## **Технология программирования и основные этапы ее развития**

Технологией программирования называют совокупность методов и средств, используемых в процессе разработки программного обеспечения. Как любая другая технология, технология программирования представляет собой набор технологических инструкций, включающих:

• указание последовательности выполнения технологических операций;

• перечисление условий, при которых выполняется та или иная операция;

•описания самих операций, где для каждой операции определены исходные данные, результаты, а также инструкции, нормативы, стандарты, критерии и методы оценки и т. п.

Кроме набора операций и их последовательности, технология также определяет способ описания проектируемой системы, точнее модели, используемой на конкретном этапе разработки.

**Технология серверного скриптового языка PHP**

## Что такое PHP?

В 1995 году датский программист (ныне живущий в Канаде) Расмус Лердорф (Rasmus Lerdorf) создал набор скриптов на основе Perl/CGI для вывода и учёта посетителей его онлайн-резюме, обрабатывающий шаблоны HTML-документов. Лердорф назвал набор Personal Home Page (Личная Домашняя Страница). Вскоре функциональности и быстроты Perl — интерпретатора скриптов — перестало хватать, и в 1996 году Лердорф разработал с использованием языка C новый интерпретатор шаблонов PHP/FI (англ. Personal Home Page / Forms Interpreter — «Личная Домашняя Страница / Интерпретатор форм»). PHP/FI включал базовую функциональность современного PHP: оформление переменных в стиле Perl ($имя\_переменной для вывода значения), автоматическую обработку форм и встраиваемость в HTML-текст и многое другое. Новорождённый язык отличался от своего прародителя более простым и ограниченным синтаксисом.

В 1997 году после длительного бета-тестирования вышла вторая версия обработчика, написанного на языке C — PHP/FI 2.0. Её использовали около 1 % (приблизительно 50 тысяч) всех интернет-доменов мира.

PHP (рекурсивный акроним для "PHP: Hypertext Preprocessor") это широко распространённый Открытый ресурс - язык скриптинга (сценариев) общего назначения, который создан специально для Web и который можно внедрять в HTML.

Простой ответ, но что это означает? Вот пример:

<html>

<head>

<title>Пример</title>

</head>

<body>

<?php

echo "Привет, я PHP-скрипт!";

?>

</body>

</html>

Заметьте, как это отличается от скриптов, написанных на [языке Perl](http://bourabai.ru/dbt/web/6.htm) или [языке C](http://bourabai.ru/C-Builder/c/index.htm) - вместо написания программы с большим количеством команд для вывода HTML, вы пишете HTML-разметку с некоторым количеством встроенного кода (скриптами) для выполнения каких-либо действий (в данном случае - для вывода некоторого текста). Код PHP заключён в специальные начальный и конечный тэги, что позволяет вам входить в и выходить из "режима PHP".

PHP отличается от других подобных языков, типа клиентского [JavaScript](http://bourabai.ru/jscript/index.htm), тем, что код выполняется на сервере. Если вы имеете скрипт, аналогичный вышеприведённому на сервере, то клиент получит результат работы этого скрипта, не имея возможности определить, каков был исходный код. Вы также можете сконфигурировать ваш web-сервер таким образом, чтобы он обрабатывал все ваши HTML-файлы с помощью PHP, и реально пользователь не будет иметь способа определить, что у вас "в рукаве". Мало того, есть серверная ZIP-программа [Zend Guard](http://bourabai.ru/php/zend_guard.htm), которая может скрыть текст РНР-скрипта даже от хакеров, вскрывших сайт и хозяев хостинга, распаковывая зашифрованный скрипт только для передачи РНР-процессору во время выполнения скрипта на сайте.

PHP позволяет расположить клиента базы данных на сервере, тем самым защищая ее от посягательств, то есть реализует многоуровневую [клиент-серверную технологию](http://bourabai.ru/dbt/client2.htm) с так называемыми ["толстым" и "тонким" клиентами](http://bourabai.ru/dbt/richclient.htm).

Наилучшим качеством PHP является то, что он предельно прост для новичка в программировании, но предлагает много продвинутых возможностей для программиста-профессионала. Не пугайтесь, увидев длинный список возможностей PHP. Вы скоро освоитесь и через несколько часов начнёте писать код.

Хотя разработка на PHP сконцентрирована вокруг серверного скриптинга, вы можете сделать с помощью РНР ещё много чего.

**Новые технологии в PHP5**

Не так давно вышла новая пятая версия PHP - PHP5. Рассмотрим основные изменения.

PHP5 - объектно-ориентированный скриптовый язык, имеющий в отличие от старых версий полный набор инструментария для ООП и технологию компиляционного ускорения трансляции Zend 2.0

**Новая объектная модель Zend 2.0**

Zend 1.0, а также более ранние версии "движка" PHP, нерационально использовали память, поскольку объекты обрабатывались как примитивные типы (переменные, списки), то есть посредством копирования. В новой объектной модели, которая используется в Zend 2.0 используется технология объектных дескрипторов, напоминающая аналогичную технологию в Java. Теперь объекты доступны только через дескрипторы, позволяя экономить память и получать полный контроль над созданием и дублированием объектов.

**Улучшенная производительность**

Zend Engine 2 использует новый менеджер памяти. Два главных его преимущества: улучшенная поддержка многопоточных сред (распределенные блоки не нуждаются во взаимных исключительных блокировках) и то, что после каждого запроса намного эффективнее освобождаются распределенные блоки памяти.

Изменение коснулись производительности встроенных функций PHP. По различным оценкам, увеличение производительности достигает 40%.

**Новые директивы**

В конфигурационном файле PHP - php.ini теперь доступны новые директивы:

* session.hash\_function - выбор функции хеширования сессий (SHA1 или MD5);
* session.hash\_bits\_per\_character - определяет, какое число бит в символе используется при хешировании сессий (от 4 до 6);
* register\_long\_arrays - включает/отключает регистрацию длинных массивов;
* mail.force\_extra\_parameters - дополнительные почтовые параметры;
* zend.ze1\_compatibility\_mode - включение совместимости с Zend Engine 1 (PHP4).

**Новые механизмы работы с потоками**

* Поддержка потоков претерпела множество изменений:
* Более высокая производительность функций readfile(), fpassthru() и некоторых других стандартных функций для работы с потоками;
* Новая функция stream\_socket\_client() - более мощный аналог сетевой функции fsockopen();
* Новая функция stream\_socket\_accept() - принимает соединения клиента;
* Новая функция stream\_socket\_get\_name() - получает локальное или удаленное имя сокета;
* Новая функция stream\_copy\_to\_stream() - позволяет копировать поток в поток;
* Новая функция, позволяющая читать информацию из потока до определенного символа или строки - stream\_get\_line();
* Поддержка динамической загрузки OpenSSL;
* Возможность работы функций opendir(), stat(), unlink() с FTP - соединениями.

**Реферат на тему**

**Технология серверного скриптового языка PHP**

Выполнил Рыбкин Н.А.

20.05.15