# PHP EXCEL

@iliaa

Slides : http://ilia.ws/

# Me, Myself & I

* PHP Core Developer since 2001

* Author of Excel extension ;-)

# The Need…

# Ability to parse and generate Excel documents

# PRIOR TO PHP-EXCEL

# PEAR ExcelReader/Writer - Old & Abandoned...

# PHPExcel - 30kb per cell & really, really slow…

# HENCE PHP-EXCEL

# Based around LibXL library

http://libxl.com/

**Costs $199**

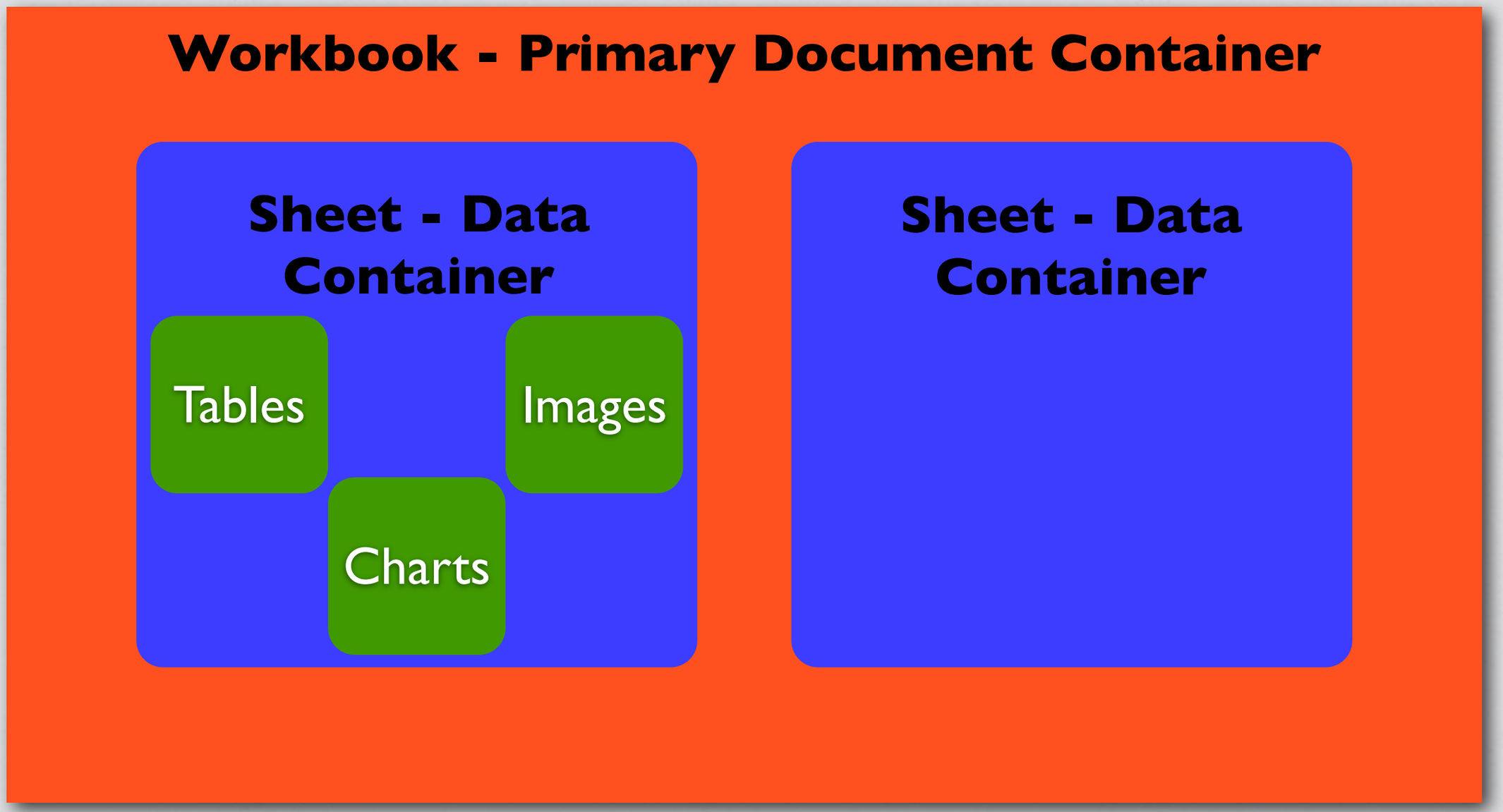
# PHP Interface is Open Source available at:

**https://github.com/iliaal/php_excel**

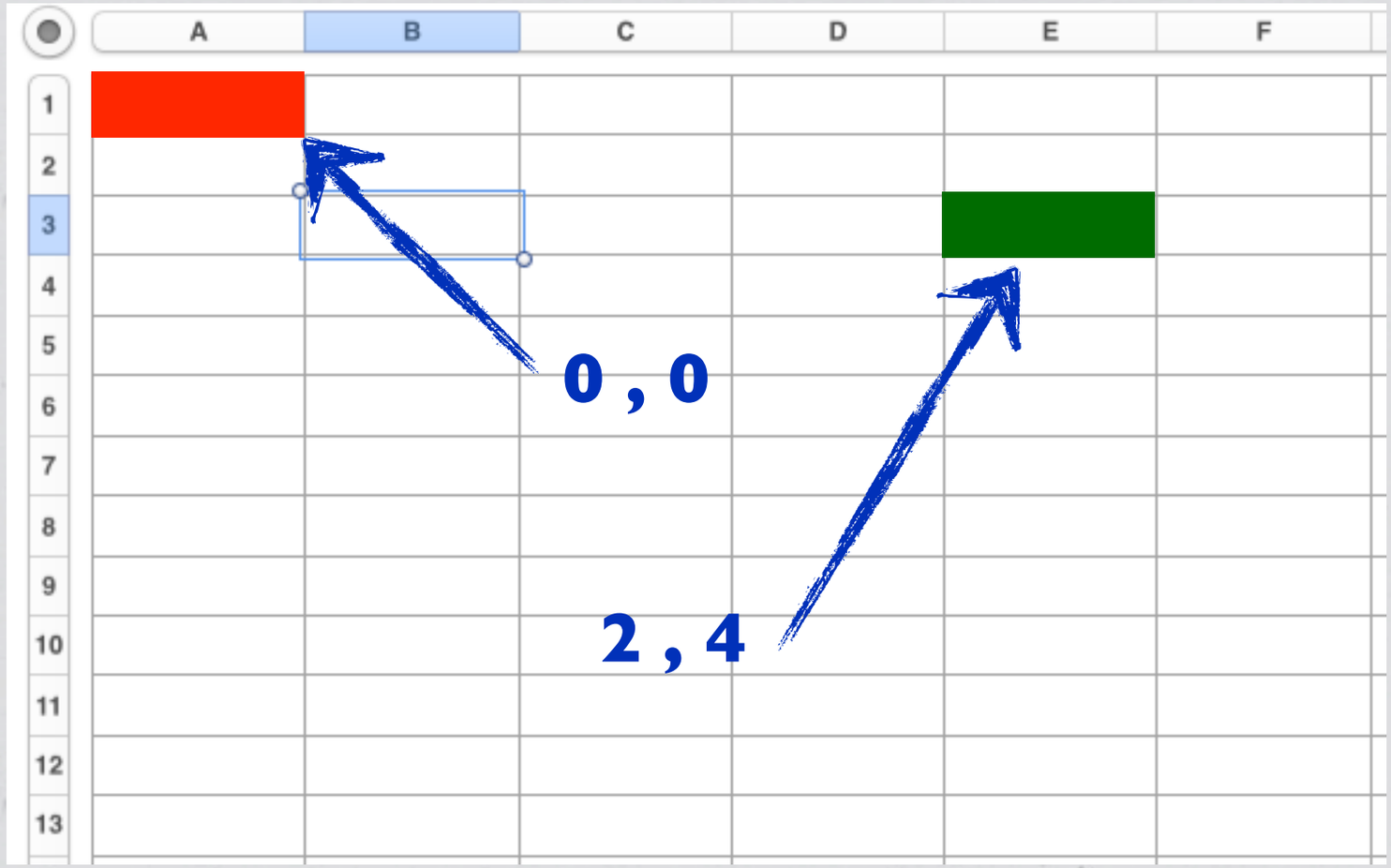
# Core Features

* Can create Biff-8 and XML based documents
* Can parse Biff 5 through 8 and XML based documents
* Really, really fast... 500,000 cells per second fast
* Does not eat a ton of memory
* Simple OO Interface
* Lots of features!

