**Что такое cookies и как с ними работать**

Откуда возник термин "cookie" никто достоверно не знает, хотя считается, что во времена зарождения [Unix-систем](http://citforum.ru/operating_systems/unix.shtml) где-то использовалось словосочетание Magic Cookies. Имелись в виду "квитанции" (token, ticket), которыми обменивались программы.

Cookie является решением одной из наследственных проблем [HTTP протокола](http://citforum.ru/internet/cgi_tut/http.shtml) (HyperText Transfer Protocol). Эта проблема заключается в непостоянстве соединения между клиентом и сервером, как при [FTP](http://citforum.ru/internet/ftp/index.shtml) или Telnet сессии, т.е. для каждого документа (или файла) при передаче по HTTP протоколу посылается отдельный запрос. Включение cookie в HTTP протокол дало частичное решение этой проблемы. Иначе говоря, транзакция завершается после того, как браузер сделал запрос, а сервер выдал соответствующий ответ. Сразу после этого сервер "забывает" о пользователе и каждый следующий запрос того же пользователя считает новым пользователем.

Используя cookie, можно эмулировать сессию по HTTP протоколу. Коротко принцип эмуляции сессии таков: на первом запросе выдается соотвествующее значение cookie, а при каждом последующем запросе это значение читается из переменной окружения HTTP\_COOKIE и соответствующим образом обрабатывается.

Простой пример: есть форма, где пользователю предлагается указать свое имя, из нее вызывается скрипт, прописывающий значение cookie в браузер пользователя. При каждом последующем заходе на основе анализа значения cookie из браузера пользователя на странице появляется либо именное приветствие (если есть установленное значение cookie), либо первоначальная форма с запросом имени пользователя (если значение cookie не установлено).

Cookie - это небольшая порция текстовой информации, которую сервер передает браузеру. Браузер будет хранить эту информацию и передавать ее серверу с каждым запросом как часть HTTP заголовка. Одни значения cookie могут храниться только в течение одной сессии, они удаляются после закрытия броузера. Другие, установленные на некоторый период времени, записываются в файл. Обычно этот файл называется 'cookies.txt' и лежит в рабочей директории установленного на компьютер браузера.

**Что можно делать с помощью cookie?**  
Сами по себе cookies не могут делать ничего, это только лишь некоторая текстовая информация. Однако сервер может считывать содержащуюся в cookies информацию и на основании ее анализа совершать те или иные действия. Например, в случае авторизованного доступа к чему либо через WWW в cookies сохраняется login и password в течение сессии, что позволяет пользователю не вводить их снова при запросах каждого документа, защищенного паролем.

На использовании cookies также часто строят функции оформления заказов в онлайновых магазинах, в частности, в самом крупном виртуальном книжном магазине [Amazon Books](http://www.amazon.com/) реализована своеобразная виртуальная корзина покупателя, как в обычном реальном супермаркете, в которую сервер записывает информацию обо всех заказанных книгах. Пользователь просто помечает интересующие его книги, а затем оформляет покупку сразу всех отмеченных книг.

Еще одна распространенная область использования cookies - при настройке индивидуального профиля каждого зарегистрированного пользователя.

И, наконец, самая последняя область - использование механизма cookie в рекламном бизнесе на Интернет. Еще год назад реклама в Интернет за деньги была довольно экзотической услугой, а сейчас этот бизнес уже устоялся и стремительно развивается. Однако рекламодатели начинают предъявлять более жесткие условия к оценке эффективности своих расходов. Cookie используются для таргетинга рекламы (определения целевой аудитории, например, по географическому положению пользователей), отслеживания интересов пользователей, учета количества показов и проходов сквозь баннеры.

**Работа с cookie**  
Теперь, когда с принципами действия и областями применения cookie все более или менее понятно, можно приступить к изучению формата и синтаксиса, а также способов задания значений cookie.

