

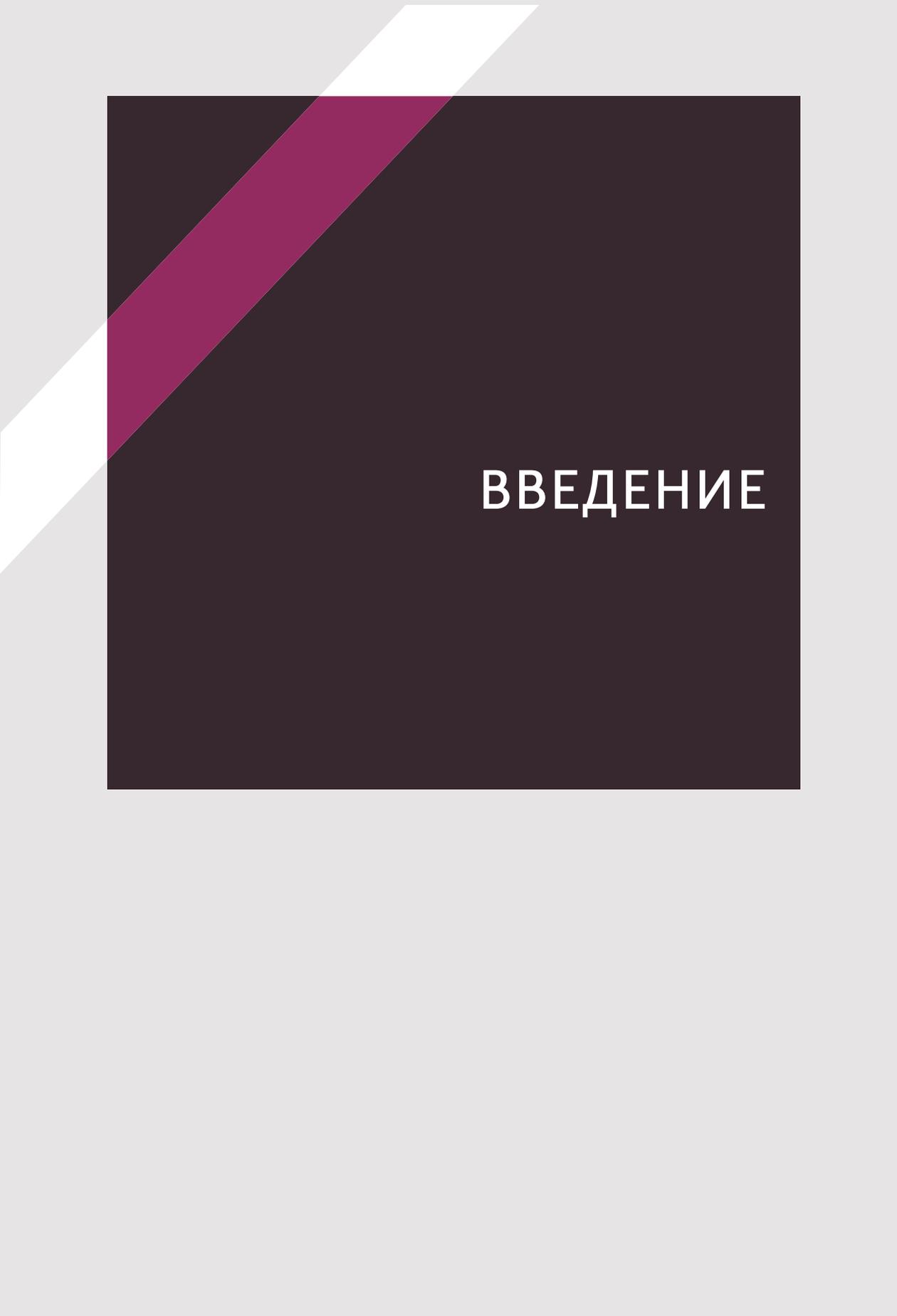


# ОГЛАВЛЕНИЕ

<b>Введение</b> .....	7
<b>Часть А Основы программирования</b> .....	23
Глава 1 Переменные, выражения и операторы .....	35
Глава 2 Управляющие конструкции.....	73
Глава 3 Функции.....	109
Глава 4 Классы и объекты.....	149
<b>Часть Б Динамические веб-страницы</b> .....	183
Глава 5 Встроенные функции .....	207
Глава 6 Получение данных из браузера.....	237
Глава 7 Изображения и файлы .....	291
Глава 8 Дата и время.....	315
Глава 9 Cookie и сессии .....	335
Глава 10 Обработка ошибок.....	355
<b>Часть В Сайты на основе баз данных</b> .....	387
Глава 11 Язык структурированных запросов (SQL) .....	403
Глава 12 Получение и вывод информации из базы данных .....	439
Глава 13 Изменение информации в базе данных .....	489
<b>Часть Г Расширение функционала и модернизация учебного приложения</b> .....	527
Глава 14 Рефакторинг и внедрение зависимостей .....	539
Глава 15 Пространства имен и библиотеки.....	563
Глава 16 Регистрация пользователей .....	609
Глава 17 Добавление нового функционала .....	639
<b>Предметный указатель</b> .....	669
<b>Примечания</b> .....	677

Загрузить код для этой книги можно на странице  
<http://addons.eksmo.ru/it/phpbook.zip>





# ВВЕДЕНИЕ

Изучив эту книгу, вы научитесь создавать веб-сайты на языке программирования PHP и хранить информацию в базе данных MySQL.

PHP — это язык программирования, который был создан для работы на веб-сервере, чтобы генерировать HTML-страницы по запросу. Это означает, что страницы сайта не являются статичными, а могут отображать разную информацию для разных пользователей. Это обязательное требование для любого сайта, который позволяет пользователям выполнять такие задачи, как:

- **регистрация и вход в систему.** Имя, адрес электронной почты и пароль каждого пользователя уникальны;
- **совершение покупки.** Детали заказа, оплаты и доставки каждого клиента уникальны;
- **поиск на сайте.** Результаты поиска зависят от поисковой фразы.

Язык программирования PHP создан для работы с базами данных, например MySQL, которые могут хранить такие данные, как содержимое страниц, сведения о продаваемых товарах или посетителях сайта. Используя PHP, вы научитесь создавать веб-страницы, позволяющие пользователям обновлять информацию, хранящуюся в базе данных. Например:

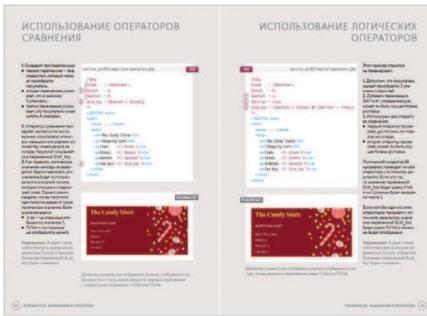
- **системы управления контентом** позволяют владельцам (администраторам) сайта обновлять с помощью формы содержимое сайта. Эти обновления отображаются для пользователей без написания нового кода;
- **интернет-магазины** позволяют владельцам выставлять товары для продажи, а покупателям — делать покупки;
- **социальные сети** позволяют пользователям регистрироваться и входить в систему, создавать свои профили, загружать собственный контент и просматривать страницы в зависимости от интересов.

Поскольку отображаемая на страницах сайтов информация хранится в базе данных, такие сайты известны как **веб-сайты на основе баз данных**.

Ниже представлены описанные в книге типы страниц. Различный дизайн страниц передает разную информацию.



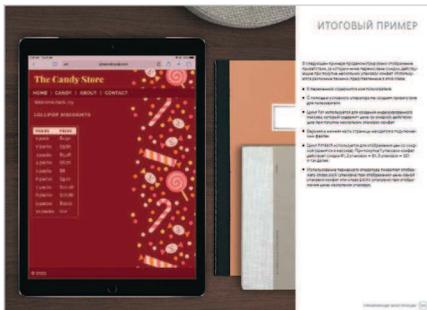
Информационные страницы отображаются на белом фоне. На информационных страницах обычно освещаются какие-либо темы, описывается контекст.



