

Чтение файлов Microsoft Word с использованием языка программирования PHP

Ересь Артём Владимирович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье будет рассмотрен процесс чтения файлов Microsoft Word с применением функционала языка программирования PHP. Подробно выделены особенности и возможности пользователя. На примере наглядно продемонстрирована работа созданного скрипта для чтения.

Ключевые слова: PHP, файл, Microsoft Word, PHPWord

Reading the Microsoft Word files about use of the PHP programming language

Yeres Artem Vladimirovich

Sholom-Aleichem Priamursky State University

Student

Abstract

In this article process of reading the Microsoft Word files with use of functionality of the PHP programming language will be considered. Features and possibilities of the user are marked in detail out. On an example work of the created script for reading is visually shown.

Keywords: PHP, файл, Microsoft Word, PHPWord

Достаточно часто пользователи сталкиваются с ситуациями, когда необходимо срочно прочесть важный документ. Но иногда нет возможности воспользоваться стандартным программным обеспечением. В этом случае программисты могут воспользоваться средствами языка программирования PHP в частности - библиотека PHPWord.

Целью данной работы является рассмотрение процесса чтения файлов Microsoft Word с использованием языка программирования PHP.

Тема данной статьи актуальна в современных научных исследованиях. Авторы И.Ю. Большаков и Н.Н. Иванова разработали собственную систему для создания и заполнения файлов. В работе использована библиотека PHPWord, эффективно применяемая для этих целей [1]. Группа исследователей И.А. Гурин, Н.А. Спирин, В.В. Лавров и Н.И. Декун в своей статье подробно разобрали все возможные манипуляции с файлами Microsoft Word относительно создания приложений. Были реализованы основные методы работы с документами на основе шаблонов [2]. В.Г. Козич и Р.И.

Баженов создали систему для учета индивидуальных достижений студентов, работающую с использованием документов. Сервис автоматизировал множество процессов и облегчил работу по обработке информации [3]. Примером применения PHPWord служит следующий источник, поясняющий процессы автоматического заполнения заявлений [4]. В интернет-ресурсе рассказывается о принципах работы и методах применения библиотеки для обработки документов на языке PHP [5].

Вначале работы необходимо установить библиотеку PHPWord, используя github.com.

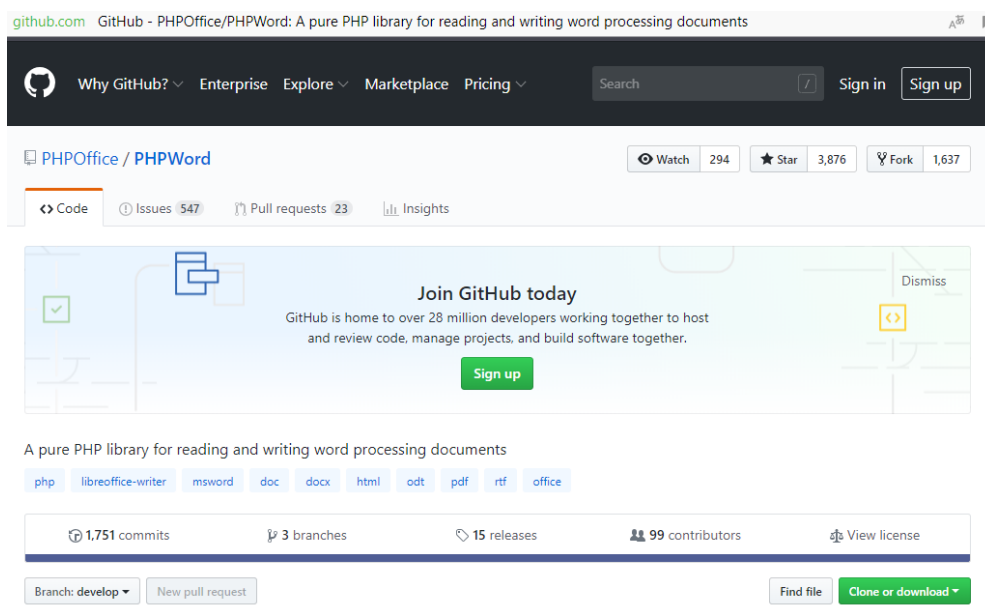


Рис. 1. Установка библиотеки

Далее следует создать файл для нашего примера и именно в него добавить PHPWord. Обозначим расположение документа и укажем его.

```
1 $source = __DIR__."/docs/text.docx";
```