# Excel Document Structure



# Table Addressing




# Rows 1st, Columns 2nd

# CODE TIME!

# Simple Document Creation

```
$x = new ExcelBook();

$s = $x->addSheet("Sheet 1");
$s->write(1, 1, 'Test');
$s->write(2, 2, 123);

$x->save("file.xls");
```

| | A | B | C |
|---|---|---|---|
| 1 | | | |
| 2 | | Test | |
| 3 | | | 123 |
| 4 | | | |
| 5 | | | |

# XML Document Handling

```
$x = new ExcelBook(NULL, NULL, TRUE);

$s = $x->addSheet("Sheet 1");
$s->write(1, 1, 'Test');
$s->write(2, 2, 123);

$x->save("file.xlsx");
```

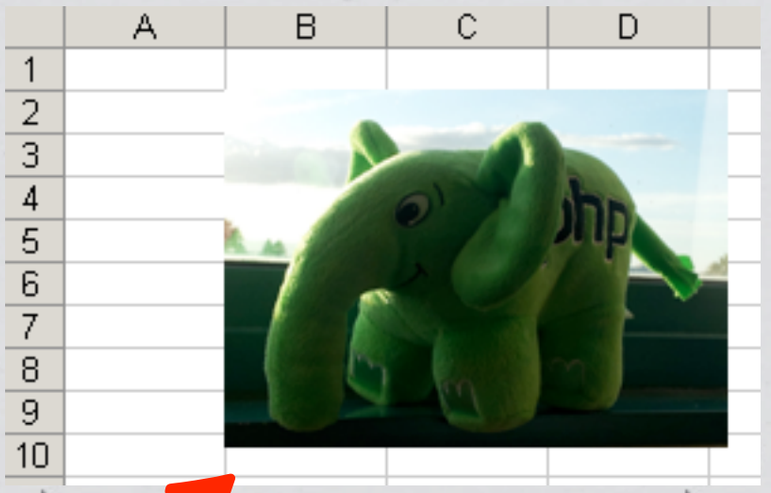
Fancy Office 2010
XML format

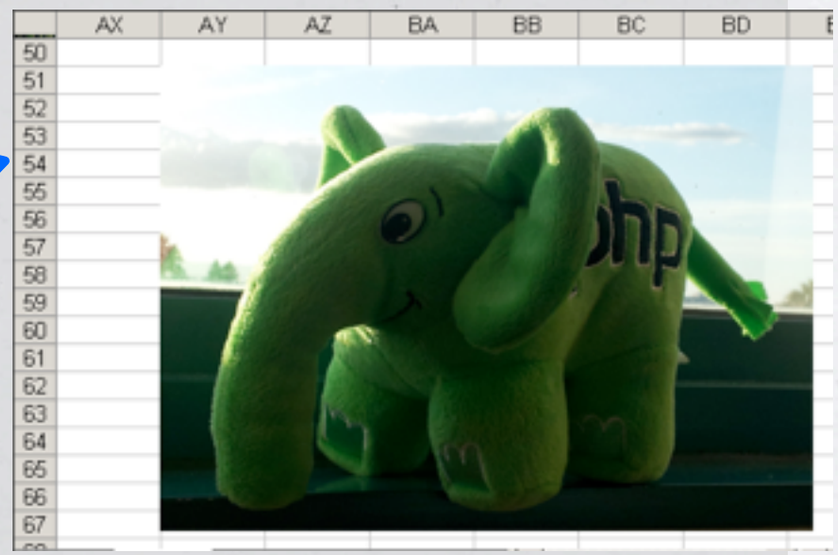
# Writing an Entire Row @ Once

```php
$db = new PDO('database');

$x = new ExcelBook;
$s = $x->addSheet("Weekly TPS Report");

$i = 0;
foreach ($db->query('SELECT * FROM tps', PDO::FETCH_NUM) as $row) {
    $s->writeRow($i++, $row);
}

$x->save("file.xls");
```