Страницы с кодом отображаются на бежевом фоне. Они представляют собой примеры кода по каждой изученной теме.



Диаграммы и инфографика отображаются на темном фоне.



Итоговые примеры в конце некоторых глав содержат код, обобщающий всю информацию, изученную в этой главе.



В конце каждой главы тезисно повторяется весь изученный в ней материал. На этих страницах кратко приведены ключевые понятия по теме.

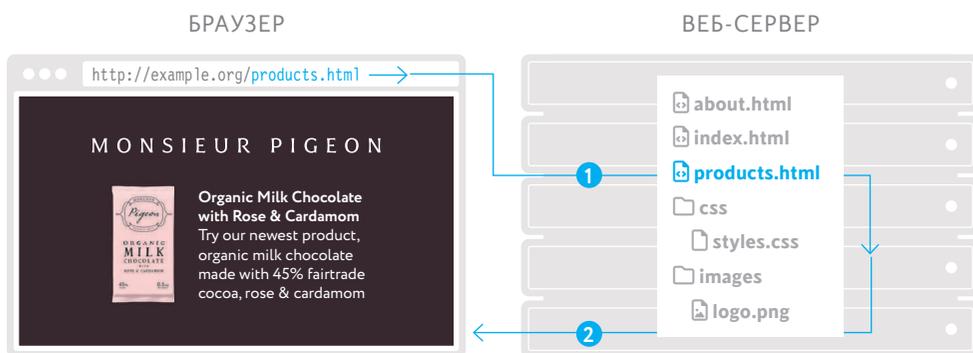
# СТАТИЧЕСКИЕ И ДИНАМИЧЕСКИЕ ВЕБ-САЙТЫ

Когда сайт разработан только с использованием HTML и CSS, каждый пользователь будет видеть одно и то же содержимое, так как всем отправляются одни и те же HTML- и CSS- файлы.

1. Когда браузер запрашивает страницу сайта, созданного с использованием только HTML и CSS, запрос отправляется на **веб-сервер**, на котором размещен этот сайт.
2. Веб-сервер находит запрошенный браузером HTML-файл и возвращает его браузеру. Затем браузер запрашивает все файлы, которые упомянуты в HTML, такие как CSS для стилизации страницы, мультимедиа (например, изображений), JavaScript и другие используемые страницей файлы.

Поскольку всем пользователям отправляются одни и те же HTML-файлы, все они будут видеть один и тот же контент. Такой тип сайта известен как **статический сайт**.

Владельцам статических веб-сайтов необходимо владеть навыками HTML и CSS, чтобы иметь возможность обновлять информацию на сайте. Если владелец (администратор) такого сайта захочет обновить текст на странице, то HTML-код придется обновить вручную и загрузить на веб-сервер.



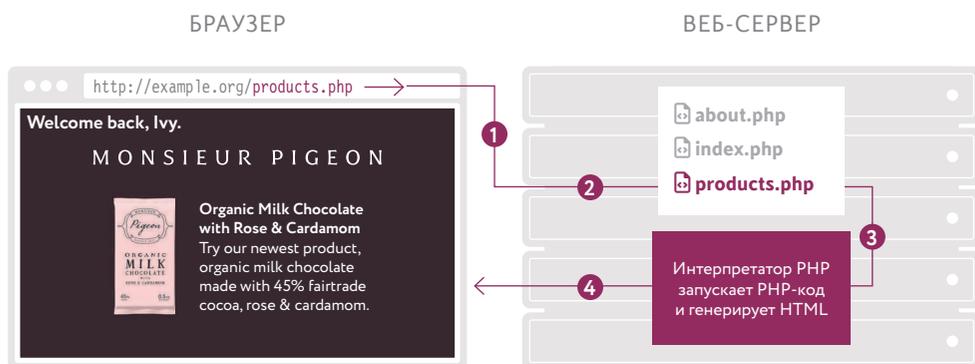
Материал этой книги предполагает базовое знакомство с HTML и CSS. Если вы только изучаете HTML и CSS, рекомендуем ознакомиться с нашей книгой: <https://book24.ru/product/html-i-css-razrabotka-i-dizayn-veb-saytov-5136049/>.

