Взаимодействие с пользователем

URL и строка запроса

<схема>://<логин>:<пароль>@<хост>:<порт>/<URL-путь>?<параметры>#<якорь>

В URL можно использовать только: *латинские буквы*, *цифры* и лишь *некоторые знаки препинания*, все остальные символы должны быть перекодированы особым образом.

Параметры – *строка запроса* с передаваемыми на сервер параметрами. *Разделитель параметров* – знак *&*.

```
?param_1=value_1&param_2=value_2&param_3=value_3
```

Спецификацией HTTP определен ряд методов для передачи запросов между компьютерами, но чаще всего из них используются только два:

- GET – через адресную строку браузера (получение данных по URL);
- POST – отправка данных с помощью HTML-форм.

Суперглобальный массив $_GET

Ассоциативный массив параметров, переданных скрипту через URL.

```
# http://localhost/?name=Kirill

echo 'Hello ' . htmlspecialchars($_GET['name']);

# Выводит: "Hello Kirill"
```

Суперглобальный массив $_POST

Ассоциативный массив данных, переданных скрипту методом HTTP POST.

```
<form action="index.php" method="post">
    Имя: <input name="name" type="text" />
    <input type="submit" value="Send" />
</form>
```

```
# Например в форму ввели Kirill

echo 'Hello ' . htmlspecialchars($_POST['name']);

# Выводит: "Hello Kirill"
```

Функции обработки URL

```
string urlencode ( string str )
```

URL-кодирование строки (RFC 1738 - application/x-www-form-urlencoded).

```
string rawurlencode ( string str )
```

URL-кодирование строки согласно RFC 3986.

```
string urldecode ( string str )
string rawurldecode ( string str )
```

Дерадирование URL-кодированной строки.

```
array parse_url ( string url [, int component ] )
```

Возвращает ассоциативный массив, содержащий все компоненты URL.

```
string http_build_query (
    array formdata [, string numeric_prefix
    [, string arg_separator [, int enc_type ]]]
)
```

Возвращает URL-кодированную строку из переданного массива formdata.

PHP_QUERY_RFC1738 (по-умолчанию), PHP_QUERY_RFC3986.

Обработка запросов пользователя

Никогда не доверяйте пользователю!

Всегда проверяйте входные данные!
Валидируйте или очищайте!

Никогда не работайте на прямую
с суперглобальными массивами!

Убедитесь, что PHP настроен верно!

Соблюдение этих простых правил
обеспечит эффективную защиту
от «прямых» атак

Обработка запросов пользователя

string trim (string str [, string charlist])

Удаляет пробелы (или другие символы) из начала и конца строки.

bool empty (mixed var)

Проверка переменной на пустое значение.

register_globals boolean

Регистрировать или нет переменные EGPCS как глобальные переменные.

Была помечена устаревней в PHP 5.3.0 и была удалена в PHP 5.4.0.

- Используйте isset() для проверки введенных пользователем данных;
- Выполняйте переприсвоение данных из суперглобальных массивов локальным переменным;
- Можете выполнять очистку суперглобальных массивов.

HTML-конвертация

```
string strip_tags ( string str [, int allowable_tags ] )
```

Удаляет HTML и PHP теги из строки.

```
string htmlspecialchars (
    string str [, int flags [, string encoding [, bool double_encode ]]]
)
```

Преобразует специальные символы в HTML-сущности.

```
string htmlentities (
    string str [, int flags [, string encoding [, bool double_encode ]]]
)
```

Преобразует все возможные символы в HTML-сущности.

```
string get_html_translation_table (
    int table [, int flags [, string encoding ]]
)
```

Возвращает таблицу преобразований, используемую функциями htmlspecialchars() и htmlentities().

Экранирование спецсимволов

string addslashes (string str)

Экранирует специальные символы в строке, добавляя перед ними обратный слэш.

string mysqli_real_escape_string (mysqli link [, string escapestr])

Экранирует специальные символы в строке для использования в SQL выражении, используя текущий набор символов соединения.

string stripslashes (string str)

Удаляет экранирование символов.

Конец