

## Лекция 1. КОМПОНЕНТЫ PHP-ПРИЛОЖЕНИЯ

Для создания и развития динамических веб-страниц вам потребуется понимать и использовать различные технологии. Разработка динамических веб-страниц включает три основных компонента: **веб-сервер**, **язык программирования сценариев**, исполняемых на стороне сервера, и **базу данных**. Разрабатывая веб-приложения с применением PHP, надо понимать все три компонента. Мы начнем с элементарных сведений об истории и назначении Apache (веб-сервер), PHP (язык программирования сценариев, исполняемых на стороне сервера) и MySQL (база данных). Это поможет вам осознать, как они вписываются в общую картину разработки веб-приложений.

### PHP

Язык PHP родился из потребности разрабатывать и поддерживать веб-сайты, обладающие динамической клиент-серверной функциональностью. В 1994 году *Расмус Лерддорф (Rasmus Lerdorf)* разработал набор сценариев с открытым исходным кодом на языке Perl, которые впоследствии были переписаны на языке C и превратились в то, чем является современный язык PHP.

Язык PHP широко распространен и совместим со всеми основными операционными системами. Он прост в изучении, что делает его идеальным инструментом для начинающих веб-программистов.

### Apache

Apache – это веб-сервер, который превращает запросы браузера в конечные веб-страницы и знает, как обрабатывать программный код PHP. PHP – это всего лишь язык программирования, и без поддержки веб-сервера, например Apache, у пользователей Сети нет никакой возможности получить страницы, содержащие программный код PHP.

Такие веб-серверы, как Apache, разрабатывались для поддержки HTML-файлов, поэтому для обработки PHP-кода им нужно знать, как это делается. Веб-сервер Apache использует систему управления модулями, которая позволяет наращивать функциональность путем подгрузки расширений.

## SQL и реляционные базы данных

**Язык структурированных запросов (Structured Query Language, SQL)** – самый распространенный язык, предназначенный для записи, извлечения, обновления и удаления информации в системах управления реляционными базами данных. Реляционная означает, что база данных соответствует реляционной модели, и относится к схеме и принципам хранения данных. Схема описывает структуру хранимых данных.

Реляционная база данных представляет собой набор таблиц, однако достаточно часто в состав базы данных входят и другие элементы, позволяющие дополнительно влиять на организацию и структуру данных в соответствии с определенным набором требований.

MySQL – это полнофункциональная свободно распространяемая система управления реляционными базами данных. MySQL начали разрабатывать в 1990-х годах, поскольку потребность в разумном управлении компьютерной информацией постоянно росла. Разработчики ядра MySQL пытались решить эту проблему с помощью маленькой и простой базы данных *mSQL*. Когда выяснилось, что *mSQL* не справляется со всеми задачами, которые создателям хотелось на нее возложить, они создали более мощную базу данных, которая превратилась в MySQL.

MySQL поддерживает несколько различных механизмов базы данных (*database engines*). Механизмы базы данных определяют, как MySQL в данный момент обрабатывает хранение и извлечение данных. Как следствие, каждый механизм хранения обладает собственным набором возможностей и преимуществ. Со временем имеющиеся механизмы базы данных становятся все более мощными и быстрыми.

В таблице ниже указано, когда в MySQL добавлялись разные функциональные возможности:

### Основные версии MySQL

Версия	Возможности
3.23	В качестве механизма по умолчанию добавлен механизм базы данных <i>MyISAM</i> . Эффективно обрабатывает большие объемы данных. Дебютировал механизм базы данных <i>InnoDB</i> для безопасного выполнения транзакций и поддержки внешних ключей