Когда веб-сайт разработан с использованием PHP, каждый пользователь может видеть разный контент, так как страница PHP каждый раз заново создает HTML-код, который отправляется пользователю.

Такие сайты, как eBay, Facebook<sup>®</sup> и новостные порталы, практически при каждом новом посещении отображают новую информацию. Если вы посмотрите на исходный код полученной страницы в браузере, то увидите HTML-код, однако разработчик не будет постоянно вручную обновлять его.

Такой тип сайта известен как **динамический веб-сайт**, поскольку HTML-код, который отправляется в браузер, каждый раз создается кодом PHP в ответ на запрос пользователя.

1. Когда браузер запрашивает страницу сайта, созданного на PHP, запрос отправляется на веб-сервер.
2. Веб-сервер находит файл PHP.
3. Весь имеющийся в файле PHP-код выполняется с помощью программы, называемой **интерпретатором PHP**, и новый HTML-код создается для конкретного пользователя, запросившего страницу.
4. Веб-сервер отправляет HTML-код, созданный для этого пользователя, в браузер. Копия файла не сохраняется. При следующем запросе PHP-файла HTML-код создается для этого же пользователя заново.



PHP-код в веб-браузер не отправляется. Он используется для создания HTML-кода, который затем направляется в браузер. Поскольку PHP-код выполняется на стороне веб-сервера, он относится к **серверному программированию**.

PHP можно использовать для создания HTML-страниц, адаптированных или персонализированных для каждого отдельного пользователя. На страницах может отображаться имя пользователя, интересные его темы или сообщения его друзей.

<sup>®</sup> Принадлежит компании Meta, которая признана экстремистской и запрещена на территории РФ.

# PHP: ЯЗЫК И ИНТЕРПРЕТАТОР

Интерпретатор PHP — это программа, которая запускается на веб-сервере. Она выполняет программный код, написанный на языке PHP.

Программное обеспечение позволяет использовать компьютер для выполнения определенных задач. При этом нет необходимости досконально знать, как именно компьютер выполняет эту задачу. Например:

- программы электронной почты позволяют отправлять и получать электронные письма. При этом пользователю не нужно понимать, как именно компьютеры хранят или передают электронные письма;
- программа Adobe Photoshop позволяет редактировать изображения, но от пользователя не требуется разбираться, каким образом компьютеры манипулируют изображениями.

С помощью программы мы каждый раз выполняем одни и те же задачи, но при этом манипулируем разными данными:

- программа электронной почты может использоваться для создания, отправки, получения и хранения электронных писем. Но содержание и получатели каждого электронного письма могут быть разными;
- программа Adobe Photoshop выполняет такие задачи, как добавление фильтра, изменение размера или обрезка изображения. Программа может выполнять задачи с любым изображением.