# Picture Time!

```php
$x = new ExcelBook();
$s = $x->addSheet("Sheet 1");

$pic_file = "./theGreenOne.jpg";

// Loads picture into memory
$pic = $x->addPictureFromFile($pic_file);

// Place picture on 2nd row, 50% scale
$s->addPictureScaled(1, 1, $pic, 0.5);

// Place picture with specific dimensions
$s->addPictureDim(50, 50, $pic, 400, 285);

$x->save("e.xls");
```



# LOTS OF FORMATTING OPTIONS

# Borders

```php
$x = new ExcelBook();
$s = $x->addSheet("Borders");

$oClass = new ReflectionClass('ExcelFormat');
foreach ($oClass->getConstants() as $c => $val) {
    if (strpos($c, 'BORDERSTYLE_') !== 0) {
        continue;
    }
    $format = $x->addFormat();
    $format->borderStyle($val);

    $name = preg_replace('!.*_!', '', $c);

    $s->write($row++, $col++, $name, $format);
}

$x->save("borders.xls");
```

# Output

NONE

THIN

MEDIUM

DASHED

DOTTED

THICK

DOUBLE

HAIR

MEDIUMDASHED

DASHDOT

MEDIUMDASHDOT

DASHDOTDOT

MEDIUMDASHDOTDOT

SLANTDASHDOT

# Border Colors

```
$x = new ExcelBook();
$s = $x->addSheet("Border Colors");

$oClass = new ReflectionClass('ExcelFormat');
foreach ($oClass->getConstants() as $c => $val) {
    if (strpos($c, 'COLOR_') !== 0) {
        continue;
    }
    $format = $x->addFormat();
    $format->borderColor($val);
    $format->borderStyle(ExcelFormat::BORDERSTYLE_THICK);

    $name = preg_replace('!.*_!U', '', $c);

    $s->write($row++, $col++, $name, $format);
}

$x->save("borders.xls");
```

# Output

BLACK
WHITE
RED
BRIGHTGREEN
BLUE
YELLOW
PINK
TURQUOISE
BLUE_CL
SKYBLUE
LIGHTTURQUOISE
LIGHTGREEN
LIGHTYELLOW
PALEBLUE
ROSE
LAVENDER
TAN
LIGHTBLUE
GRAY50
PERIWINKLE_CF
PLUM_CF
IVORY_CF
LIGHTTURQUOISE_CF
DARKPURPLE_CF
CORAL_CF
OCEANBLUE_CF
ICEBLUE_CF
DARKBLUE_CL
PINK_CL
YELLOW_CL
TURQUOISE_CL
VIOLET_CL
DARKRED_CL
TEAL_CL

# Specific Border Functions

BorderLeftStyle

BorderLeftColor

BorderRightStyle

BorderRightColor

BorderTopStyle

BorderTopColor

BorderBottomStyle

BorderBottomColor

# Text Styles

```
$x = new ExcelBook();
$s = $x->addSheet("Text Stuff...");

$font_styles = array('Arial', 'Helvetica', 'Courier',
                     'Times New Roman', 'Tahoma', 'Courier New');

foreach ($font_styles as $style) {
    $font = $x->addFont();

    $font->name($style);

    $format = $x->addFormat();
    $format->setFont($font);

    $s->write($row, $col++, $style, $format);
}
$x->save("font.xls");
```

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | Arial | Helvetica | Courier | Times New Roman | Tahoma | Courier New | |

