

PHPExcelによるExcelファイル作成

岩田 員典（愛知大学経営学部）

要旨

本学で開講している情報リテラシー入門・応用では、WordやExcelの自動採点システムを独自に構築・運用している。本システムではExcelの採点にPHPのライブラリであるPHPExcelを使用している。PHPExcelを利用するとPHPによってExcelファイルの読み込みや、新規Excelファイルの作成が可能になる。本システムでは採点にはExcelファイルを読み込む機能だけを利用している。それに加えて問題作成の労力を軽減するために、模範解答から解答欄を空欄にした問題を作成する機能も実装している。その際に問題ファイルにたいして、編集されては困るデータ部分は編集できないように一部のセルにロックを掛けている。前稿ではPHPExcelによるExcelファイルの読み込み方法についてのみ言及したので、本稿ではこれらセルへのロックなども行ったExcelファイル作成について説明する。

キーワード：Excel自動採点，PHPExcel，Excelファイル作成，情報リテラシー，PHP

1. はじめに

本学の情報リテラシー入門・応用では、学生がWordやExcelの多数の問題に取り組めるように、自動採点システムを独自に構築・運用している¹⁾。本システムのExcelの採点には、Office 2007以降はファイル形式がOffice Open XML²⁾に変更されたため、PHPのライブラリであるPHPExcel³⁾を用いている。そのため、前稿⁴⁾では採点時に必要となるPHPExcelを利用したExcelファイルの読み込みについて解説をした。しかし、採点システムではファイルを読み込むだけでなく、模範解答例から問題を作成する機能も実装している。その際に作成した問題

ファイルでは、学生がデータ部分を編集できないように一部のセルにロックをかけている。そこで本稿ではこれらの機能をPHPExcelを用いてどのように実現するかについて説明する。

2. PHPとは

正式な名称はPHP: Hypertext Preprocessor⁵⁾であり、通常PHPと省略される。PHPの部分は“Personal Home Page”に由来していると言われている。PHPは動的にHTMLデータを生成することによって、動的なWebページを実現することを主な目的としている。PHPといった場合にはプログラミング言語か

その言語プロセッサを指す。PHPはスクリプト言語であり、サーバサイドで動作するアプリケーションの開発に用いられる。通常は、Webサーバ上で動作し、サーバ上でPHPスクリプトの文書が閲覧要求されるたびに、そのスクリプトが実行される。そして、その結果をウェブブラウザに対して送信することで動的なWebページを実現している。C言語に似た平易な文法のため幅広く用いられている。また、本稿で取り上げるPHPExcelsのような多くのフレームワークやライブラリが存在し、様々なWebアプリケーションの開発が容易である。

尚、本稿執筆時において、PHPの最新のバージョンは7.1.9であるが、サンプルコードの動作確認は開発環境の関係で7.1.7で行っている。

2.1 PHPのインストール

PHPExcelsを利用するには、PHPをインストールする必要がある。

2.2.1 Windowsへのインストール

Window版のバイナリは、参考文献⁶⁾で示したURLからダウンロードすることが出来る。このサイトにはzipファイルが8種類存在する。そこから環境と用途に適切な物をダウンロードする。

- x86 Non Thread Safe
 - Zip
 - Debug Pack

- x86 Thread Safe
 - Zip
 - Debug Pack
- x64 Non Thread Safe
 - Zip
 - Debug Pack
- x64 Thread Safe
 - Zip
 - Debug Pack

x86かx64かはOSとCPUのビット数(32bitもしくは64bit)にあわせて選択する。そして、Non Thread SafeかThread SafeかはWebサーバの種類とVersionによる。Apache2以降ならばThread Safeを利用し、IISやApache1ならばNon Thread Safeを利用する。また、Debug Packはデバッグ用のプログラムデータベース含んだものである。これは統合開発環境でデバッグをする場合のみ必要となる。したがって、本稿で説明するPHPExcelsを利用するだけならビット(bit)数があるzipファイルさえあればよい。

ダウンロードしたzipファイルを好みの場所に展開する。展開したフォルダに存在するphp.exeをコマンドプロンプト等で実行できれば、そこにパスを通すことでインストールは完了する。尚、環境によっては実行に「Visual Studio 2012 更新プログラム4のVisual C++再頒布可能パッケージ」などのライブラリが必要となることがある。

