

Работа с cookie в PHP

Что такое cookie

Файлы cookie – это небольшие текстовые файлы, которые создаёт браузер в специальной папке по команде PHP программы. Хотя cookie может создать и JavaScript, но в этой статье не об этом. Тут мы будем говорить только о установке cookie через PHP программу.

Файлы cookie содержат в себе информацию в виде пар имя=значение. Например, в cookie можно записать логин и пароль пользователя и хранить их на компьютере клиента, чтобы он в следующий раз не залогинивался заново. Конечно, хранение логина и пароля в cookie не самый безопасный способ, и эти данные лучше хранить используя [сессии в PHP](#), но в качестве примера использования cookie этот случай подойдёт.

PHP программа даёт команду браузеру-клиенту установить cookie используя строку в заголовке ответа. То есть вы должны представлять, как работает интернет, что такое заголовок запроса и заголовок ответа, чтобы понять как работает установка cookie.

Итак, всю информацию о том, какие cookie должен установить браузер, PHP программа отправляет в заголовке ответа.

На самом деле всё очень просто, читаем дальше и смотрим примеры.

Установка cookie в PHP

Для установки cookie в языке PHP есть функция `setcookie()`, она и задаёт cookie, которое будет передано браузеру вместе с другими HTTP заголовками.

Все заголовки, которые создаёт ваш скрипт, должны быть отправлены **до** того, как ваш скрипт что-то выведет в окно браузера.

Вернёмся к нашей функции. Вот её синтаксис:

```
bool setcookie (  
    string имя,  
    string значение,  
    int время жизни,  
    string путь,  
    string домен,  
    bool протокол,  
    bool http only )
```

`bool`, `string`, `int` в этом синтаксисе - это типы данных. То есть «`bool setcookie()`» обозначает то, что функцией `setcookie()` возвращает значение булевого типа. Далее, первый и второй аргумент (имя и значение) должны быть типа строка (`string`), и так далее.

Сразу скажу, для начала вам понадобится только первые три параметра. На остальные можете пока не обращать внимания.

1. `string` имя – имя в паре имя=значение
2. `string` значение – значение в паре имя=значение
3. `int` время жизни – время хранения `cookie`, это метка времени Unix, т.е. желательно задавать это время с помощью функции `time()`, прибавляя время в секундах, через которое срок действия `cookie` должен истечь. Также можно воспользоваться функцией `mktime()`.
4. `string` путь – путь к директории на сервере, из которой будут доступны `cookie`.
5. `string` домен – домен, которому доступны `cookie`.
6. `bool` протокол – указывает на то, что значение `cookie` должно передаваться от клиента по защищенному HTTPS соединению. Если задано `TRUE`, `cookie` от клиента будет передано на сервер, только если установлено защищенное

соединение. При передаче cookie от сервера клиенту следить за тем, чтобы cookie этого типа передавались по защищенному каналу, должен программист веб-сервера.

7. `bool http only` – если задано TRUE, cookie будут доступны только через HTTP протокол. То есть cookie в этом случае не будут доступны скриптовым языкам, вроде JavaScript. Эта возможность была предложена в качестве меры, эффективно снижающей количество краж личных данных посредством XSS атак (несмотря на то, что поддерживается не всеми браузерами). Стоит однако же отметить, что вокруг этой возможности часто возникают споры о ее эффективности и целесообразности. Аргумент добавлен в PHP 5.2.0. Может принимать значения TRUE или FALSE.

Только первый параметр обязателен. Если упустить третий параметр, то по умолчанию он будет равным нулю, это значит что cookie исчезнут сразу после закрытия браузера.

Вот пример установки cookie:

```
<?php
setcookie("message", "welcome", time() + 60);
echo "cookie установлены";
```

[смотреть пример](#)

Этот пример только устанавливает cookie на компьютер клиента.

Далее мы рассмотрим более содержательные примеры.

Получение cookie в PHP

В прошлом примере мы отправили браузеру команду создать cookie. Теперь он их создал, и когда в следующий раз браузер отправит запрос серверу, то он передаст содержимое cookie в заголовке запроса.

В PHP-программе можно определить, прислал браузер cookie или нет очень просто. Они доступны в массиве `$_COOKIE`.

Давайте усложним прошлый пример:

```
<?php
setcookie("message", "welcome", time() + 60);
echo "cookie установлены. Приветствие: $_COOKIE[message]";
echo "<pre>";
print_r($_COOKIE);
echo "</pre>";
```

[смотреть пример](#)

Когда вы откроете страницу примера в первый раз, то за приветствием ничего не будет. Это потому что cookie положены на ваш компьютер, но серверу ещё не отправлены.

Хотя в массиве `$_COOKIE` уже может что-то быть, но тогда эти cookie ложит не наш пример.

Только обновив страницу второй раз вы увидите, что PHP программа получила пару `$_COOKIE[message] => welcome`.

Обратите внимание, что cookie положены на одну минуту, а это значит что через минуту, если обновить страницу, приветствие опять будет отсутствовать.

Пожалуй вот и всё что касается получения доступа к информации из cookie в PHP.

Удаление cookie в PHP

Чтобы удалить cookie нужно просто поставить их срок жизни в прошедшем времени.

```
<?php
setcookie("message", "welcome", time() - 60*60*24*32);
echo "cookie удалены";
?>
```

[смотреть пример](#)

Обратите внимание, я поставил прошедшее время с запасом, на случай если на компьютере клиента время установлено не точно.

Пример работы cookie в PHP

Напишем примитивный счётчик просмотра страниц сайта посетителем, используя cookie.

```
<?php
if (!$_COOKIE[counter]) $counter = 1;
else $counter = ++$_COOKIE['counter'];
setcookie("counter", $counter, time() + 60);
echo "Визит страницы: $counter";
```

[смотреть пример](#)

Обратите внимание, что данные в этом примере живут 60 секунд, потом счётчик обнулится.

На этом тема работа с cookie в PHP раскрыта практически полностью.

Далее нужно изучать [сессии в PHP](#).

[Все скрипты примером](#)

[Первоисточник](#)