# Text Styles

```php
$x = new ExcelBook();
$s = $x->addSheet("Text Stuff...");

$underline_types = array(
    ExcelFont::UNDERLINE_NONE, ExcelFont::UNDERLINE_SINGLE,
    ExcelFont::UNDERLINE_DOUBLE, ExcelFont::UNDERLINE_SINGLEACC,
    ExcelFont::UNDERLINE_DOUBLEACC
);

foreach ($underline_types as $style) {
    $font = $x->addFont();
    $font->underline($style);

    $format = $x->addFormat();
    $format->setFont($font);

    $s->write($row, $col++, $style, $format);
}
$x->save("font.xls");
```

| Underline value | View |
|---|---|
| UNDERLINE_NONE | UNDERLINE_NONE |
| UNDERLINE_SINGLE | UNDERLINE_SINGLE |
| UNDERLINE_DOUBLE | UNDERLINE_DOUBLE |
| UNDERLINE_SINGLEACC | UNDERLINE_SINGLEACC |
| UNDERLINE_DOUBLEACC | UNDERLINE_DOUBLEACC |

# More Text Stuff...

```php
$x = new ExcelBook();
$s = $x->addSheet("Text Stuff...");

foreach (array('italics', 'bold', 'strike') as $style) {
    $font = $x->addFont();
    $font->$style(true);

    $format = $x->addFormat();
    $format->setFont($font);

    $s->write($row, $col++, $style, $format);
}
$x->save("font.xls");
```

| | A | B | C |
|---|---|---|---|
| 1 | *italics* | **bold** | ~~strike~~ |

# There is more still…

```
$x = new ExcelBook();
$s = $x->addSheet("Text Stuff...");

foreach (array('NORMAL', 'SUPERSCRIPT','SUBSCRIPT')
as $style) {
    $font = $x->addFont();
    $font->mode(constant("ExcelFont::" . $style));

    $format = $x->addFormat();
    $format->setFont($font);

    $s->write($row, $col++, $style, $format);
}
$x->save("font.xls");
```

| | A | B | C |
|---|---|---|---|
| 1 | NORMAL | SUPERSCRIPT | SUBSCRIPT |
| 2 | | | |

```php
$x = new ExcelBook();
$s = $x->addSheet("Text Stuff...");

$colors = array(
    ExcelFormat::COLOR_BROWN, ExcelFormat::COLOR_RED,
    ExcelFormat::COLOR_BRIGHTGREEN, ExcelFormat::COLOR_ICEBLUE_CF,
);

for ($i = 0; $i < 4; $i++) {
    $font = $x->addFont();
    // valid range is from 0 > 40
    $font->size($i * $i);

    $font->color(array_pop($colors));

    $format = $x-axes>addFormat();
    $format->setFont($font);

    $s->write($row, $col++, "ABC 123", $format);
}
$x->save("font.xls");
```

| | A | B | C | D |
|---|---|---|---|---|
| 1 | ABC 123 | | ABC 123 | ABC 123 |

# Date Formatting

```
$x = new ExcelBook();
$s = $x->addSheet("Sheet 1");
$time = mktime(5,43,11,1,21,1980);

$dc = array(
        "NUMFORMAT_DATE",
        "NUMFORMAT_CUSTOM_D_MON_YY",
        "NUMFORMAT_CUSTOM_D_MON",
        "NUMFORMAT_CUSTOM_MON_YY",
        "NUMFORMAT_CUSTOM_HMM_AM",
        "NUMFORMAT_CUSTOM_HMMSS_AM",
        "NUMFORMAT_CUSTOM_HMM",
        "NUMFORMAT_CUSTOM_HMMSS",
        "NUMFORMAT_CUSTOM_MDYYYY_HMM",
        "NUMFORMAT_CUSTOM_MMSS",
        "NUMFORMAT_CUSTOM_H0MMSS",
        "NUMFORMAT_CUSTOM_MMSS0"
    );
```