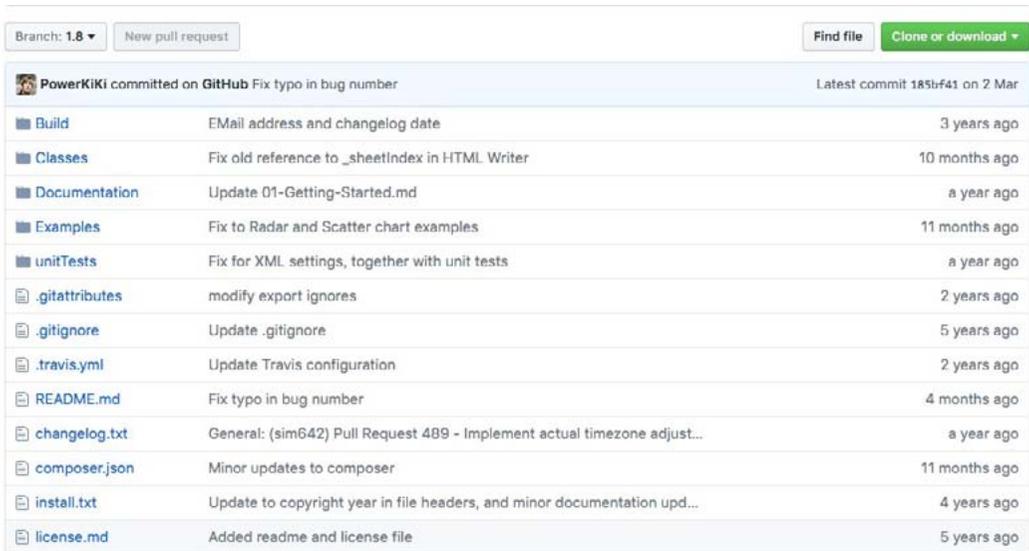


図1 PHPEXcelのページ

2.1.2 その他のOSへのインストール

Mac OS X 10.0.0以降では、PHPは標準でインストールされている。そのため、新しいバージョンを利用するのでなければターミナルなどからすぐに利用することが出来る。2017年9月の時点ではmacOS v10.13 (High Sierra)にはPHP 7.1.7がインストールされている。

その他のUnixシステムでは、OS毎のパッケージマネージャーを利用するかソースファイルをダウンロードしてインストールすることになる。詳細は参考文献⁵⁾のDocumentationに書かれている。

また、サーバサイドで利用する場合にはApache等のWebサーバの設定も必要であり、その際には同じく参考文献⁵⁾に設定方法が記載されている。

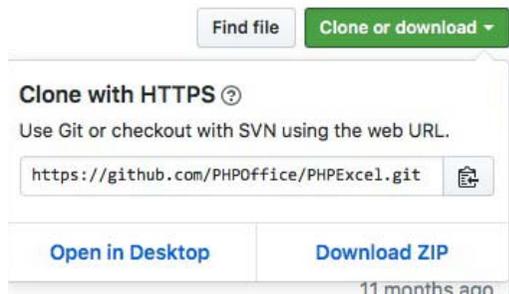


図2 PHPEXcelのダウンロード

3. PHPEXcelの利用方法

3.1 PHPEXcelのインストール

PHPEXcelは参考文献³⁾に示したURLにアクセスしダウンロードする(図1)。このサイトでClone or downloadをクリックし、Download ZIPを選択することで取得できる(図2)。本稿執筆時には

1.8.1 (2015/4/30リリース)が最新版である。

ダウンロードしたファイルはzipで圧縮されているため展開する。展開すると、Classes, Documentation, Examplesの3つのディレクトリと、changelog.txt, composer.json, install.txt, license.mdの4つのファイルが作成される。PHPExcelの実行に必要なファイルはClasses内のファイルのだけである。そこで、コードを作成するディレクトリにClassesディレクトリを移動する。ここでは、testPHPExcelと名付けたディレ