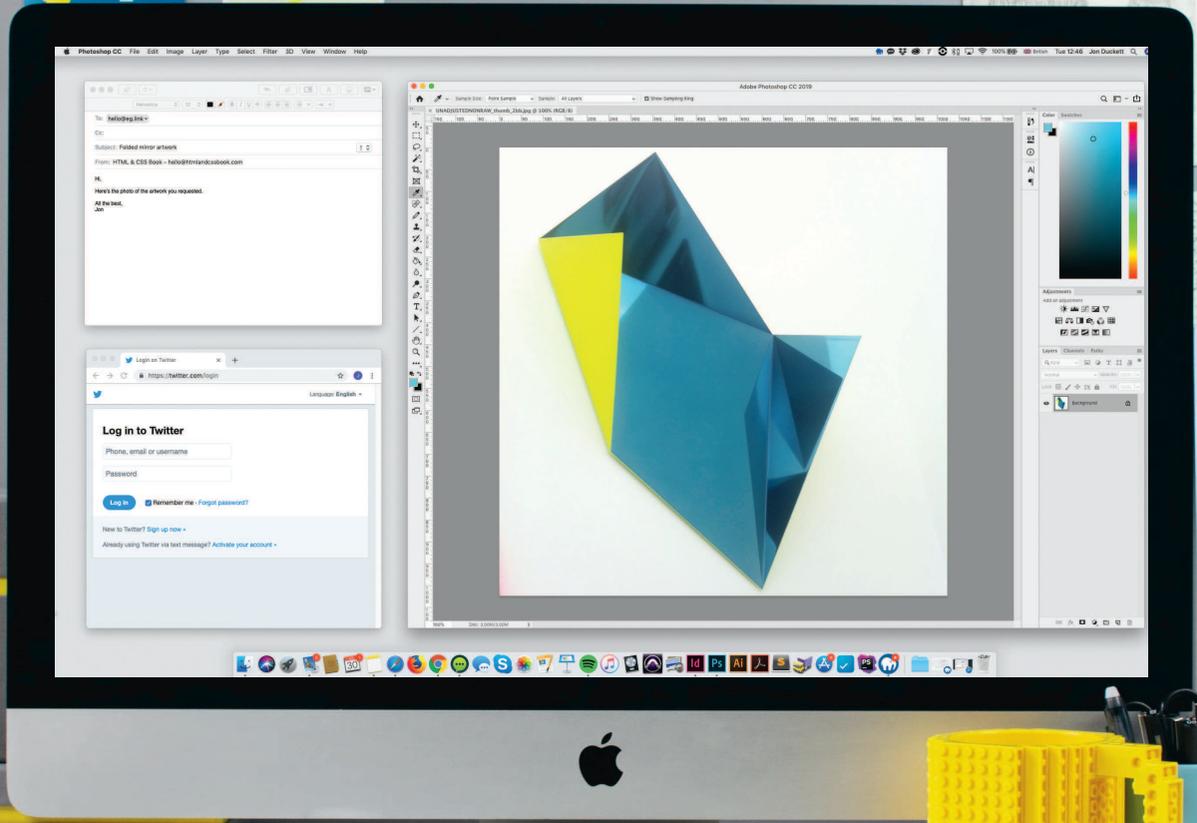
Обе эти программы имеют графический пользовательский интерфейс, с которым вы взаимодействуете для выполнения каких-либо задач.

Интерпретатор PHP — это тоже программа, которая работает на веб-сервере. Однако вы сообщаете интерпретатору PHP задачи, которые ему необходимо выполнить, не с использованием графического пользовательского интерфейса, а с помощью кода, написанного на языке PHP.

Когда вы создаете веб-страницу, используя PHP, эта страница всегда будет выполнять одни и те же задачи. Однако она может выполнять их, используя разные данные при каждом запросе страницы. Например, сайт, написанный на PHP, может содержать следующие страницы:

- одна и та же страница входа в систему, которую каждый пользователь использует для входа, несмотря на то что адрес электронной почты и пароль каждого пользователя уникальны;
- одна и та же страница учетной записи пользователя, на которой каждый пользователь видит детали своего личного кабинета. Даже если этой страницей одновременно пользуются сотни разных пользователей, они будут видеть только собственные данные.

Это становится возможным благодаря тому, что программа использует одни и те же инструкции при выполнении этих задач, но данные пользователей будут каждый раз разные.



Otatur Eliasson — Baroque

# ВЫПОЛНЕНИЕ ОДНОЙ И ТОЙ ЖЕ ЗАДАЧИ С РАЗНЫМИ ДАННЫМИ

Языки программирования позволяют создавать правила, сообщающие компьютеру, как выполнять ту или иную задачу. При этом данные, которые использует программа при каждом выполнении этой задачи, могут быть разными.

Работая с каким-либо языком программирования, вы указываете компьютеру точные инструкции для выполнения задачи. Эти инструкции значительно отличаются от тех, которые вы даете человеку, когда просите выполнить какое-либо задание.