Версия	Возможности
	(foreign keys). Внешние ключи позволяют организовать однозначные взаимоотношения между таблицами базы данных
4.0	Появилась поддержка <i>объединений (union)</i> в запросах. Объединения позволяют совместить результаты двух запросов. Стало возможным изменять конфигурацию (настройки) без перезапуска базы данных
4.1	Появилась новая команда help для клиентов базы данных. Добавлена поддержка неименованных обзоров (unnamed views), также называемых <i>подзапросами (subquery)</i> . Неименованные обзоры позволяют рассматривать запрос как отдельную таблицу внутри другого запроса. Появилась поддержка наборов символов Юникода
5.0	<p>Добавлены <i>триггеры (triggers)</i>, <i>хранимые процедуры (stored procedures)</i>, <i>механизмы обеспечения ссылочной целостности (constraints)</i> и <i>курсоры (cursors)</i>.</p> <p>Триггер позволяет запускать выполнение некоторого кода в базе данных при определенном событии, например при вставке данных в таблицу. Хранимые процедуры – это способ определения и запуска программ внутри базы данных. Механизмы обеспечения ссылочной целостности дают возможность определять правила добавления и изменения записей в базе данных. Курсоры позволяют запускать выполнение некоторого кода в базе данных для каждой строки, соответствующей запросу</p>
5.1	<p>Добавлены механизмы <i>секционирования (partitioning)</i> и <i>планирования (scheduling)</i>, <i>интерфейс модулей расширения (Plug-in API)</i> и <i>механизм построчной репликации (row-based replication)</i>.</p> <p>Механизм секционирования позволяет разбивать физические хранилища крупных таблиц на основе определенных правил. Обычно применяется для ускорения производительности при работе с такими крупными таблицами, как архивы. Механизм планирования дает возможность запускать программный код, находящийся в базе данных, в определенные моменты времени. Интерфейс модулей расширения предоставляет возможность расширять или сужать функциональность сервера MySQL без его перезапуска. Механизм построчной репликации позволяет копировать данные между серверами с применением низкоуровневых операций.</p>

Действующей версией MySQL считается последняя доступная версия 5.x. MySQL 5.x по производительности сопоставима с любой из

гораздо более дорогих баз данных уровня предприятия, например Oracle, Informix, DB2 (IBM) или SQL Server (Microsoft). Такое повышение производительности стало возможным благодаря усилиям многих талантливых разработчиков открытого исходного кода, а также тестированию в сообществе. Механизм базы данных по умолчанию MyISAM прекрасно справляется с основными задачами баз данных, связанных с веб-приложениями.

## **Совместимость**

Такие веб-браузеры, как Safari, Firefox, Opera, Google Chrome и Internet Explorer, предназначены для обработки документов в формате HTML, и им все равно, под управлением какой операционной системы работает веб-сервер.

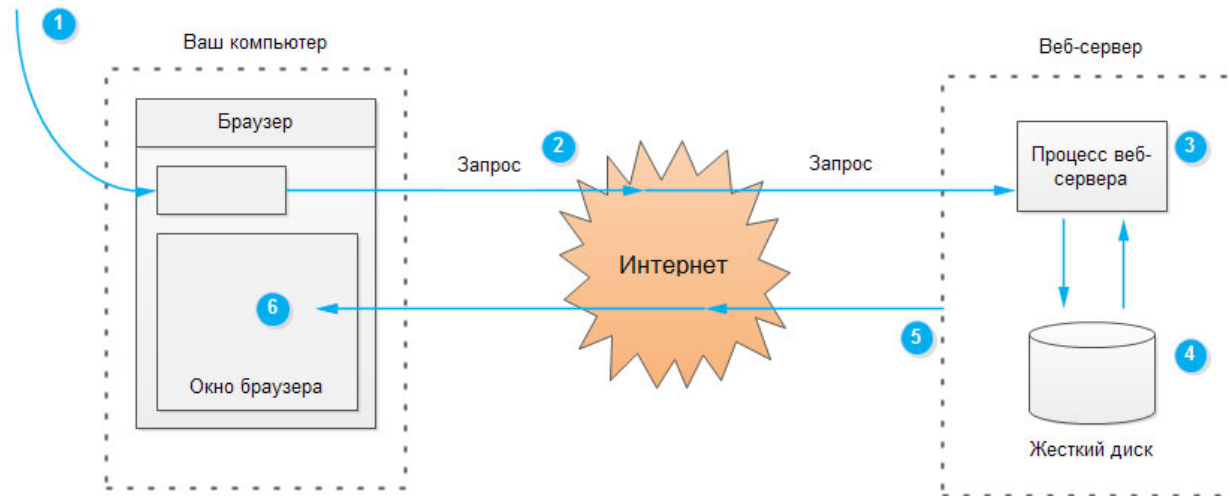
Apache, PHP и MySQL рассчитаны на поддержку многих операционных систем, поэтому вы не ограничены выбором какой-то определенной операционной системы для сервера или клиента. От беспокойства по поводу совместимости программного обеспечения вы избавлены.