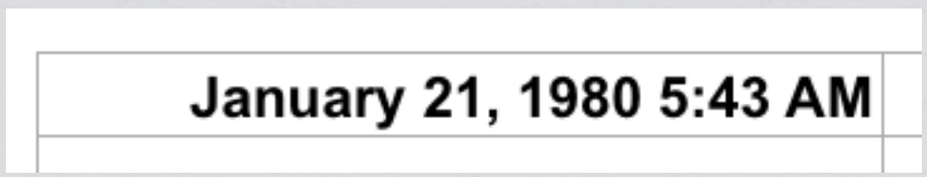
# Date Formatting

```
$row = 0;
foreach ($dc as $v) {
    $fmt = constant('ExcelFormat::' . $v);

    $format = $x->addFormat();
    $format->numberFormat($fmt);

    $s->write($row, 0, $v);
    $s->write($row++, 1, $time, $format, ExcelFormat::AS_DATE);
}
```

| | |
|---|---|
| NUMFORMAT_DATE | 1/21/80 |
| NUMFORMAT_CUSTOM_D_MON_YY | 21-Jan-80 |
| NUMFORMAT_CUSTOM_D_MON | 21-Jan |
| NUMFORMAT_CUSTOM_MON_YY | Jan-80 |
| NUMFORMAT_CUSTOM_HMM_AM | 5:43 AM |
| NUMFORMAT_CUSTOM_HMMSS_AM | 5:43:11 AM |
| NUMFORMAT_CUSTOM_HMM | 5:43 |
| NUMFORMAT_CUSTOM_HMMSS | 5:43:11 |
| NUMFORMAT_CUSTOM_MDYYYY_HMM | 1/21/80 5:43 |
| NUMFORMAT_CUSTOM_MMSS | 43:11 |
| NUMFORMAT_CUSTOM_H0MMSS | 701789:43:11 |
| NUMFORMAT_CUSTOM_MMSS0 | 43:11.0 |

# Custom Date Formatting

```
$x = new ExcelBook();
$s = $x->addSheet("Sheet 1");
$time = mktime(5,43,11,1,21,1980);

// Custom format guidelines
// http://libxl.com/custom-format.html
$myfmt = $x->addCustomFormat("mmmm dd, yyyy h:mm AM/PM");

$df = $x->addFormat();
$df->numberFormat($myfmt);

$s->write(0, 0, $time, $df, ExcelFormat::AS_DATE);

$x->save("test.xls");
```

