

Переменные окружения в PHP

HTTP_ACCEPT

- В этой переменной перечислены все MIME-типы данных, которые могут быть восприняты браузером. Строка */* означает, что браузер понимает любой тип.

HTTP_ACCEPT= image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash, application/vnd.ms-excel, application/msword, */*

HTTP_REFERER

- Эта переменная представляет сведения о странице, с которой пользователь попал на данную.

HTTP_REFERER= <http://www.spravkaweb.ru/php/pril/>

HTTP_COOKIE

- В этой переменной хранятся все Cookies в URL-кодировке.

HTTP_COOKIE= hotlog=1; ZDEDebuggerPresent=php,phtml,php3; b=b;
PHPSESSID=16805922a9258cda274316e60f649cf8

HTTP_USER_AGENT

- Идентифицирует браузер пользователя. Для установления типа браузера нужно проверить эту строку на наличие слов: если браузер - Internet Explorer, то будет присутствовать подстрока MSIE, а если в наличии лишь слово Mozilla, то это Netscape.

HTTP_USER_AGENT= Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; MyIE2; Maxthon)

HTTP_HOST

- Содержит доменное имя Web-сервера, на котором запустился сценарий. Эту переменную достаточно удобно использовать, например, для генерации полного пути, который требуется в заголовке Location, чтобы не привязываться к конкретному серверу.

HTTP_HOST= www.spravkaweb.ru

HTTP_FROM

- Адрес электронной почты пользователя, направившего запрос.

SERVER_NAME

- Доменное имя или IP-адрес сервера.

SERVER_NAME= www.spravkaweb.ru

SERVER_SOFTWARE

- Имя и версия программы-сервера, отвечающей на запрос клиента.

SERVER_PORT

- Эта переменная содержит порт сервера, к которому обратился браузер пользователя. Обычно это 80. Переменная так-же может применяться для формирования параметра заголовка Location.

SERVER_PORT= 80

SERVER_PROTOCOL

- Переменная содержит имя и версию информационного протокола, который был использован для запроса.

SERVER_PROTOCOL= HTTP/1.1

REMOTE_ADDR

- Эта переменная содержит IP-адрес (или доменное имя) узла пользователя, на котором был запущен браузер.

REMOTE_PORT

- Порт, который закрепляется за браузером пользователя для получения ответа сервера.

REMOTE_USER

- Идентификационное имя пользователя, посылающего запрос.

SCRIPT_NAME

- Содержит имя файла, содержащего данный сценарий. Эту переменную удобно использовать при формировании заголовка Location при переадресации на себя (self-redirect), а также для подставления значения атрибута action тега <form> на странице, которую вызывает сценарий при запуске без параметров (для того чтобы не привязываться к конкретному имени сценария).

DOCUMENT_ROOT

- Корневой каталог дерева Web-документов.

REQUEST_METHOD

- Метод, который применяет пользователь при передаче данных. Надо заметить, что грамотно составленный сценарий должен сам определять на основе этой переменной, какой метод задействует пользователь, и принимать данные из соответствующего источника, а не рассчитывать, что передача будет осуществляться, например, только методом POST.

QUERY_STRING

- Содержит параметры, которые в URL указаны после вопросительного знака. Напомним, что они доступны как при методе GET, так и при методе POST (если в последнем случае они были определены в атрибуте action тега <form>).

PATH_INFO

- Содержит дополнительную информацию о пути.

PATH_TRANSLATED

- Та же информация, что и в переменной PATH_INFO с префиксом, задающим путь к корневому каталогу дерева Web-документов.

CONTENT_TYPE

- Медиа-тип данных запроса.

CONTENT_LENGTH

- Возвращает количество байт данных, присланных пользователем. Эту переменную необходимо анализировать, если вы занимаетесь приемом и обработкой POST-формы.

GATEWAY_INTERFACE

- Версия CGI, которую использует сервер.