Рис. 2. Расположение документа

Теперь необходимо прочитать наш файл для создания объектного класса. А эта процедура подразумевает под собой добавление специфического класса-чтения.

```
1 $objReader = \PhpOffice\PhpWord\IOFactory::createReader('Word2007');
```

Рис. 3. Создание класса-чтения

Производим чтение.

```
1 $phpWord = $objReader->load($source);
```

Рис. 4. Чтение документа

Итогом этих процессов можно считать успешное чтение документа и отправка информации с него в объект `php_Word`. Далее появляется задача открытия этих данных.

Стоит отметить, что информация библиотеки `PHPWord` хранится в секционном виде и состав каждой – это набор элементов.

Используем `get sections` для входа в секции файла:

```
1 foreach($phpWord->getSections() as $section) {
2   $arrays = $section->getElements();
3
4 }
```

Рис. 5. Вход в секции

Благодаря циклу мы можем увидеть внутренние элементы и перейти к отдельным ячейкам в более углублённый вид.

```
1 foreach($arrays as $e) {
2
3 }
```

Рис. 6. Просмотр ячеек

Теперь необходимо определить состав переменной «а». Для этого служит объект, который отмечает, что сложная структура составляется из мелких элементов

```
1 if(get_class($e) === 'PhpOffice\PhpWord\Element\TextRun') {
```

Рис. 7. Проверка переменной

Повторяем процедуру по элементам в цикле.

```
1 <?php
2
3 require 'vendor/autoload.php';
4
5 $source = __DIR__."/docs/text.docx";
6
7 $objReader = \PhpOffice\PhpWord\IOFactory::createReader('Word2007');
8
9 $phpWord = $objReader->load($source);
10
11
12 $body = '';
13 foreach($phpWord->getSections() as $section) {
14   $arrays = $section->getElements();
15
16   foreach($arrays as $e) {
17     if(get_class($e) === 'PhpOffice\PhpWord\Element\TextRun') {
18       foreach($e->getElements() as $text) {
19
20         $font = $text->getFontStyle();
21
22         $size = $font->getSize()/10;
23         $bold = $font->isBold() ? 'font-weight:700;' : '';
24         $color = $font->getColor();
25         $fontFamily = $font->getName();
26
27         $body .= '<span style="font-size:' . $size . 'em;font-family:' . $fontFamily . ';" . $bold.';
28         $body .= $text->getText().</span>';
29
30       }
31     }
32
33   }
34 }
35
36 include 'templ.php';
```

Рис.8. Цикл

В результате мы получаем элемент text, открывающийся методом get. Для получения более подробных параметров необходимо использовать следующие методы:

1. Size – шрифт
2. Color - цвет текста
3. Name – наименование шрифта.

В итоге состав файла отражается в body и на экране это отражается как шаблон для заполнения. Возможные чистые строки меняются с помощью следующего метода:

```
1 else if(get_class($e) === 'PhpOffice\PhpWord\Element\TextBreak') {  
2     $body .= '<br />';  
3 }
```

Рис. 9. Манипуляции со строками

Теперь после всех проведенных манипуляций по работе с библиотекой PHPWord попробуем наглядно проверить работу скрипта для чтения. Далее непосредственно взглянем на сам исходный документ Microsoft Word (рис. 10)

