

одобная разметка была :

```
<div style="margin:10px; padding:20px; border-color:#00ff00;">
    Контент
</div>
```

други способ определения стилей:

```
<style>
    div.con {margin:10px; padding:20px; border-color:#00ff00;}
</style>
<div class="con">
    Контент
</div>
```

Ну и вынос style-файлов в другие файлы, потом на другие домены...

```
<link href="styles.css" rel="stylesheet">
<div class="con">
    Контент
</div>
```

Очень просто, лейтмотив этих действий - попытка максимально разнести **стилевое оформление** и **разметку html-документов**. Это делает код гораздо чище и намного более читаемым, и это понятно и действительно логично.

Все тоже самое и в отношении серверного программирования. Нужно максимально **разнести php-сценарии** и **html-разметку**. Дело в том, что в сценариях, как правило, выполняется код **получения** данных, которые затем встраиваются в разметку документа для форматированного **вывода**.

Именно поэтому, в своих программах, код-сценария мы будем писать в начале файла, а в теле html-документа будем осуществлять **вывод сформированных данных** (будем делать так, где это возможно).

Постановка задачи.

Выполнив анализ html-разметки файлов нашего проекта можно выявить четыре ярко выраженных блока: **head, header, content, footer**.

Поместим **контент (содержимое)** этих блоков в соответствующие php-переменные.

Для этого создайте в верхней части **корневого индексного файла** теги php-скрипта. Сохраните html-разметку выявленных блоков в php-переменные.

Итого скрипт будет содержать инициализацию четырех переменных:

- \$head = '';
- \$header = '';
- \$content = '';
- \$footer = '';

Пример инициализации переменной \$head представлен на рисунке 1.

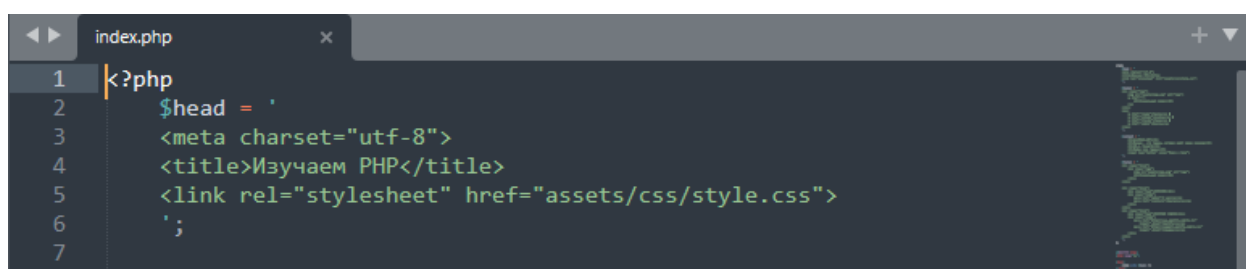


Рис.1. Инициализация переменных

Далее в теле html-разметки замените **контент блоков** на вывод **соответствующих переменных** (рис.2).

```

56
57 <!DOCTYPE html>
58 <html lang="ru">
59
60 <head>
61     <?php echo $head; ?>
62 </head>
63
64 <body>
65     <header>
66         <div class="content">
67             <?php echo $header; ?>
68         </div>
69     </header>
70
71     <div class="main">
72         <div class="content">
73             <?php echo $content; ?>
74         </div>
75     </div>
76
77     <footer>
78         <div class="content">
79             <?php echo $footer; ?>
80         </div>
81     </footer>
82 </body>
83
84 </html>

```

Рис. 2. Вывод переменных в шаблон корневой страницы

Сделайте то же самое для индексных файлов директорий: **teams**, **albums**, **tracks**, **admin**. Не забудьте переименовать индексные файлы в файлы с расширением **.php**.

Что произошло.

В php-сценарии мы **сформировали** данные для вывода. В теле html-документа эти данные **вывели**. При дальнейшем погружении в веб-разработку подобные html-фрагменты будут служить для создания **шаблонов (макетов)** html-страниц.

Что дальше.

В дальнейшем данные для вывода будут вынесены из основного файла программы. Данные мы будем получать из **внешних источников - баз данных** или **других файлов**. А потом отдавать эти данные в html-шаблон для форматированного показа запросившего страницу пользователя.

P.S. Так как разметка корневого индексного файла **отличается стилем** от остальных файлов у вас должно получиться два шаблона для вывода (рис.3).

```
94
95 <!DOCTYPE html>
96 <html lang="ru">
97
98 <head>
99     <?php echo $head; ?>
100 </head>
101
102 <body>
103     <header>
104         <div class="content ins">
105             <?php echo $header; ?>
106         </div>
107     </header>
108
109     <div class="main ins">
110         <div class="content ins">
111             <?php echo $content; ?>
112         </div>
113     </div>
114
115     <footer>
116         <div class="content">
117             <?php echo $footer; ?>
118         </div>
119     </footer>
120 </body>
121
122 </html>
```

Рис.3. Шаблон для вывода файлов второго уровня