Пример использования переменных окружения

```
<html>
<head><title>Переменные окружения</title></head>
<body>
<h1>Переменные окружения:</h1>
<hr>
<?
echo "Вы пришли из: ".getenv("HTTP_REFERER")."<BR>";
$br=getenv("HTTP_USER_AGENT");
if(strpos($br,"MSIE")!=false) echo "У вас Internet Explorer<BR>";
else echo "У вас Firefox или др.<BR>";
echo "Ваш IP: ".getenv("REMOTE_ADDR")."<BR>";
echo "Вот параметры в строке браузера: ".getenv("QUERY_STRING");
?>
</body>
</html>
```

Для исполнения php-кода в htm-файле добавляем в корневой .htaccess строку:
AddHandler application/x-httpd-php .php .shtml .htm

Массивы создаваемые PHP на основе данных, пришедших из формы:

- `$_GET` — содержит GET-параметры, пришедшие скрипту через переменную окружения `QUERY_STRING`. Например, `$_GET['login']`;
- `$_POST` — данные формы, пришедшие методом POST;
- `$_COOKIE` — все cookies, которые прислал браузер;
- `$_REQUEST` — объединение трех перечисленных выше массивов;
- `$_SERVER` — содержит переменные окружения, переданные сервером.

Заголовки http-протокола

GET

GET *сценарий?параметры* HTTP/1.0

Переменные окружения: `REQUEST_URI`; в переменной `QUERY_STRING` сохраняется значение *параметры*, в переменной `REQUEST_METHOD` — ключевое слово GET.

POST

POST *сценарий!параметры* HTTP/1.0

Переменные окружения: `REQUEST_URI`; в переменной `QUERY_STRING` сохраняется значение *параметры*, в переменной `REQUEST_METHOD` — слово POST.

Content-Type

Content-Type: application/x-www-form-urlencoded

Переменная окружения: `CONTENT_TYPE`

Host

Host: *имя_хоста*

Переменная окружения: `HTTP_HOST`.

User-Agent

User-Agent: Mozilla/4.5 [en] (Win95; I)

Переменная окружения: `http_user_agent`.

Referer

Referer: *URL_адрес*

Переменная окружения: `HTTP_REFERER`.

Content-length

Content-length: *длина*

Переменная окружения: `CONTENT_LENGTH`.

Cookie

Cookie: *значения_cookies*

Переменная окружения: `HTTP_COOKIE`.

Accept

Accept: *text/html, text/plain, image/gif, image/jpeg*

Переменная окружения: `HTTP_ACCEPT`.

Функции PHP для работы с MySQL

Функции соединения с сервером MySQL

```
mysql_connect ([string $hostname] [, string $user] [, string $password]);
```

Функция возвращает идентификатор (типа int) соединения, вся дальнейшая работа осуществляется только через этот идентификатор.

Для закрытия соединения предназначена функция

```
mysql_close(int $connection_id);
```

Функция выбора базы данных

```
mysql_select_db (string $db [, int $id])
```

выбирает базу данных, с которой будет работать PHP скрипт. Если открыто не более одного соединения, можно не указывать параметр \$id.

Пример:

```
// Попытка установить соединение с MySQL:
if (!mysql_connect($server, $user, $password)) {
echo "Ошибка подключения к серверу MySQL";
exit; }
// Соединились, теперь выбираем базу данных:
mysql_select_db($db);
```

Функции выполнения запросов к серверу баз данных

mysql_query(\$q).

Этой функции нужно передать один параметр - текст запроса. Текст запроса может содержать пробельные символы и символы новой строки (\n). Текст должен быть составлен по правилам синтаксиса SQL. Пример запроса:

```
$q = mysql_query("SELECT * FROM mytable");
```

Функции обработки результатов запроса

mysql_result() - получить необходимый элемент из набора записей;

mysql_fetch_array() - занести запись в массив;

mysql_fetch_row() - занести запись в массив;

mysql_fetch_assoc() - занести запись в ассоциативный массив;

mysql_fetch_object() - занести запись в объект.

Функция **mysql_num_rows()** позволяет узнать, сколько записей содержит результат запроса:

Запись состоит из полей (колонок). С помощью функции **mysql_num_fields()** можно узнать, сколько полей содержит каждая запись результата:

Узнать значение каждого поля можно с помощью следующей функции:

```
mysql_result (int $result, int $row, mixed $field);
```

Параметр функции \$row задает номер записи, а параметр \$field - имя или порядковый номер поля.

Если SQL-запрос вернул следующий набор данных:

Email	Name	Last_Name
ivanov@mail.ru	Ivan	Ivanov
petrov@mail.ru	Petr	Petrov

Вывести это в браузер можно следующим образом:

```
$rows = mysql_num_rows($q); $fields = mysql_num_fields($q);
```

```
echo "<pre>";
```

```
for ($c=0; $c<$rows; $c++) {
    for ($cc=0; $cc<$fields; $cc++) {
        echo mysql_result($q, $c, $cc)."\t";
        echo "\n";
    }
}
echo "</pre>";
```