Представьте, что вам необходимо купить пять плиток шоколада и вычислить их общую стоимость. Для этого необходимо умножить цену одной плитки на необходимое количество. Это правило можно записать следующим образом:

```
total = price * quantity
// итог = цена * количество
```

Теперь на основе этой формулы можно рассчитать конечную стоимость покупки.

- Если шоколад стоит \$1, а вы покупаете 5 плиток, общая сумма будет составлять \$5.
- Если цена одной плитки составляла \$1,50, правило остается тем же, но общая сумма будет равна \$7,50.
- Если вы хотите купить 10 плиток по \$2 каждая, правило остается тем же, но общая сумма будет равна \$20.

Значения, используемые вместо слов `total` (итого), `price` (цена) и `quantity` (количество), могут изменяться, но правило, используемое для расчета общей стоимости шоколадных плиток, остается прежним.

При использовании PHP для создания веб-страницы вам сначала необходимо определить следующее:

- какую задачу необходимо выполнить;
- какие данные могут изменяться при выполнении этой задачи.

Затем вы сообщаете интерпретатору PHP подробные инструкции, как выполнить задачу и как назвать те данные, которые будут изменяться.

Если вы определите следующие данные:

```
price = 3 // цена
quantity = 5 // количество
```

а затем воспользуетесь следующим правилом:

```
total = price * quantity
// итог = цена * количество
```

то значение `total` (итого) будет равно 15. В следующий раз при выполнении того же кода вы можете задать для цены или количества другие значения, и программа, используя то же правило, сможет вычислить новую сумму.

Слова, которые обозначают изменяющиеся значения, программисты называют **переменными**, так как эти значения могут изменяться каждый раз при запуске программы.

**total = price x quantity**

// ИТОГО = цена \* количество



**\$9** = **3 x 3**

# ЧТО ПРЕДСТАВЛЯЕТ СОБОЙ СТРАНИЦА НА PHP?

Страница PHP в простейшем случае<sup>1</sup> представляет собой смесь кода HTML и PHP. Она используется для генерации HTML-кода в ответ на запрос браузера.

Ниже представлен пример страницы на PHP, содержащей смесь языков HTML и PHP.

- HTML-код обозначен синим цветом;
- код PHP выделен фиолетовым цветом.

Что делает интерпретатор PHP, когда открывает файл:

- отправляет любой HTML-код как есть;
- исполняет все команды PHP-кода, который обычно генерирует контент для HTML-страницы.

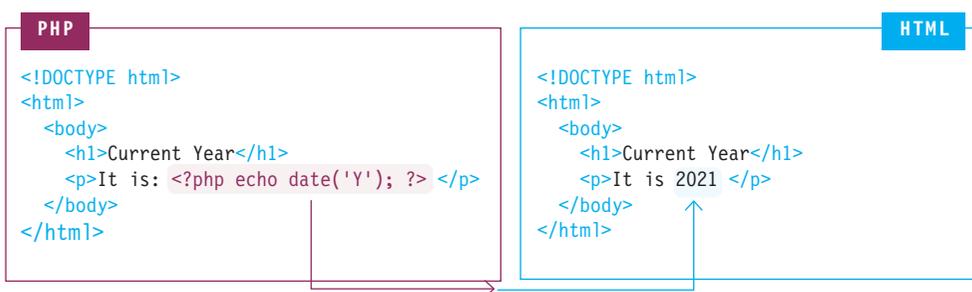
Представленный здесь PHP-код определяет, какой сейчас год, и выводит его между открывающим `<p>` и закрывающим `</p>` тегами.

PHP-код может выполнять как простейшие операции — арифметические вычисления или вычисление текущей даты, так и более сложные, например обновление сведений в базе данных на основе информации, отправленной через HTML-форму.

Когда интерпретатор PHP завершает обработку файла PHP, он отправляет сгенерированную на лету HTML-страницу в браузер.

Ниже представлен пример HTML-страницы, которая будет отправлена в браузер после выполнения интерпретатором PHP приведенного выше кода.

