

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

урок 10 из 107

[Верстка](#) [JavaScript](#) [PHP](#) [NodeJs](#) [Vue](#) [React](#) [Laravel](#) [WordPress](#) [AJAX](#) [Парсинг](#)[Бесплатные курсы по React для новичков. Начало 4-го ноября→](#) [Конкурс CSS картинок. Тема: Хэллоун. Призовой фонд: 100\\$. Подробности→](#)

Хранение классов в отдельных файлах

До этого урока мы писали наши классы в том же файле, где и вызывали их.

В реальной жизни классы обычно хранятся в отдельных файлах, причем каждый класс - в своем отдельном файле.

При этом существует соглашение о том, что файл с классом следует называть так же, как и сам класс.

Например, если у нас есть класс **User**, то хранить его мы будем в файле **User.php**.

Давайте посмотрим на практике. Сделаем файл **User.php** с классом **User**:

```
1  <?php
2      // Файл User.php
3
4      class User
5      {
6
7      }
8  ?>
```

Пусть теперь у нас есть файл **index.php**, в котором мы хотим воспользоваться нашим классом **User**.

Мы не можем в этом файле просто взять и создать объект класса **User** - это вызовет ошибку, так как PHP не сможет найти код этого класса:

```
1  <?php
2      // Файл index.php
3
4      $user = new User; // это вызовет ошибку
5  ?>
```

Для того, чтобы класс **User** был доступен в файле **index.php**, нужно подключить файл **User.php** с этим классом к нашему файлу **index.php**.

Это делается с помощью команды **require**. Она работает подобно команде **include**, но не вставляет текст подключаемого файла прямо в это место, а просто говорит PHP, чтобы он сделал доступными все переменные, функции и классы из подключаемого файла.

Как раз то, что нам надо:

```
1  <?php
2      // Файл index.php
3
4      require 'User.php'; // подключаем наш класс
5      $user = new User; // теперь ошибки не будет
6  ?>
```

На самом деле подключение классов через **require** не очень удобная вещь, так как, если вы попытаетесь сделать подключение файла в нескольких местах (такое часто нужно), то повторное подключение приведет к ошибке - PHP скажет, что такой класс с таким именем уже создан.

Более удобная практика - использовать команду **require_once**, которая подключает файл только если он не был подключен ранее:

```
1  <?php
2      // Файл index.php
3
4      require_once 'User.php'; // подключаем наш
                               класс
5  ?>
```

```
$ user = new User;
```

```
6 | ?>
```

Пока, возможно, вам не особо понятна принципиальная разница между **require** и **require_once**. Пока просто везде подключайте классы через **require_once**. В дальнейшем на практике вы поймете удобство именно этого варианта подключения.

В следующих уроках для простоты я не буду указывать, что классы хранятся в разных файлах, но при решении задач делайте именно так.

Задача 10.1

Сделайте класс **Employee**, в котором будут следующие свойства: **name** (имя), **surname** (фамилия), **patronymic** (отчество) и **salary** (зарплата).

Пусть этот класс хранится в отдельном файле.

Задача 10.2

Подключите к файлу **index.php** класс **Employee**, создайте два объекта этого класса с произвольными данными, выведите на экран сумму их зарплат.

