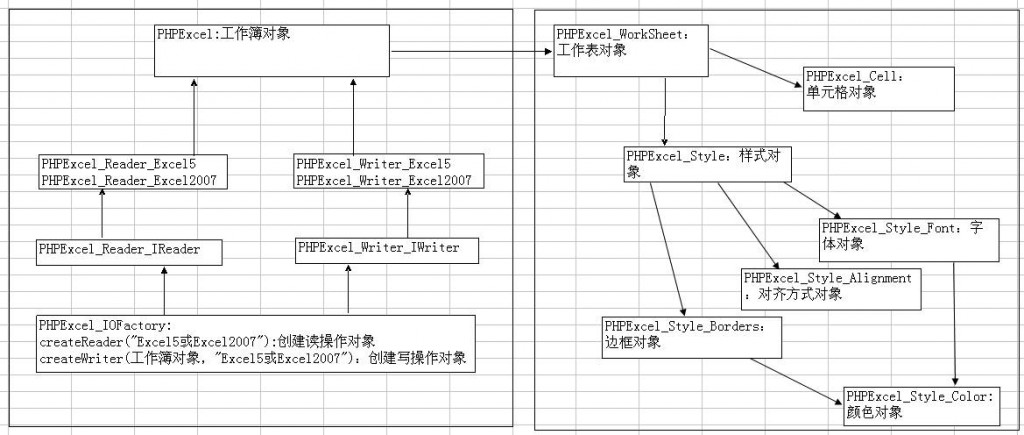
# PHPExcel使用说明

PHPExcel是PHP操作Excel文档的方法,能实现Excel文件的生成，读取。

# PHPExcel工作流程图



# PHPExcel加载文档

## 1.引入读写类。

* 1. require\_once ‘/Classes/PHPExcel/IOFactory.php’;

----->创建读取对象。

//创建读取对象，需要填写加载的文档类型，

//Excel5是后缀为xls的文件，Excel2007是后缀为xlsx的文件

$objReader = PHPExcel\_IOFactory::createReader();

$objWrite = PHPExcel\_IOFactory::createWriter();

B) 直接调用IOFactory.php中的load方法

------->创建对象

//load方法中会自动判断文件的类型（主要根据文件的后缀名进行判断，详见源码）

$objExcel = PHPExcel\_IOFactory::load($name);

## 2.引入PHPExcel初始化类库文件。

a) require\_once '/Classes/PHPExcel.php';

----->创建对象。

$objPHPExcel = new PHPExcel();

## 3.引入PHPExcel初始化类库文件错误。

因为PHPExcel自己实现的自动加载，有一定概率与框架中的自动加载冲突。当出现大量文件无法引入（确保不是路径问题），需要修改该文件的自动加载类。

//修改PHPExcel/Autoloader.php文件中的Register方法

$functions = spl\_autoload\_functions();

foreach ( $functions as $function){

spl\_autoload\_unregister($function);

}

$functions = array\_merge(array(array('PHPExcel\_Autoloader','Load')),$functions);

foreach ( $functions as $function){

$x = spl\_autoload\_register($function);

}

return $x;

## 4.初始化PHPExcel类与初始化IOFactory类的区别

PHPExcel类中没有load() 方法

IOFactory中有load()方法。

## 5.创建类的方法区别

PHPExcel\_IOFactory::\*\* （依次向下调用）

|->Load(文件路径) 没有setReadDataOnly(true)方法

|->createReaderForFile(文件路径) 此方法可以自动确认文件类型

|->createReader(文件类型)

# 方法索引

## PHPExcel：工作簿对象

1. getActiveSheet()：获得当前活动状态的工作表，返回工作表对象。
2. setActiveSheetIndex()：设置当前活动状态工作表的索引，返回工作表对象。
3. getSheetByName()：通过工作表名称得到当前工作表对象，返回工作表对象。
4. getDefaultStyle()：获得excel文档默认的样式（设置默认样式），返回样式对象。

------------工作表对象（以上）--------------------------

1. getActiveSheetIndex()：获得当前活动状态工作表的索引值，返回int。
2. getSheetCount()：获得excel文档中工作表的数量，返回int。
3. getSheetNames()：获得excel文档中所有工作表名称组成的数组。