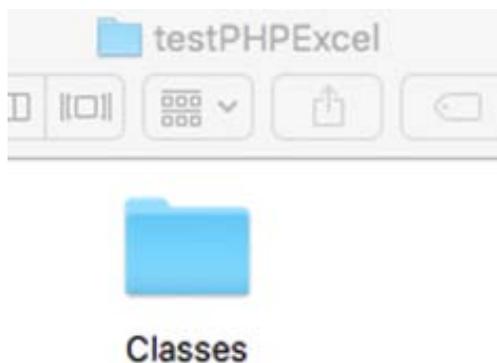


図3 PHPExcelのインストール

クトリで作業を行う(図3)。

3.2 Excelファイルの書き出し

本節ではExcelファイルをさまざまな設定で書き出すためのコードについて説明をする。全てのコードを一カ所に記載すると長いため分割して説明する。尚、PHPのコードは全てタグ“<?php”, “?”>”に囲まれているものとする。

3.2.1 書き出しの準備

図4にExcelファイルを書き出すための基本的な設定を示す。

1行目はライブラリを読み込むためのパスの設定を行う。Webアプリケーションとしてサーバサイドで利用する場合も、同様の設定を利用する。

2行目はTime Zoneの設定であり、PHPの設定状況によっては必要ない。

3~4行目が利用するライブラリの読み込みとなる。1行目で設定したパスに対しての相対的な参照となる。

5行目でExcelファイルを作成するた

```
1. set_include_path(get_include_path() . PATH_SEPARATOR . './Classes/');
2. date_default_timezone_set('Asia/Tokyo');
3. include_once("PHPExcel.php");
4. include_once("PHPExcel/IOFactory.php");
5. $objPHPExcel = new PHPExcel();
6. $objPHPExcel->getActiveSheet()->getDefaultStyle()->getFont()->setName('MS ゴシック');
7. $objPHPExcel->getActiveSheet()->getDefaultStyle()->getFont()->setSize(11);
```

図4 ファイル書き出しの基本設定

めのオブジェクトのインスタンスを新規作成している。

6行目, 7行目では, 使用するフォントの種類とフォントサイズを指定している。

3.2.2 ファイルプロパティの設定

Excelファイルのプロパティは図5のコードにより設定できる。

8行目でプロパティ内のタイトルを設定している。9行目では件名を指定して

```
8. $objPHPExcel->getProperties()->setTitle("Excelファイルの作成");
9. $objPHPExcel->getProperties()->setSubject("PHPExcel Programming");
10. $objPHPExcel->getProperties()->setCreator("岩田 員典");
11. $objPHPExcel->getProperties()->setManager("岩田 員典");
12. $objPHPExcel->getProperties()->setCompany("愛知大学");
13. $objPHPExcel->getProperties()->setCategory("PHPExcel");
14. $objPHPExcel->getProperties()->setKeywords("PHPExcel, Writer");
15. $objPHPExcel->getProperties()->setDescription("PHPExcelで作成");
```

図5 ファイルプロパティの設定

全般 概要 統計情報 コンテンツ ユーザー設定

タイトル: Excelファイルの作成

件名: PHPExcel Programming

作成者: 岩田 員典

管理者: 岩田 員典

会社名: 愛知大学

分類: PHPExcel

キーワード: PHPExcel, Writer

コメント: PHPExcelで作成

ハイパーリンクの基点:

テンプレート:

このドキュメントと一緒にプレビューの図を保存する

図6 ファイルプロパティの設定結果

いる。

10～12行目ではそれぞれ作成者、管理者、会社名を設定している。13行目で分類を指定している。また14行目・15行目のようにキーワードとコメントを記入することもできる。

このコードによるプロパティの設定結果を図6に示す。

3.2.3 シートの設定

図7では編集対象となるシートを、どのように指定するかについて記述している。16行目でデータを書き込みたいシートを指定している。シートの番号は先頭から0, 1, 2, …となる。17行目でそのシートを編集対象として取り出している。そして、18行目でシート名を設定している。

```
16. $objPHPExcel->setActiveSheetIndex(0);
17. $sheet = $objPHPExcel->getActiveSheet();
18. $sheet->setTitle("書き込みテスト");
```

図7 編集対象のシート指定とシート名の設定

```
19. $sheet->setCellValue('A1', 'PHPExcel');
20. $sheet->setCellValue('A2', '2');
21. $sheet->setCellValue('A3', '3');
22. $sheet->setCellValue('A4', '4');
23. $sheet->setCellValue('A5', '=sum(A2:A4);
```