## **Запрос данных веб-страницы**

Бывает непросто понять, как все эти составные части работают вместе. Если веб-сервер обнаруживает PHP-код, он пытается определить, не является ли данный файл файлом PHP-сценария. Если это так, файл передается для обработки интерпретатору PHP без какого-либо участия со стороны браузера. Но если в HTML-файле есть ссылка на внешний CSS-файл, то прежде чем отобразить страницу, браузер отдельно пошлет запрос, чтобы получить эту таблицу стилей.

Обработка PHP-кода сервером называется обработкой на стороне сервера. Запрашивая веб-страницу, вы запускаете целую цепь событий. Это взаимодействие между вашим локальным компьютером и веб-сервером (находящимся на веб-хосте в Интернете) иллюстрируется на рисунке: Пользователь только вводит URL и нажимает клавишу Enter, а обработка запроса выполняется "за кадром" в несколько шагов

Описание шагов, приведенных на рисунке:



1. Вы вводите в адресной строке браузера адрес веб-страницы.

2. Браузер разбивает адрес на составляющие и отправляет имя страницы веб-серверу. Например, после ввода адреса <http://www.tutorial.promo.net.ua/index.html> серверу [www.promo.net.ua](http://www.promo.net.ua) будет отправлен запрос на получение страницы `index.html`.

3. Программа на веб-сервере, называемая *процессом веб-сервера*, принимает запрос на получение страницы `index.html` и находит требуемый файл.

4. Веб-сервер читает файл `index.html` с жесткого диска.

5. Веб-сервер возвращает браузеру содержимое файла `index.html`.

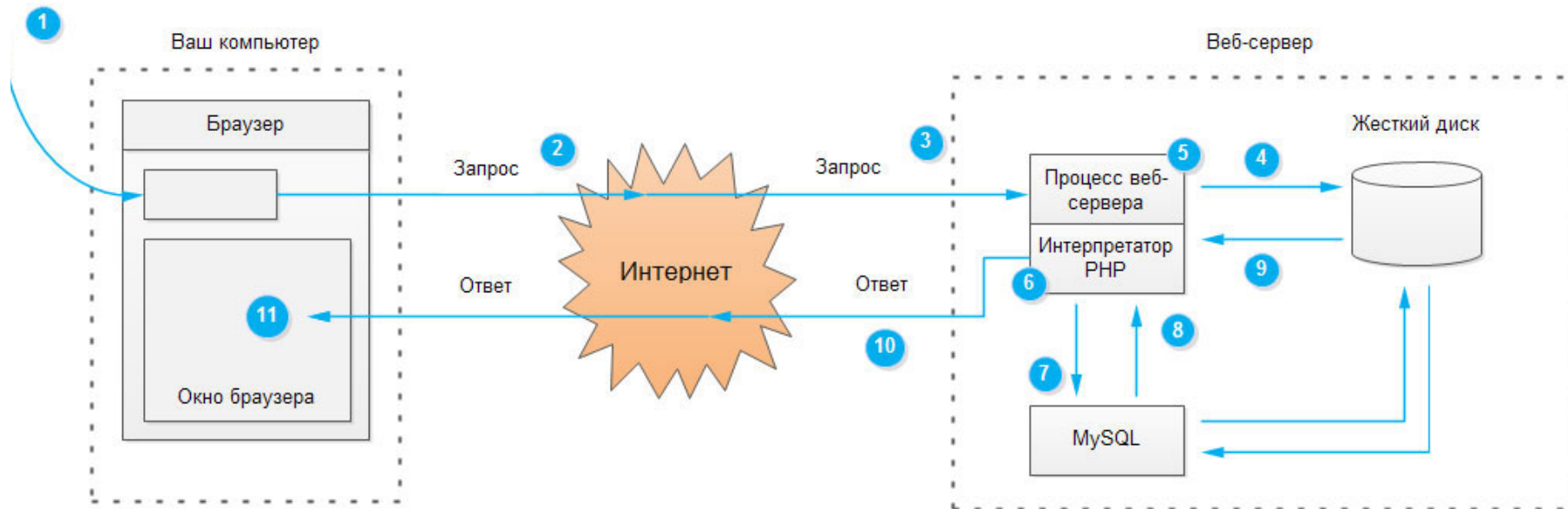
6. Браузер формирует внешний вид страницы на экране вашего компьютера в соответствии с разметкой HTML, полученной от веб-сервера.