-----------其他对象（以下）-------------

1. createSheet():在当前活动工作表后创建一个新的工作表。
2. getProperties()：获得当前活动状态工作表的属性对象，返回属性对象。

## PHPExcel\_Worksheet：工作表对象（需要获取工作簿对象）

1. getCell(“坐标”)：获得单元格对象

-->getValue(): 获取单元格的值

1. getStyle():获得指定单元格的样式，返回样式对象
2. getDefaultStyle()：获得工作表默认的样式，返回样式对象
3. getCellByColumnAndRow()：通过列索引和行索引获得指定单元格，返回单元格对象
4. getColumnDimension():获得当前列
5. getHighestColumn()：获得工作表的最大列，返回列的名称
6. getParent()：获得父类对象，返回excel文档对象

---------得到列，行的个数

1. getHighestRow():得到有几行（1,2,3,4..）
2. getHighestColumn():得到有多少列（A，B，C.....）

---------对象

1. getRowDimension()：获取行（1,2,3）
2. getColumnDimension(): 获取列（A，B，C）
3. getDefaultColumnDimension()
4. getDefaultRowDimension()

---------

1. fromArray():从数组中获取数据填充到工作表，返回工作表对象
2. toArray()：把工作表中的数据转换成数组

----------

1. setCellValue():设置单元格的值，返回工作表对象或单元格对象，完全取决于参数的值
2. setCellValueExplicit()：设置单元格的值，并显示指定数据类型，返回工作表对象
3. setCellValueByColumnAndRow()：通过列索引和行索引设置单元格的值，
4. setCellValueExplicitByColumnAndRow()：通过列和行索引设置单元格值
5. setTitle():设置工作表标题

----------

1. getTitle()：获得工作表的标题或名称，返回字符串类型

## PHPExcel\_Style：样式对象(需要工作表对象)

1. getActiveCell()：获得当前活动的单元格的名称，返回string；如，A1。
2. getActiveSheet()：获得当前活动的工作表，返回工作表对象。
3. getAlignment()：获得对齐方式对象，返回对齐方式对象。
4. getBorders()：获得边框对象，返回边框对象。
5. getFill()：获得填充对象。
6. getFont()：获得字体对象。
7. setFont()：设置字体，返回样式对象。

## PHPExcel\_Style\_Alignment：对齐方式对象(需要样式对象)

1. getHorizontal()：获得水平居中方式。
2. getVertical()：获得垂直居中方式。
3. setHorizontal()：设置水平居中方式，返回对齐方式对象。
4. setVertical()：设置垂直居中方式，返回对齐方式对象。

HORIZONTAL\_CENTER（水平居中）  
HORIZONTAL\_CENTER\_CONTINUOUS  
HORIZONTAL\_GENERAL   
HORIZONTAL\_JUSTIFY   
HORIZONTAL\_LEFT（水平靠左）  
HORIZONTAL\_RIGHT（水平靠右）  
VERTICAL\_BOTTOM（垂直靠低）  
VERTICAL\_CENTER（垂直居中）  
VERTICAL\_JUSTIFY  
VERTICAL\_TOP（垂直靠顶部）

## PHPExcel\_Style\_Font：字体对象(需要样式对象)

1. setBold（）：设置字体加粗。
2. setColor（）：设置字体颜色。

与setColor等效：getColor()->setARGB(PHPExcel\_Style\_Color::COLOR\_WHITE)

1. setItalic（）：设置字体倾斜。
2. setName（）：设置字体名。
3. setSize（）：设置字体大小。
4. setUnderline（）：设置字体下划线。

## 背景填充

//填充类型

『样式对象』->getFill()->setFillType(PHPExcel\_Style\_Fill::FILL\_SOLID);

//填充颜色

『样式对象』->getFill()->getStartColor()->setARGB('00558bb6');

## 设置间距

『工作表对象』->getRowDimension("2")->setRowHeight('15');

『工作表对象』->getDefaultColumnDimension()->setWidth('30');

## 设置样式类

//样式数组

$styleBorderTop = array(

"borders" => array(

"right" => array(

"style" => PHPExcel\_Style\_Border::BORDER\_THIN,

"color" => array("rgb" => "000000"),

),

"bottom" => array(

"style" => PHPExcel\_Style\_Border::BORDER\_THIN,

"color" => array("rgb" => "000000"),

),

"left"=>array(

"style" => PHPExcel\_Style\_Border::BORDER\_THIN,

"color" => array("rgb" => "000000"),

)

),

);

//$styleBorderButton样式的集合

『工作表对象』->getStyle("A1:F1")->applyFromArray($styleBorderButton);

# 保存方法

/\*\*

\* $objPHPExcel 工作对象；

\* Excel2007 保存的文件格式

\* $patchExcel 保存的文件路径

\* $name 文件名

\*/

$objWriter = PHPExcel\_IOFactory::createWriter($objPHPExcel, 'Excel2007');

$objWriter->save($patchExcel . $name);