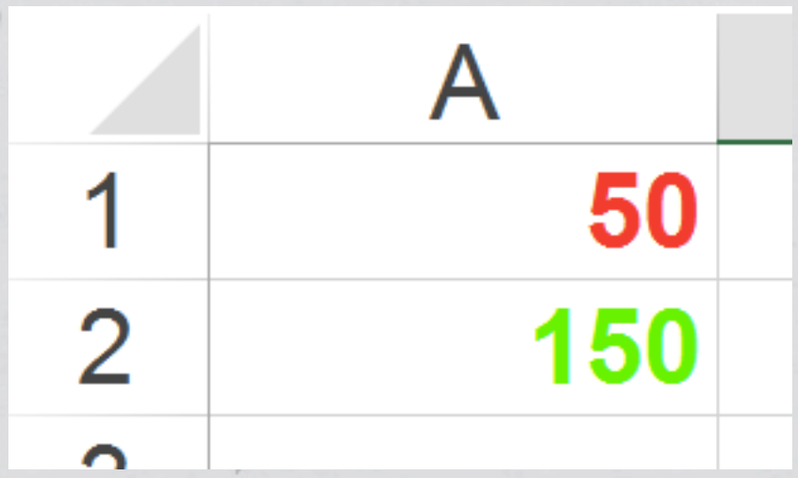


# Number Formatting

| Constant | Description | Example |
|---|---|---|
| **NUMFORMAT_NUMBER** | general number | 1000 |
| **NUMFORMAT_NUMBER_D2** | number with decimal point | 1000 |
| **NUMFORMAT_NUMBER_SEP** | number with thousands separator | 100,000 |
| **NUMFORMAT_NUMBER_SEP_D2** | number with decimal point and thousands separator | 100,000 |
| **NUMFORMAT_CURRENCY_NEGBRA** | monetary value, negative in brackets | -$1000 |
| **NUMFORMAT_CURRENCY_NEGBRARED** | monetary value, negative is red in brackets | -$1000 |
| **NUMFORMAT_CURRENCY_D2_NEGBRA** | monetary value with decimal point, negative in brackets | -$1000 |
| **NUMFORMAT_CURRENCY_D2_NEGBRARED** | monetary value with decimal point, negative is red in brackets | -$1000 |
| **NUMFORMAT_PERCENT** | percent value, multiply the cell value by 100 | 75% |
| **NUMFORMAT_PERCENT_D2** | percent value with decimal point, multiply the cell value by 100 | 75% |
| **NUMFORMAT_SCIENTIFIC_D2** | scientific value with E character and decimal point | 1.00E+02 |
| **NUMFORMAT_FRACTION_ONEDIG** | fraction value, one digit | 10 1/2 |
| **NUMFORMAT_FRACTION_TWODIG** | fraction value, two digits | 10 23/95 |
| **NUMFORMAT_NUMBER_SEP_NEGBRA** | number with thousands separator, negative in brackets | -4,000 |
| **NUMFORMAT_NUMBER_SEP_NEGBRARED** | number with thousands separator, negative is red in brackets | -4,000 |
| **NUMFORMAT_NUMBER_D2_SEP_NEGBRA** | number with thousands separator and decimal point, negative in brackets | -4,000 |
| **NUMFORMAT_NUMBER_D2_SEP_NEGBRARED** | number with thousands separator and decimal point, negative is red in brackets | -4,000 |
| **NUMFORMAT_ACCOUNT** | account value | 5,000 |
| **NUMFORMAT_ACCOUNTCUR** | account value with currency symbol | $ 5000.00 |
| **NUMFORMAT_ACCOUNT_D2** | account value with decimal point | 5,000 |
| **NUMFORMAT_ACCOUNT_D2_CUR** | account value with currency symbol and decimal point | $ 5,000.00 |
| **NUMFORMAT_TEXT** | text value | any text |

# Custom Number Formatting

```
$x = new ExcelBook();
$s = $x->addSheet("Sheet 1");

// Custom format guide lines
// http://libxl.com/custom-format.html
$myfmt = $x->addCustomFormat("[Red][<=100];[Green][>100]");

$nf = $x->addFormat();
$nf->numberFormat($myfmt);

$s->write(0, 0, 50, $nf);
$s->write(0, 1, 150, $nf);

$x->save("test.xls");
```



# Parsing Documents

```
$x = new ExcelBook();

$x->loadFile("file.xls");

$s = $x->getSheet();

for ($i = 0, $e = $s->lastRow(); $i < $e; $i++) {
    print_r(array_filter($s->readRow($i)));
}
```

# Precision Parsing

```php
$x = new ExcelBook();
$x->loadFile("file.xls");
$s = $x->getSheet();

// simple 1 cell read
$data = $s->read(0, 0);

//read data & obtain format
$data = $s->read(0, 0, $format);

// $format == ExcelFormat instance
```

# Formulas

```
$x = new ExcelBook();
$s = $x->addSheet("Sheet 1");

for ($i = 0; $i < 4; $i++)
    $s->write($i, 0, $i * $i);

$s->write(4, 0, "SUM(A1:A4)",
            NULL, ExcelFormat::AS_FORMULA);

$x->save("test.xls");
```

| | A |
|---|---|
| 1 | 0 |
| 2 | 1 |
| 3 | 4 |
| 4 | 9 |
| 5 | 14 |

# Reading Formulas

```php
$x = new ExcelBook();
$s = $x->addSheet("Sheet 1");

for ($i = 0; $i < 4; $i++)
    $s->write($i, 0, $i * $i);

$s->write(4, 0, "SUM(A1:A4)", NULL, ExcelFormat::AS_FORMULA);

var_dump($s->isFormula(4, 0)); // true
var_dump($s->read(4, 0)); // "SUM(A1:A4)"
```

# License Stuff

## In-Application

```
new ExcelBook("LICENSE NAME", "LICENSE KEY");
```

## Globally - INI Settings

```
excel.license_name

excel.license_key
```

# Other stuff you can do

- Freeze Panes
- Manipulate Excel Sheets
- Set printing layouts
- Copy Data
- Configure Cell Layout
- Zoom Controls
- Split Sheets
- Text Wrapping
- Text Alignment
- Lock & Freeze cells
- Etc….

# Thanks for Listening

Ilia Alshanetsky

http://ilia.ws/

@iliaa