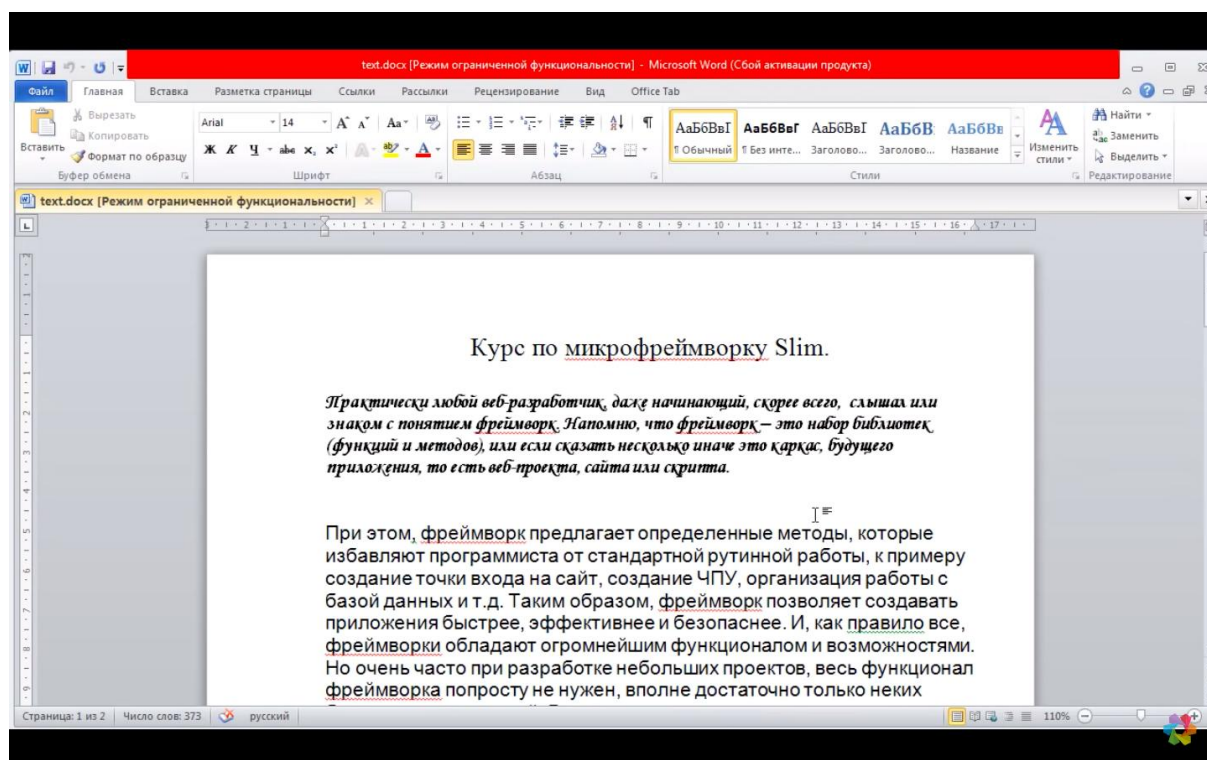


Рис. 10. Документ в программе Microsoft Word

Теперь запустим работу готового скрипта и проверим работу библиотеки PHPWord, которая должна отобразить содержимое DOCX документа на страницу браузера в HTML формате (рис. 11)