図8 セルの値の設定

シートを追加する場合や、ファイルを開いたときに表示されるシートの指定については後述する。

3.2.4 セルへのデータ入力

セルに値を設定するには、setCellValueメソッドを利用する。第1引数にセル番号を'A1'のように指定する。そして、第2引数でそのセルに入力する値を指定する。例えばA2セルに2を入力したければsetCellValue('A2', '2');と記述する。また、第2引数には関数による計算を指定することもできる。

図8のコードを実行すると、A1には文字列が入力される。そして、A2からA4までのセルに数値が入り、その和を求める式がA5に入力される。

3.2.5 一部のセルの保護

授業では入力済みのデータに対して計算を行うように指示している。そのため、それらのデータを編集されてしまうと正しく採点できないという問題がおきる。そこで、問題ファイルの解答を入力するセル以外は編集できないように保護している。本節では、このようにセルの編集を制限する方法について説明する。

セルを保護するための基本的な方針は以下のような手順になる。

1. シート全体を保護する。
2. 編集可能にするセルを個別に指定する。

24行目の命令で、現在編集集中のシート全体を保護することになる（16行目の `setActiveSheetIndex(0)` で指定したシート）。そして、25行目で保護のためのパスワードを設定している。このパスワードを入力すれば、保護されているセルも編集可能になる。

26行目から29行目では、それぞれ行と

列の追加と削除を禁止するための命令である。行と列の削除に関しては、保護されたセルがある場合には実行できないため、ここでは必須というわけではない。

30行目はコードが長いいため紙面の都合上改行しているが、本来は一行で記述する。セルA2の保護を解除している。この命令を実行することで、セルA2だけ編集可能になる。

3.2.6 セルのスタイル設定

各セルのスタイルも設定することができる（図10）。背景色を設定する場合には31行目・32行目のように、`getFill()`で対象となるインスタンスを取り出してから設定する。それ以外のフォントの種類、サイズ、文字の色などは`getFont()`により取得したインスタンスに対して各種値を設定する。図10のコードでは以下の設定を行っている。

33行目：フォントの種類

34行目：太字にするかどうか

```
24. $sheet->getProtection()->setSheet(TRUE);
25. $sheet ->getProtection()->setPassword("Password");
26. $sheet ->getProtection()->setProtection()->setInsertRows(TRUE);
27. $sheet ->getProtection()->setProtection()->setDeleteRows(TRUE);
28. $sheet ->getProtection()->setProtection()->setInsertColumns(TRUE);
29. $sheet ->getProtection()->setProtection()->setDeleteColumns(TRUE);
30. $sheet->getStyle("A2")->getProtection()
    ->setLocked(PHPExcel_Style_Protection::PROTECTION_UNPROTECTED);
```

図9 セルの保護

TRUEの場合に太字になる
35行目：斜体にするかどうか
TRUEの場合に斜体になる
36行目：下線の設定（紙面の関係上途中で改行しているが、実際は一行）
37行目：フォントサイズ

38行目：文字の色
また、getFont()のインスタンスに対して指定できる設定をまとめて実行したい場合はapplyFromArray()メソッドを利用すると便利である。図10の33行目から38行目と同じ実行結果になるapplyFromArrayの使用例

```
31. $sheet->getStyle("A2")->getFill()->setFillType(PHPExcel_Style_Fill::FILL_SOLID);
32. $sheet->getStyle("A2")->getFill()->getStartColor()->setRGB('FFC16E');
33. $sheet->getStyle("A2")->getFont()->setName('MS ゴシック');
34. $sheet->getStyle("A2")->getFont()->setBold(TRUE);
35. $sheet->getStyle("A2")->getFont()->setItalic(TRUE);
36. $sheet->getStyle("A2")->getFont()->
    setUnderline(PHPExcel_Style_Font::UNDERLINE_SINGLE);
37. $sheet->getStyle("A2")->getFont()->setSize(20);
38. $sheet->getStyle("A2")->getFont()->getColor()->setRGB('043BB1');
```

図10 セルのスタイル設定

```
$sheet->getStyle("A2")->getFont()->applyFromArray(
    array(
        'name' => 'MS ゴシック',
        'bold' => TRUE,
        'italic' => FALSE,
        'underline' => PHPExcel_Style_Font::UNDERLINE_DOUBLE,
        'strike' => FALSE,
        'size' => 20,
        'color' => array(
            'rgb' => '0000FF'
        )
    )
);
```

