

Запись на курсы по HTML, CSS, JavaScript, PHP, фреймворкам и CMS,
а также: помощь в поиске работы и заказов, стажировка на реальных проектах→

Верстка JavaScript PHP NodeJs Vue React Laravel WordPress AJAX Парсинг

Бесплатный тренинг "Работа с узлами DOM в JavaScript". Начало 10-го декабря→
Занимательные задачи JavaScript. Перезапуск! Начнем, когда наберется более 100 желающих→

Разработка своего MVC фреймворка на PHP

В предыдущих уроках вы тренировались использовать готовый MVC фреймворк. Как правило, этим PHP программисты и занимаются - берут один из популярных готовых фреймворков, изучают его, а потом используют. В настоящее время есть 3 самых популярных фреймворка: [Laravel](#), Yii и Symfony.

Полученных вами знаний уже достаточно для изучения настоящих фреймворков. При желании или при недостатке времени вы можете не проходить дальнейшие уроки данного учебника по ООП, а перейти сразу на фреймворки.

Далее в данном учебнике мы с вами разработаем свой MVC фреймворк на PHP. Еще раз подчеркну, что данный этап уже не обязателен, но поспособствует закреплению ваших знаний ООП на практическом примере.

Итак, приступим к разработке. На самом деле вы будете разрабатывать не какой-то абстрактный фреймворк, а просто реализуете функционал того учебного фреймворка, который мы использовали в предыдущих уроках.

Вы уже знаете, какой функционал у вас должен быть. Осталось его реализовать. Приступим!

Файл index.php

Для начала в корневой папке вашего домена сделаем файл index.php со следующим содержанием:

```
1  <?php
2      namespace Core;
3
4      error_reporting(E_ALL);
5      ini_set('display_errors', 'on');
6  ?>
```

Задача 101.1

Создайте файл **index.php** и добавьте в него указанные строки.

Настраиваем htaccess

Теперь сделаем так, чтобы все запросы из адресной строки адресовались на наш файл index.php. То есть всё, что бы не вбил пользователь в браузер будет адресоваться на index.php, а код этого файла будет разбираться с тем, какую именно страницу показать пользователю.

Вот содержимое файла htaccess:

```
RewriteEngine On
RewriteBase /

RewriteRule .+ index.php
```

Вспомним теперь, что адреса из папки *project/webroot* не должны перенаправляться на индекс, так как там мы храним стили, скрипты, картинки и тому подобные вещи. Учтем это в нашем htaccess:

```
RewriteEngine On
RewriteBase /

RewriteCond %{REQUEST_URI} !^/project/webroot/
RewriteRule .+ index.php
```



Задача 101.2

Создайте файл **htaccess**. Выполните описанную настройку. Проверьте ее работу: все адреса, кроме адресов из папки *project/webroot* должны редиректится на index.

Автозагрузка классов

Наш учебный фреймворк использует нестандартные правила для автозагрузки классов (расскажите, в чем отличия), поэтому нам придется реализовать свою автозагрузку в файле index.php:

```
1  <?php
2      namespace Core;
3
4      error_reporting(E_ALL);
5      ini_set('display_errors', 'on');
6
7      spl_autoload_register(function($class) {
8          // ваша реализация автозагрузки
9      });
10 ?>
```

Задача 101.3

Реализуйте функцию автозагрузки. Проверьте ее работу.