Интерпретатор PHP получает значение текущего года и отображает его на созданной им HTML-странице.



Интерпретатор PHP получает значение текущего года и выводит его внутри тегов абзаца.

Каждая страница обычно выполняет одну и ту же задачу, но способна отображать различную информацию при каждом запросе.

Веб-сайт, разработанный на PHP, состоит из набора страниц, каждая из которых выполняет определенную задачу. Например, сайт, который позволяет пользователям войти в систему, может содержать следующие страницы:

- страница входа в систему. С ее помощью пользователь заходит на сайт;
- страница профиля. Отображает личный кабинет пользователя.

Каждый раз, когда запрашивается одна из этих страниц, она должна уметь работать с данными каждого пользователя. Следовательно, ей потребуется:

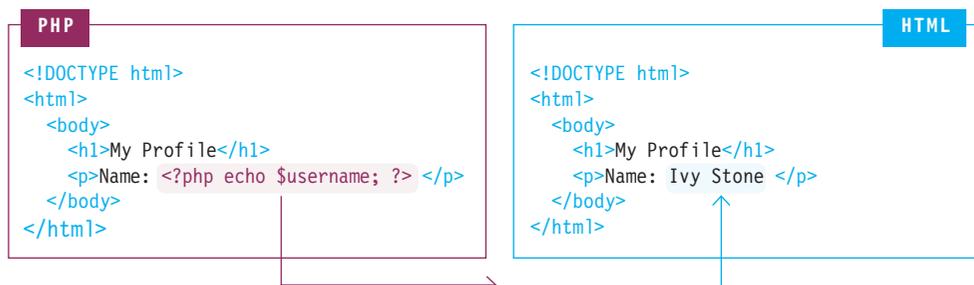
- содержать инструкции, как выполнить задачу, для которой она предназначена;
- определить именованные фрагменты данных, которые могут изменяться при каждом запросе.

В PHP каждая переменная имеет свое имя, которое описывает суть представляемого ей значения. При этом само значение может изменяться при каждом запросе страницы. Код PHP сообщает интерпретатору следующее:

- какое имя переменной использовать для фрагмента данных, который может изменяться при *каждом* запросе;
- какое значение использовать при обработке *данного* конкретного запроса.

После отправки HTML-страницы пользователю, интерпретатор PHP удалит все значения, хранящиеся в переменных, поэтому эта же задача может быть выполнена для следующего пользователя, запрашивающего страницу, но с другими значениями.

Для более продолжительного хранения данных их необходимо поместить в базу данных, например MySQL.



Интерпретатор PHP считывает значение переменной `$username` и выводит его внутри тегов абзаца.

# ЗНАКОМСТВО С MYSQL

MySQL – это система управления базами данных (СУБД). Базы данных (БД) хранят информацию в структурированном виде, поэтому вы можете легко прочитать или обновить данные, которые в ней хранятся.

Программа для работы с электронными таблицами, например Excel, хранит информацию в виде таблицы, состоящей из столбцов и строк. Затем она может использовать хранящиеся в электронной таблице данные для вычислений или манипулировать ими с помощью формул.

MySQL – это программа, которая хранит информацию примерно таким же образом: в виде **таблиц**, которые точно так же состоят из строк и колонок. При этом вы можете использовать PHP для получения информации из базы данных, а также для ее обновления.

Одна база данных может содержать несколько таблиц. Каждая таблица обычно содержит данные одного типа. Ниже представлены два примера таблиц БД, которые содержат:

- данные пользователей, зарегистрированных на сайте;
- отображаемые на сайте публикации.

В каждой таблице **имена столбцов** описывают тип информации, которую содержит каждый столбец таблицы:

- таблица `member` (пользователь) содержит столбцы, в которых для каждого пользователя хранится имя, фамилия, адрес электронной почты, пароль, дата регистрации и изображение профиля;
- таблица `article` (публикация) содержит столбцы, в которых для каждой публикации хранятся заголовок, краткое содержание, полный текст, дата создания и некоторые другие данные.

