

RНР дополнительные темы

Включение файлов, сессии, куки

Включение файлов

- **include**

- включение файла **не является обязательным** для работы php-скрипта
- если файл не удастся включить – возникает предупреждение (**warning**)

```
include 'filename';
```

- **require**

- включение файла **необходимо для работы** php скрипта
- если файл не удастся включить – возникает ошибка (**error**), которая приводит к завершению работы php

```
require 'filename';
```

Включение файлов. Примеры

- footer.php

```
<?php  
echo "<p>Copyright &copy; 1999-" . date("Y") . " ИУ-4</p>";  
?>
```

- Включение файла

```
<html>  
<body>  
  
<h1>Заголовок</h1>  
<p>Контент.</p>  
<p>Контент.</p>  
<?php include 'footer.php';?>  
  
</body>  
</html>
```

Включение файлов. Примеры

- db.php

```
<?php
    $servername = "localhost";
    $username = "username";
    $password = "password";
    $dbname = "cities";
?>
```

- Включение файла

```
<?php
require 'db.php';
$conn = mysqli_connect($servername, $username, $password, $dbname);

// Используем подключение к MySQL
// ...
?>
```

Cookie. Куки

- **Cookie** – небольшой объем данных, отправленный веб-сервером и хранимый на компьютере пользователя
- Веб-клиент (обычно веб-браузер) всякий раз при попытке открыть страницу соответствующего сайта пересылает **cookie** веб-серверу в составе HTTP-запроса

Cookie. Применение

- Аутентификации пользователя
- Хранения персональных предпочтений и настроек пользователя
- Отслеживания состояния сеанса доступа пользователя
- Ведения статистики о пользователях

Cookie. Использование в PHP

- Синтаксис

`setcookie(name, value, expire, path, domain, secure, httponly);`

- **name** – обязательный параметр, остальные необязательные
- **Важно:** функция **setcookie** должна быть перед тегом **<html>**

Cookie. Использование в PHP

- Установка **cookie**

```
<?php
$cookie_name = "user";
$cookie_value = "John Doe";
$cookie_expire = time() + (86400 * 30); // 86400 = 1 день
$path = "/";
setcookie($cookie_name, $cookie_value, $cookie_expire, $path);
?>
```


Cookie. Использование в PHP

- Использование **cookie**

```
<?php
if(!isset($_COOKIE[$cookie_name])) {
    echo "Cookie '" . $cookie_name . "' не установлены!";
} else {
    echo "Cookie '" . $cookie_name . "' установлены!<br>";
    echo "Значение: " . $_COOKIE[$cookie_name];
}
?>
```

Session. Использование в PHP

- **Сессия** – способ хранения и передачи данных между разными страницами
- В отличие от **cookie** данные хранятся на сервере, а не на компьютере пользователя

Начало работы с сессией в PHP

```
<?php
    // Функция для начала работы с сессией
    session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
    // Задание переменных в сессии
    $_SESSION["favcolor"] = "зеленый";
    $_SESSION["favourite"] = "кошка";
    echo "Переменные сессии установлены.";
?>

</body>
</html>
```

Использование переменных сессии

```
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>
```

```
<?php
// Вывод переменных, установленных на предыдущей странице
echo "Любимый цвет" . $_SESSION["favcolor"] . "<br>";
echo "Любимое животное" . $_SESSION["favourite"] . ".";
?>
```

```
</body>
</html>
```

Удаление сессии в PHP

```
<?php  
session_start();  
?>
```

```
<!DOCTYPE html>  
<html>  
<body>
```

```
<?php  
// удаление всех переменных в сессии  
session_unset();
```

```
// удаление сессии  
session_destroy();  
?>
```

```
</body>  
</html>
```

Регулярные выражения – это ...

