурс]. http://znanium.com/bookread.php?book=336649.
2. Гагарина Л.Г. Технология разработки программного обеспечения: Учеб. пос. / Л.Г.Гагарина, Е.В.Кокорева, Б.Д.Виснадул; Под ред. проф. Л.Г.Гагариной - М.: ИД ФОРУМ: НИЦ Инфра-М, 2017. - 400 с. [Электронный ресурс]. http://znanium.com/bookread.php?book=389963.
3. Голицына О. Л. Программирование на языках высокого уровня: Учебное пособие / О.Л. Голицына, И.И. Попов. - М.: Форум, 2016. - 496 с. [Электронный ресурс]. http://znanium.com/bookread.php?book=139428.