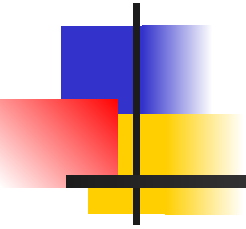


PHR: сессии



Краткий обзор

Сеансы и HTTP

- Необходимость сеансов:
 - Авторизация пользователя в течение сеанса
 - Хранение переменных сеанса
- Пути установления сеансов
 - Ключики
 - Параметры GET или POST
- Где хранить переменные сеанса?

Идея сессий

- Формируем метку, идентифицирующую сеанс работы – номер сессии;
- Отправляем метку – номер сессии на сторону клиента
 - Вариант 1: ставим ключик с номером сессии
 - Вариант 2: добавляем параметр – номер сессии в каждый вызов с локальным адресом;
- Анализируем HTTP запрос на предмет наличия метки – номера сессии
- Можем связывать с меткой наборы переменных
- Следим за сроком годности
- Все это реализуемо самостоятельно, но есть встроенное в PHP
- Функции работы с сессиями – более 20 штук

Создание сессии и её идентификатор: `session_start()`

- **`bool session_start(void)`** – стартует новую или возобновляет существующую сессию;
 - без параметров
 - возвращает **TRUE** в случае успешного выполнения, иначе **FALSE**
- Каков алгоритм действия препроцессора PHP при старте сессии?
- **`string session_id ([string $id])`**
 - параметр `$id` определяет(заменяет) идентификатор сессии;
 - возвращает установленный идентификатор текущей сессии, если она существует, иначе пустая строка;
 - идентификатор сессии – строка 32 символа длиной, буквы и цифры, регистрозависим;
 - рекомендуется использовать для проверки наличия текущей сессии.

Создание сессии: пример

```
<php?
if (session_id()=="")
    { session_start();
      // создаем или восстанавливаем сессию
    echo session_id(); }
// выводим идентификатор сессии
?>
```

Чтение/установка имени сессии: `session_name()`

- Что такое «имя сессии»?
- По умолчанию используется имя PHPSESSID
- `string session_name ([string $name])`
 - параметр `$name` определяет новое имя сессии
 - если функции не передается параметр, она возвращает имя текущей сессии, в противном случае – прежнее имя сессии

Переменные сессии

- Идея переменных сессии
- Суперглобальный массив `$_SESSION`
 - `$_SESSION['login']=$_GET['LOGIN'];`
 - Только строчные индексы!

Переменные сессии: пример

```
<PHP?
```

```
session_start(); // создаем сессию
```

```
$_SESSION['login']=$_GET['LOGIN']; // регистрируем переменную login
```

```
$_SESSION['passwd']=$_GET['PASSWORD'];
```

```
?>
```

```
<PHP?
```

```
session_start(); // восстанавливаем текущую сессию
```

```
$login=$_SESSION['login'];
```

```
$password=$_SESSION['passwd'];
```

```
echo $login, $password
```

```
?>
```


Удаление переменных сессии: `session_unset`

- `void session_unset (void)`
 - освобождает все зарегистрированные переменные сессии
 - функция не имеет параметров
 - функция не возвращает значений
- Не используйте `unset($_SESSION)` – этот суперглобал удалять нельзя!
- Допустимо удалять отдельные зарегистрированные переменные сессии:
`unset($_SESSION[$name])`

Удаление сессии: `session_destroy`

- `bool session_destroy (void)`
 - завершает текущую сессию
 - не имеет параметров
 - возвращает **TRUE** в случае успешного завершения, иначе **FALSE**
- переменные сессии не освобождаются до конца работы скрипта, вызвавшего функцию. Для этого можно использовать `session_unset()`

Работа с сессиями в PHP

- `session_start()`
- `session_id()`
- `session_name()`
- `session_unset()`
- `session_destroy()`
- `$_SESSION`