- Формальный язык поиска и осуществления манипуляций с подстроками в тексте, основанный на использовании метасимволов (wildcard characters)
- Строка-образец (pattern) – шаблон или маска, состоящая из символов и метасимволов и задающая правило поиска

Синтаксис

- «.» - любой символ
- | – используется для задания перечислений
- () скобки используются для обозначения приоритета операций и группировки

`gray | grey gr(a | e)y`

- \ – используется для экранирования

`\.com | \.ru`

Повторение шаблона

Представление	Число повторений
{n}	Ровно n раз
{n, m}	От m до n включительно
{n, }	Не менее n
{, m}	Не более m

`(bla){2, 3}`

`blabla`

`blablabla`

~~`bla`~~

~~`blablablabla`~~

Повторение шаблона

Представление	Соответствие	Число повторений
?	{0, 1}	Ноль или одно
*	{0,}	Ноль или более
+	{1,}	Одно или более

`(bla-?)+`

`abc`

`bla`

`blabla`

`bla-bla-blabla`

Символьные классы

- `[]` – символьный класс, набор символов в скобках
- `[^]` – символы, которые не входят в набор или символьный класс

Символьные классы

Представление	Эквивалент	Описание
<code>\d</code>	<code>[0-9]</code>	Цифра
<code>\D</code>	<code>[^\d]</code>	Любой символ, кроме цифры
<code>\w</code>	<code>[A-Za-zА-Яа-я0-9_]</code>	Символы, образующие «слово» (буквы, цифры и символ подчёркивания)
<code>\W</code>	<code>[^\w]</code>	Символы, не образующие «слово»
<code>\s</code>	<code>[\t\v\r\n\f]</code>	Символ пустого пространства
<code>\S</code>	<code>[^\s]</code>	Любой символ, не соответствующий символу пустого пространства

Символьные классы POSIX

Представление	Эквивалент	Описание
[[:upper:]]	[A-Z]	Верхний регистр
[[:lower:]]	[a-z]	Нижний регистр
[[:alpha:]]	[[:upper:]][[:lower:]]	Буквы
[[:digit:]]	[0-9], т.е. \d	Цифры
[[:xdigit:]]	[[:digit:]]A-Fa-f	Шестнадцатиричные цифры
[[:alnum:]]	[[:alpha:]][[:digit:]]	Буквы и цифры
[[:word:]]	[[:alnum:]]_, т.е. \w	Символы, образующие «слово»
[[:punct:]]	[-!"#\$%&'()*+,-./:;<=>?@[\\ \]_`{ }~]	Знаки пунктуации
[[:blanc:]]	\t	Пробелы
[[:space:]]	[[:blanc:]\v\r\n\f], т.е. \s	Символы пустого пространства
[[:cntrl:]]	[\x00-\x1F\x7F]	Управляющие символы
[[:graph:]]	[\x21-\x7E]	Печатные символы
[[:print:]]	[\x20-\x7E], т. е. [[:graph:]]	Печатные символы с пробелом

Позиция внутри строки или текста

Обозначение	Описание
<code>^</code>	Начало текста
<code>\$</code>	Конец текста
<code>\b</code>	Граница слова
<code>\B</code>	Не граница слова
<code>\G</code>	Предыдущий успешный поиск

Представление символов по коду

Представление	Описание	Кодировка
\0xn	n – 8-ное число	8-битная
0\xdd	d – 16-ная цифра	
0\xdddd		16-битная

Управляющие символы

Представление	Описание
<code>\t</code>	Табуляция
<code>\v</code>	Вертикальная табуляция
<code>\n</code>	Переход на новую строку
<code>\b</code>	Удаление (backspace) Должен находиться внутри квадратных скобок (иначе интерпретируется как граница слова)
<code>\r</code>	Возврат каретки
<code>\f</code>	Конец страницы
<code>\a</code>	Звонок
<code>\e</code>	Escape-символ
<code>\cA ... \cZ</code>	Cntrl + A ... Cntrl + Z Эквивалентно <code>\x01 ... \x1A</code>

Регулярные выражения в PHP

- Функция **preg_match** – проверка строки на соответствие шаблону

```
if (!preg_match("/^[a-zA-Z ]*$/", $name)) {  
    $nameErr = "В имени пользователя допустимы только  
        буквы, цифры и пробелы.";  
}
```