HTML-файл с именем `index.html` (который запрашивается на рисунке) называется статической веб-страницей, потому что любой, кто

запросит страницу index.html, получит точно такую же страницу.

Если веб-сервер должен настроить возвращаемую страницу, к набору добавляются PHP и MySQL. На рисунке ниже показаны добавочные шаги в последовательности событий на веб-сервере:

Совместные действия интерпретатора PHP, MySQL и веб-сервера по созданию страницы



Описание всех шагов последовательности:

1. Вы вводите адрес веб-страницы в адресной строке браузера.
2. Браузер разбивает адрес на составляющие и отправляет имя страницы веб-серверу.
3. Процесс веб-сервера на хосте принимает запрос на получение страницы index.php.
4. Веб-сервер считывает файл index.php с жесткого диска хоста.

5.Веб-сервер определяет, что это сценарий PHP, а не простой HTML-файл, и поэтому передает его на обработку другому процессу –интерпретатору PHP.

6.Интерпретатор PHP исполняет PHP-код, который он обнаружил в тексте, полученном от процесса веб-сервера. Этот код включает в себя обращения к базе данных MySQL.

7.Интерпретатор PHP запрашивает у процесса базы данных MySQL обработку обращений к базе данных.

8.Процесс базы данных MySQL возвращает результаты запроса к базе данных.

9.Интерпретатор PHP завершает исполнение PHP-кода, добавляя данные, полученные из базы данных, и возвращает результат процессу веб-сервера.

10.Веб-сервер возвращает результат браузеру в виде HTML-текста.

11.Веб-браузер формирует внешний вид веб-страницы на экране вашего компьютера в соответствии с полученным HTML-текстом.

Может показаться, что шагов много, тем не менее, все это автоматически выполняется всякий раз, когда запрашивается веб-страница, содержащая PHP-код. Фактически, описанный процесс может неоднократно повторяться для одной веб-страницы, если она содержит много изображений и определений CSS, поскольку каждое из них браузер получает посредством отдельного запроса к веб-серверу.

При разработке динамических веб-страниц вам придется работать с множеством переменных и компонентов сервера, играющих важную роль в создании привлекательного, удобного для навигации и сопровождения веб-сайта. В следующей статье мы покажем, как установить все три основных компонента, необходимых для этой работы: Apache, PHP и MySQL.

## **Преимущества использования PHP и СУБД MySQL**

Количество различных базовых систем, на которых могут быть сформированы веб-приложения, довольно велико. В настоящем разделе приведено сравнение PHP с некоторыми другими базовыми системами и отмечены сильные стороны системы PHP и СУБД MySQL.

## Стоимость

PHP — один из языков, обозначенных буквой "P" в аббревиатуре LAMP, которая представляет широко применяемый стек программ. В стек LAMP входят такие распространенные программные продукты, как Linux, Apache, MySQL и PHP/Perl/Python, которые эксплуатируются на многих веб-сайтах и поддерживают много веб-приложений.

Значительная часть компонентов стека LAMP предоставляется бесплатно, и система PHP не составляет исключения. Под бесплатным предоставлением системы PHP подразумевается, что не нужно платить за предоставляемую возможность проводить разработку на языке PHP и эксплуатировать программы, разработанные на этом языке.

Разумеется, лицензия на СУБД MySQL предоставляется на других условиях, которые не всегда допускают отсутствие платы, но в любом случае остается возможность получить выпуск *Community Server* бесплатно. Разработчики СУБД MySQL предлагают несколько уровней контрактов поддержки на свой сервер базы данных. Более подробные сведения можно получить по адресу [www.mysql.com](http://www.mysql.com).

И система PHP, и СУБД MySQL могут эксплуатироваться на многих разновидностях базовых систем, включая различные варианты Linux, Microsoft Windows и др. Использование этого программного обеспечения в такой операционной системе, как Linux, позволяет получить в свое распоряжение полностью бесплатную базовую систему эксплуатации веб-приложений, для приобретения которой не требуется никаких предварительных расходов.

Безусловно, рассуждая о разработке программного обеспечения и прикладных платформах, следует учитывать, что предварительные расходы на лицензирование программного обеспечения в любом случае составляют лишь часть совокупной стоимости владения. Но долгие годы практического использования Linux, Apache, MySQL и PHP в производственной среде показали, что общая стоимость поддержки этих базовых систем является более низкой и во много раз ниже по сравнению с поддержкой инфраструктуры, основанной на коммерческом, платном программном обеспечении.