*Формат и синтаксис cookie*  
Предлагаемое мной в этой статье описание формата и синтаксиса cookie является вольным пересказом изначальной спецификации Netscape Communications "[Persistent Client State HTTP Cookies](http://www.netscape.com/newsref/std/cookie_spec.html)". В настоящий момент идет разработка более строгой спецификации для cookie. Итак, cookie является частью HTTP заголовка.

**NAME=VALUE** - строка символов, исключая перевод строки, запятые и пробелы. NAME-имя cookie, VALUE - значение. Не допускается использование двоеточия, запятой и пробела.

**expires=DATE** - время хранения cookie, т.е. вместо DATE должна стоять дата в формате "expires=Monday, DD-Mon-YYYY HH:MM:SS GMT", после которой истекает время хранения cookie. Если этот атрибут не указан, то cookie хранится в течение одного сеанса, до закрытия броузера.

**domain=DOMAIN\_NAME** - домен, для которого значение cookie действительно. Например, "domain=cit-forum.com". В этом случае значение cookie будет действительно и для домена cit-forum.com, и для www.cit-forum.com. Но не радуйтесь, указания двух последних периодов доменных имен хватает только для [доменов иерархии](http://citforum.ru/internet/dns/toplevel.shtml) "COM", "EDU", "NET", "ORG", "GOV", "MIL" и "INT". Для обсуждаемых сейчас новых семи доменов первого уровня ("FIRM", "SHOP", "WEB", "ARTS", "REC", "INFO", "NOM"), вероятно, это условие сохранится. Для доменов иерархии "RU", например, придется указывать три периода.

Если этот атрибут опущен, то по умолчанию используется доменное имя сервера, на котором было задано значение cookie.

**path=PATH** - этот атрибут устанавливает подмножество документов, для которых действительно значение cookie. Например, указание "path=/win" приведет к тому, что значение cookie будет действительно для множества документов в директории /win/, в директории /wings/ и файлов в текущей директории с именами типа wind.html и windows.shtml. Для того, чтобы cookie отсылались при каждом запросе к серверу, необходимо указать корневой каталог сервера, например, "path=/".

Если этот атрибут не указан, то значение cookie распространяется только на документы в той же директории, что и документ, в котором было установлено значение cookie.

**secure** - если стоит этот маркер, то информация cookie пересылается только через HTTPS (HTTP с использованием SSL - Secure Socket Level), в защищенном режиме. Если этот маркер не указан, то информация пересылается обычным способом.

*Синтаксис HTTP заголовка для поля Cookie*  
Когда запрашивается документ с HTTP сервера, браузер проверяет свои cookie на предмет соответствия домену сервера и прочей информации. В случае, если найдены удовлетворяющие всем условиям значения cookie, броузер посылает их в серверу в виде пары имя/значение:

Cookie: NAME1=OPAQUE\_STRING1; NAME2=OPAQUE\_STRING2 ...

Одновременно можно задавать несколько значений cookie.

В случае, если cookie принимает новое значение при имеющемся уже в браузере cookie с совпадающими параметрами NAME, domain и path, то старое значение заменяется новым. В остальных случаях новые значения cookie добавляются к старым.

Использование expires не гарантирует сохранность cookie в течение заданного периода времени, поскольку клиент (браузер) может удалить запись из-за нехватки выделенного места или каких-либо других причин.

Клиент (браузер) имеет следующие ограничения для cookies:

* всего может храниться до 300 значений cookies
* каждый cookie не может превышать 4Кбайт
* с одного сервера или домена может храниться до 20 значений cookie

Если ограничение 300 или 20 превышается, то удаляется первая по времени запись. При превышении лимита объема в 4Кбайт корректность значения cookie страдает - отрезается кусок записи (с начала этой записи) равный превышению объема.

В случае кэширования документов, например, proxy-сервером, поле Set-cookie HTTP заголовка никогда не кэшируется.

Если proxy-сервер принимает ответ, содержащий поле Set-cookie в заголовке, предполагается, что поле доходит до клиента вне зависимости от кода возврата 304 (Not Modified) или 200 (OK). Соответственно, если клиентский запрос содержит в заголовке Cookie, то он должен дойти до сервера, даже если жестко установлен параметр If-modified-since.