図11 applyFromArray の使用例

```
39. $objPHPExcel->createSheet();
40. $objPHPExcel->setActiveSheetIndex(1);
41. $sheet = $objPHPExcel->getActiveSheet();
42. $sheet->setTitle("シート2");
43. $sheet->setCellValue('A1', '2つめのシート');
44. $objPHPExcel->setActiveSheetIndex(0);
```

図12 シートの追加

```
45. $writer = PHPExcel_IOFactory::createWriter($objPHPExcel, 'Excel2007');
46. $writer->save('./writeTest.xlsx');
```

図13 ファイルへの出力

を図11に示す。図11で示したように `applyFromArray()` の引数に連想配列で設定を渡す。この配列のキーに設定名を、値に設定する値を記載する。

3.2.7 シートの追加

シートを追加するためのコードを図12に示す。39行目で新たなシートを作成している。40行目でそのシートを編集対象にして、41行目でシート情報を取り出している。42行目でシート名を、43行目でこのシートのA1セルにデータを設定している。尚、44行目はファイルを開いたときに1枚目のシートを表示するために加えている。この行が無い場合は、ファイルを開いたときに2枚目のシートが表示される。

3.2.8 ファイルへの出力

最後にファイルへの出力方法について示す。まず、図13の45行目のように、編集を行ったオブジェクト（`$objPHPExcel`）をExcel2007形式で出力するためのwriterを作成する。xls（Excel2003などの形式）で出力したい場合はExcel5を指定する。46行目で実際に出力している。saveメソッドにファイル名（パス）を与えることで、その名前でも保存される。

3.3 実行例

これまでコードを実行した結果得られるExcelファイルを図14と図15に示す。図14はファイルを開いたときの表示でもある。先頭のシートのセルA1～A5はそれぞれ図8と図10で指定した内容が表示

	A	B
1	PHPExcel	
2	2	
3		3
4		4
5		9
6		
7		

図14 先頭のシート

	A	B
1	2つめのシート	
2		
3		
4		
5		
6		
7		
8		

図15 2枚目シート

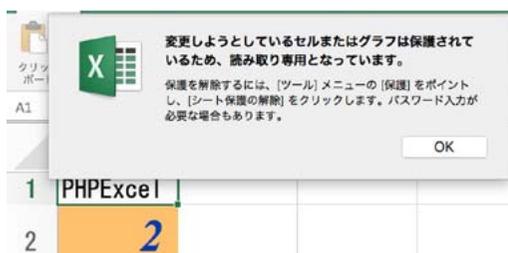


図16 セルA1の編集

されている。また、2枚目のシートは、図12で記述した内容通りに作成されている。

さらに先頭のシートはセルA2以外はロックされていて値の変更等ができないようになっている。図16にセルA1の編集を試みた場合の表示について示す。

4. まとめ

本稿では前稿⁴⁾では述べられなかったPHPExcelを利用したExcelファイルの

作成方法について述べた。また、PHPやPHPExcelの最新バージョンにあわせたインストールの仕方も示した。そして、ファイル作成のためのサンプルコードを記載し、その実行例を示した。

前稿・本稿ではPHPExcelをコマンドラインで利用についてのみ示した。しかし、PHPはWebアプリケーションとしてサーバサイドで実行することが多いため、次稿ではその設定等についても述べていきたい。

参考文献

- 1) 岩田員典, 松井吉光, 長谷部勝也, 谷口正明, 池森均, 梅垣敦紀, 齋藤毅, 澤田貴行, 土橋喜, 中尾浩, 西本寛, 古川邦之, 毛利元昭: 情報リテラシーのための Word, Excel自動採点システムの構築と運用, 教育改革ICT戦略大会 pp.294-295 (2013)。

- 2) Standard ECMA-376 Office Open XML File Formats, <http://www.ecma-international.org/publications/standards/Ecma-376.htm> (2006-2012)。
- 3) PHPExcel, <https://github.com/PHPOffice/PHPExcel>
- 4) 岩田員典：PHPExcelによるExcelファイルの読み込み, 愛知大学 情報メディアセンター紀要「COM」, Vol.26, No.1第41号, pp.73-79 (2016)。
- 5) PHP, <http://php.net>
- 6) PHP For Windows, <http://windows.php.net/download>

