

Лекция 3

Cookie

HTTP Cookies

- **HTTP Cookie (куки)** – небольшая порция текстовых данных, отправляемая веб-сервером и хранящаяся в браузере клиента. Браузер всякий раз при открытии страницы соответствующего сайта пересылает сохранённый фрагмент данных обратно веб-серверу через HTTP-заголовки.

HTTP Cookies

Куки используются для:

- аутентификации пользователя;
- хранения персональных предпочтений и настроек пользователя;
- отслеживания состояния сеанса доступа пользователя;
- ведения статистики о пользователях.

Установка Cookie

- В заголовке HTTP-ответа веб-сервера может содержаться указание браузеру сохранить куки:

```
HTTP/1.1 200 OK
```

```
Content-Type: text/html
```

```
Set-Cookie: name=value
```

```
Содержимое страницы
```

Установка Cookie

- Строка Set-Cookie, как правило, добавляется к HTTP-ответу не самим HTTP-сервером, а CGI-программой, работающей вместе с ним. HTTP-сервер только отправляет браузеру результат работы такой программы.

Чтение Cookie

- Строка Set-Cookie отправляется только тогда, когда сервер желает, чтобы браузер сохранил куки. В этом случае браузер запомнит строку name=value и отправит её обратно серверу с каждым последующим запросом.

```
GET /spec.html HTTP/1.1
```

```
Host: www.example.org
```

```
Cookie: name=value
```

- Значение Cookie может быть изменено повторной отправкой сервером «Set-Cookie».

Атрибуты Cookie

- Кроме пары «имя/значение» куки может содержать **срок действия, путь и доменное имя**, на которое оно распространяется. Пример:

```
Set-Cookie: name=newvalue; expires=date;  
path=/; domain=.example.org.
```

Атрибуты Cookie

- Домен и путь говорят браузеру, что куки нужно отправлять обратно на сервер при запросах URL для указанного домена и пути. Если они не указаны, используются домен и путь запрошенной страницы.
- Дата истечения указывается в формате «Нед, ДД Мес ГГГГ ЧЧ:ММ:СС GMT». Например:
- `Set-Cookie: RMID=732423sdfs73242; expires=Fri, 31 Dec 2013 23:59:59 GMT; path=/; domain=.example.net`

Типы Cookie

- **Куки сессии** – существует только на то время, пока пользователь производит навигацию по сайту. Куки сессии создаётся автоматически, если не указан срок действия куки.
- **Постоянные куки** – существует до тех пор, пока не закончится срок действия куки. Например, если куки имеет атрибут Max-Age установленный на 1 год (например), то значение Cookie будет отправляться браузером на Web-сервер при каждом обращении в течение года.

Безопасность Cookie

- Куки легко перехватить и подменить (например, для получения доступа к учетной записи), если пользователь использует нешифрованное соединение с сервером.

Способы задания Cookie

- 1) Через клиентский JavaScript
- 2) Через прямую установку HTTP-заголовков на сервере

Cookie в CGI

- Получение Cookie в среде CGI происходит с помощью переменной окружения **HTTP_COOKIE**, которая в точности повторяет HTTP-заголовок клиента «**Cookie**».
- Формат Cookie имеет следующий вид:

```
name=value; name2=value2
```

Cookie в PHP

- Любые cookies, отправленные серверу браузером клиента, будут автоматически включены в суперглобальный массив `$_COOKIE`

setcookie()

- Задаёт cookie, которое будет передано клиенту вместе с другими HTTP заголовками.
- Как и любой другой заголовок, cookie должны передаваться **до** того как будут выведены какие-либо другие данные скрипта (ограничение протокола).
- Все аргументы, за исключением **name**, являются необязательными. Если нужно пропустить какой-либо аргумент, можно вместо него поставить пустую строку (""), или ноль (0) для **expire**.

setcookie()

- **name** – Наименование cookie.
- **value** – Значение cookie.
- **expire** – Время, когда срок действия cookie истекает.
 - Если задать 0 или пропустить этот аргумент, срок действия cookie истечет с окончанием сессии (при закрытии браузера).
- **path** – Путь к директории на сервере, из которой будут доступны cookie.
 - Если задать '/', cookie будут доступны во всем домене domain.
 - Если задать '/foo/', cookie будут доступны только из директории /foo/ и всех ее поддиректорий (например, /foo/bar/) домена domain.
 - По умолчанию значением является текущая директория, в которой cookie устанавливается.

setcookie()

- **domain** – Домен, которому доступны cookie.
 - Задание домена 'www.example.com' делает cookie доступными в поддомене www и поддоменах более высоких порядков. Cookie доступные низким уровням, таким как 'example.com', будут доступны во всех поддоменах высших уровней, с том числе 'www.example.com'.
- **secure** – 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.
- **httponly** – Если задано TRUE, cookie будут доступны только через HTTP протокол.
 - В этом случае cookie не будут доступны скриптовым языкам, вроде JavaScript.

setcookie()

- Если перед вызовом функции клиенту уже передавался какой-либо вывод (тэги, пустые строки, пробелы, текст и т. п.), setcookie() вызовет отказ и вернет **FALSE**.
- Если setcookie() успешно отработает, то вернет **TRUE**.
 - НО! Это не означает, что клиентское приложение (браузер) правильно приняло и обработало cookie.

Пример

```
<?php
$value = 'что-то где-то';

setcookie("TestCookie", $value);

setcookie("TestCookie", $value, time()+3600);
    /* срок действия 1 час */

setcookie("TestCookie", $value, time()+3600,
"/~rasmus/", "example.com", 1);
?>
```

Вывод cookie

```
<?php
// Вывести одно конкретное значение cookie
echo $_COOKIE["TestCookie"];

// Устанавливаем Cookie до конца сессии:
// В случае успешной установки Cookie, функция
SetCookie возвращает TRUE:
if (SetCookie("Test","Value")) echo "<h3>Cookies
успешно установлены!</h3>";

/*
    В целях тестирования и отладки может
    пригодиться вывод всех cookie
*/
print_r($_COOKIE);
?>
```

Удаление cookies

- Чтобы удалить cookie достаточно в качестве срока действия указать какое-либо время в прошлом.

```
<?php
// установка даты истечения срока действия
на час назад
setcookie ("TestCookie", "", time () - 3600)
;
setcookie ("TestCookie", "", time () - 3600,
"/~rasmus/", "example.com", 1);
?>
```

setcookie() и массивы

Для того, чтобы использовать каждому cookie нужно дать имя в соответствии с правилами именования массивов. Такая возможность позволяет поместить столько значений, сколько имеется элементов в массиве.

```
<?php
// отправка cookie
setcookie("cookie[three]", "cookiethree");
setcookie("cookie[two]", "cookietwo");
setcookie("cookie[one]", "cookieone");

// после перезагрузки страницы, выведем cookie
if (isset($_COOKIE['cookie'])) {
    foreach($_COOKIE['cookie'] as $name => $value) {
        $name = htmlspecialchars($name);
        $value = htmlspecialchars($value);
        echo "$name : $value <br />\n";
    }
}
?>
```

Лабораторная работа

- Сделать браузерную мини-игру
- При первом заходе на страницу спросить логин
 - Записать логин в Cookie
 - Приветствовать уже представленных посетителей
 - Сделать возможным удаление информации о посетителе (ручной сброс)
- Реализовать при помощи форм интерактивность
 - Обеспечить сохранение информации
 - Обеспечить безопасность (не допускать взлом игровых параметров)