Каждая **строка** содержит данные, описывающие одну из тех сущностей, для хранения которых была создана таблица:

- в таблице `member` каждая строка представляет одного пользователя;
- в таблице `article` каждая строка содержит одну публикацию.

ИМЯ ТАБЛИЦЫ		ИМЯ СТОЛБЦА				СТОЛБЕЦ		
<b>member</b>								
id	forename	surname	email	password	joined	picture		
1	Ivy	Stone	ivy@eg.link	\$2y\$10\$MAdTTCA0MiOw	2021-01-01 20:28:47	ivy.jpg		
2	Luke	Wood	luke@eg.link	\$2y\$10\$NN5HEAD3atar	2021-01-02 09:17:21	NULL		
3	Emiko	Ito	emi@eg.link	\$2y\$10\$/RpRmiUMStji	2021-01-02 10:42:36	emi.jpg		
<b>article</b>								
id	title	summary	content	created	category_id	member_id	image_id	published
1	Systemic	Brochure	<p>This	2021-01-01	1	2	1	1
2	Polite	Poster	<p>These	2021-01-02	1	1	2	1
3	Swimming	Architect	<p>This	2021-01-02	4	1	3	1

СТРОКА

Используя PHP, вы научитесь:

- **извлекать данные из базы данных** и отображать эту информацию на веб-странице;
- **добавлять в нее новые строки.** Чтобы создать новую публикацию, необходимо добавить строку в таблицу `article` и внести в нее данные для каждого столбца;
- **удалять строки.** Чтобы удалить публикацию, необходимо удалить всю строку, представляющую публикацию;
- **изменять данные в существующей строке.** Чтобы обновить адрес электронной почты пользователя, необходимо найти нужную строку в таблице `member` (пользователь), а затем обновить значение в столбце `email` (электронная почта) этой строки.

Обратите внимание, что в обеих таблицах первый столбец — `id` (идентификатор). Каждая строка таблицы содержит в этом столбце уникальное значение (в данном примере значения в столбцах начинаются с 1 и увеличиваются на 1 для каждой строки). Значения в столбце `id` позволяют указать базе данных, с какой строкой данных вам необходимо работать. Например, вы можете получить пользователя с идентификатором 2 или публикацию с идентификатором 1.

member							
id	forename	surname	email	password	joined	picture	
1	Ivy	Stone	ivy@eg.link	\$2y\$10\$MAdTTCA0Mi0w	2021-01-01 20:28:47	ivy.jpg	
2	Luke	Wood	luke@eg.link	\$2y\$10\$NN5HEAD3atar	2021-01-02 09:17:21	NULL	
3	Emiko	Ito	emi@eg.link	\$2y\$10\$/RpRmiUMStji	2021-01-02 10:42:36	emi.jpg	

article								
id	title	summary	content	created	category_id	member_id	image_id	published
1	Systemic	Brochure	<p>This	2021-01-01	1	2	1	1
2	Polite	Poster	<p>These	2021-01-02	1	1	2	1
3	Swimming	Architect	<p>This	2021-01-02	4	1	3	1

MySQL — **реляционная база данных**, так как она позволяет устанавливать связи между данными, хранящимися в разных таблицах.

Например, публикации в таблицах ниже написаны разными пользователями сайта. В таблице `article` значение в столбце `member_id` указывает, какой пользователь написал публикацию. Столбец `member_id` содержит число, соответствующее одному из значений в столбце `id` таблицы пользователей.

Первую публикацию разместил пользователь, у которого в столбце `id` стоит 2 (Luke Wood). Вторая и третья публикации размещены пользователем, у которого в столбце `id` стоит 1 (Ivy Stone).

Эти взаимосвязи:

- структурируют данные так, чтобы каждая таблица содержала информацию только об одном конкретном типе данных (пользователь или публикация);
- помогают избежать дублирования одних и тех же данных в нескольких таблицах (экономия места в базе данных);
- упрощают обновление данных. Если пользователь меняет свое имя, его необходимо обновить только в таблице `member`, а не в каждой размещенной им публикации.