## Удобство в использовании

Сравнение с многими другими языками программирования показывает, что язык PHP зачастую позволяет быстрее разрабатывать мощные веб-приложения (в этом состоит одновременно его благословение и проклятье). В языке PHP уже заранее определены многие из



наиболее полезных специализированных функций (например, предназначенных для создания соединения с базой данных Oracle или выборки электронной почты с сервера IMAP). А изучение этого языка значительно упрощается благодаря наличию огромного количества готовых сценариев, с которыми можно ознакомиться, чтобы узнать, как проще всего составить ту или иную программу.

Язык PHP является легким для изучения, а программы на этом языке могут быть легко созданы, особенно теми разработчиками, которые имеют хотя бы небольшой опыт работы на языке программирования с синтаксисом, подобным языку C. В связи с этим язык PHP осваивать ненамного сложнее, чем язык HTML. Начинающие программисты получают возможность приступать к продуктивной работе, не пройдя достаточно полного обучения и не накопив опыта, поэтому иногда допускают ошибки, которые в дальнейшем вырастают в крупные проблемы защиты. В этом и состоит проклятие PHP.

А что касается СУБД MySQL, то для программистов, не имеющих опыта работы с реляционными базами данных или знакомых лишь с такой средой, как Microsoft Access, интерфейс командной строки этой СУБД, а также отсутствие готовых структурных решений могут на первый взгляд показаться довольно обескураживающими. В СУБД MySQL предусмотрено несколько инструментов с графическим интерфейсом пользователя, позволяющих упростить работу с базами данных. Однако какие бы то ни было графические инструментальные средства не заменят изучения теории и не подскажут, в чем заключаются методы качественного проектирования.

## Использование кода PHP, встроенного в код HTML

Код PHP может быть встроен в код HTML. Иными словами, сценарии формирования страниц, написанные на языке PHP, представляют собой обычные сценарии вывода страниц в коде HTML, в которых по мере необходимости осуществляется переход в режим обработки кода PHP. Ниже приведен пример подобного сценария:

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<title>Простой документ</title>
</head>
<body>
```

<p>Привет,

<?php

*// В этом участке кода произошел переход в режим PHP. Вместо статических*

*// переменных в следующих трех строках кода вполне можно было применить*

*// вызовы запросов к базе данных или функции для работы с cookie-файлами;*

*// еще один вариант состоит в том, что значения переменных могут быть переданы из формы.*

`$firstname = 'Иван';`

`$lastname = 'Петров';`

`$shortname = 'И.';`

`echo "$shortname $lastname";`

*// Все прошло успешно, и теперь происходит возврат в режим HTML*

`?>`

. Мы знаем кто вы! Вас зовут `<?php echo $firstname; ?>`.

<p>Вы посетили эту страницу `<?php echo date('Y-m-d H:i:s'); ?>`

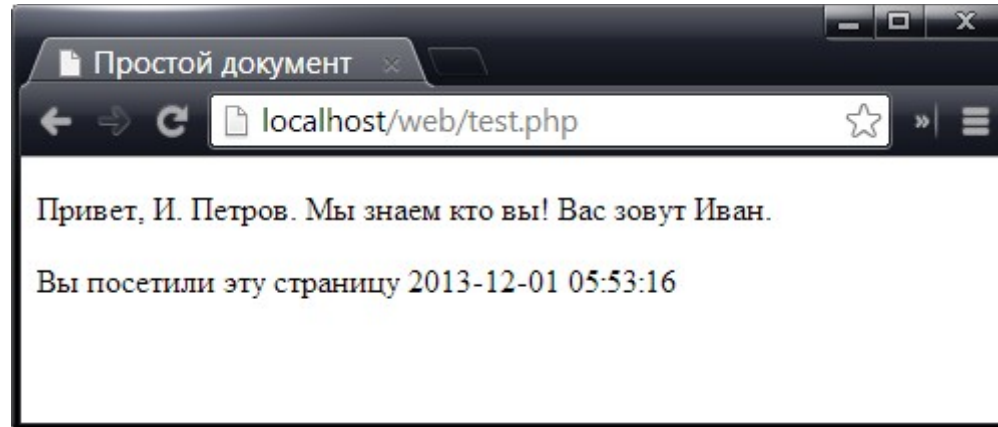
</body>

</html>

После того как клиент запрашивает эту страницу, веб-сервер выполняет ее предварительную обработку. Это означает, что сервер просматривает страницу от начала и до конца, отыскивает фрагменты кода PHP и предпринимает попытки обработать эти фрагменты.

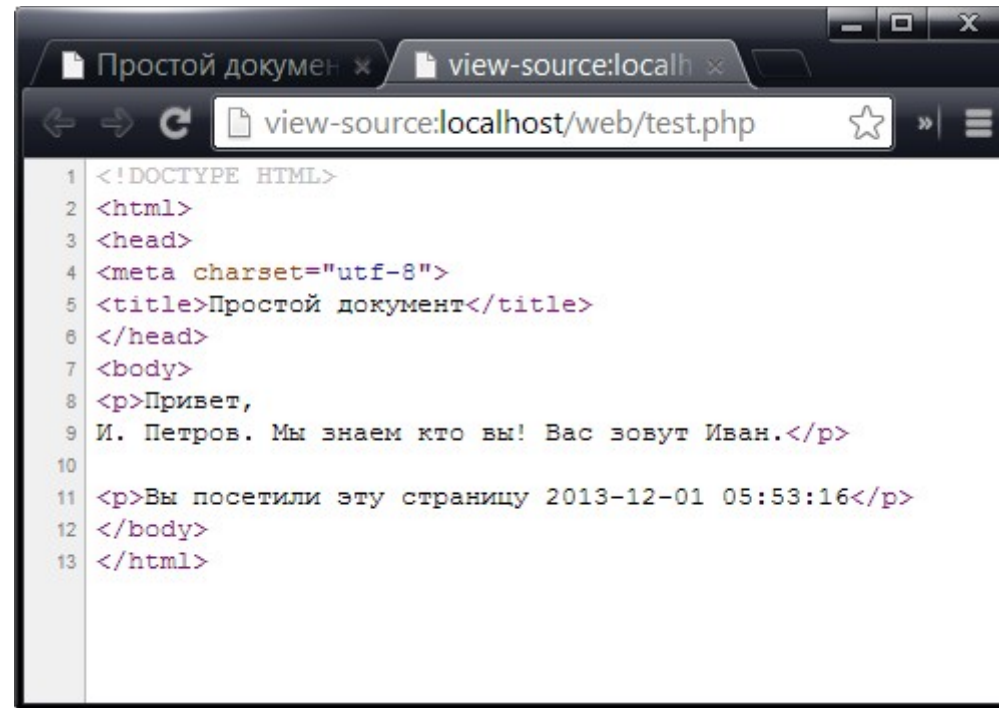
Одной из важных составляющих обработки является то, что синтаксический анализатор запоминает все переменные (обозначенные префиксами в виде знака доллара) и присвоенные им значения, чтобы в конечном итоге подставить вместо параметров вызова функций PHP (в данном случае функции echo) значения этих переменных. Если все операции обработки завершаются успешно, то препроцессор языка PHP в конечном итоге передает в клиентский браузер обычную HTML-страницу, которая отображается, как показано на рисунке:

Результат вывода кода HTML, полученного после предварительной обработки кода PHP



Исходный код страницы, приведенной на рисунке, в клиентском браузере будет выглядеть, как показано ниже (для такого просмотра необходимо выбрать команду "Просмотр кода страницы" в контекстном меню браузера):

Исходный код страницы в браузере пользователя



```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Простой документ</title>
6 </head>
7 <body>
8 <p>Привет,
9 И. Петров. Мы знаем кто вы! Вас зовут Иван.</p>
10
11 <p>Вы посетили эту страницу 2013-12-01 05:53:16</p>
12 </body>
13 </html>
```

Этот код полностью совпадает с тем, который бы отобразился в браузере, если бы эта страница была написан в коде HTML вручную. Поэтому все — очень просто!

Возможность внедрения кода PHP в код HTML влечет за собой целый ряд описанных ниже важных последствий:

- Код PHP может быть быстро добавлен к коду, сформированному с помощью программы редактирования с интерфейсом непосредственного отображения (WYSIWYG).
- Благодаря использованию языка PHP может быть обеспечено разделение труда между дизайнерами страниц и составителями сценариев.
- Код HTML можно применять непосредственно, а не перезаписывать его для вывода с помощью какого-либо языка

программирования.

- Язык PHP позволяет сократить трудозатраты и повысить эффективность труда, поскольку способствует ускоренному обучению и является простым в использовании.