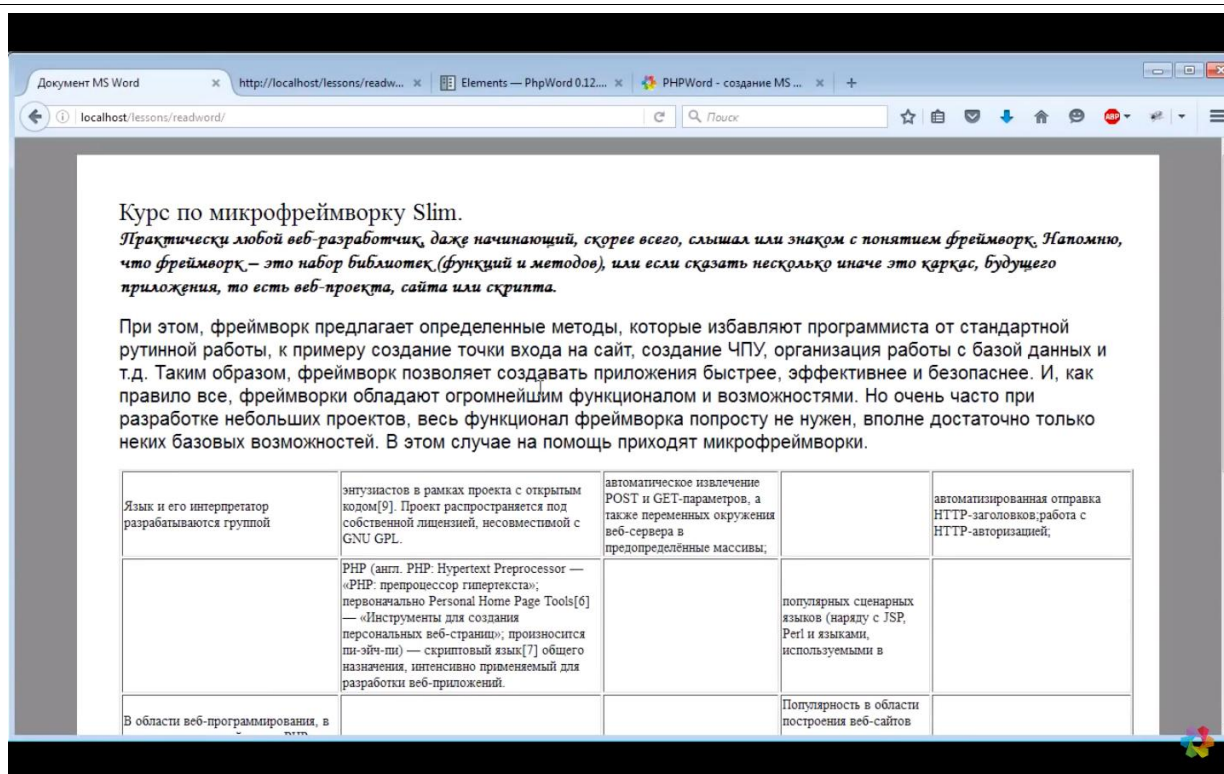


Рис. 11. Преобразованный документ в браузере

Таким образом, в работе был рассмотрен процесс чтения файлов Microsoft Word, с целью упрощения процесса заполнения документации.

Библиографический список

1. Большаков И.Ю., Иванова Н.Н. Разработка веб-приложения для автоматизации заполнения форм документов // Информатика и вычислительная техника. 2016. С. 42-46. URL: <https://elibrary.ru/item.asp?id=26901129> (Дата обращения: 14.01.2019)
2. Гурин И.А., Спиринов Н.А., Лавров В.В., Декун Н.И. Методы работы с документами Microsoft Word при разработке веб-сервисов // Теплотехника и информатика в образовании, науке и производстве. 2017. С. 199-203. URL: <https://elibrary.ru/item.asp?id=30540160> (Дата обращения: 14.01.2019)
3. Козич В.Г., Баженов Р.И. Разработка информационной системы «Портфолио научных достижений студентов» // Постулат. 2017. №9. С. 2. URL: <https://elibrary.ru/item.asp?id=30080373> (Дата обращения: 14.01.2019)
4. Заполнение шаблона договора docx с помощью PHPWord URL: <http://kumatoz.ru/programmirovanie/php/rabota-s-phpword-zapolnenie-shablona-dogovora-docx/> (Дата обращения: 14.01.2019)
5. PHPWord URL: <https://github.com/PHPOffice/PHPWord> (Дата обращения: 14.01.2019)