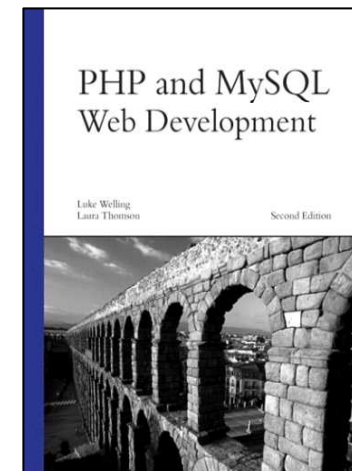
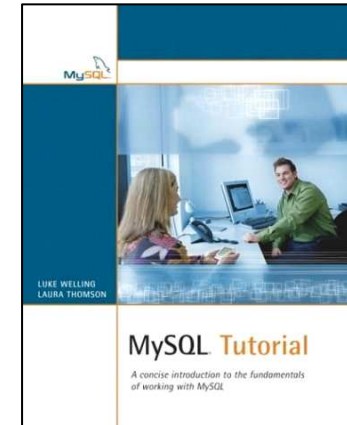


# Striving for Less Ugly Charts from PHP

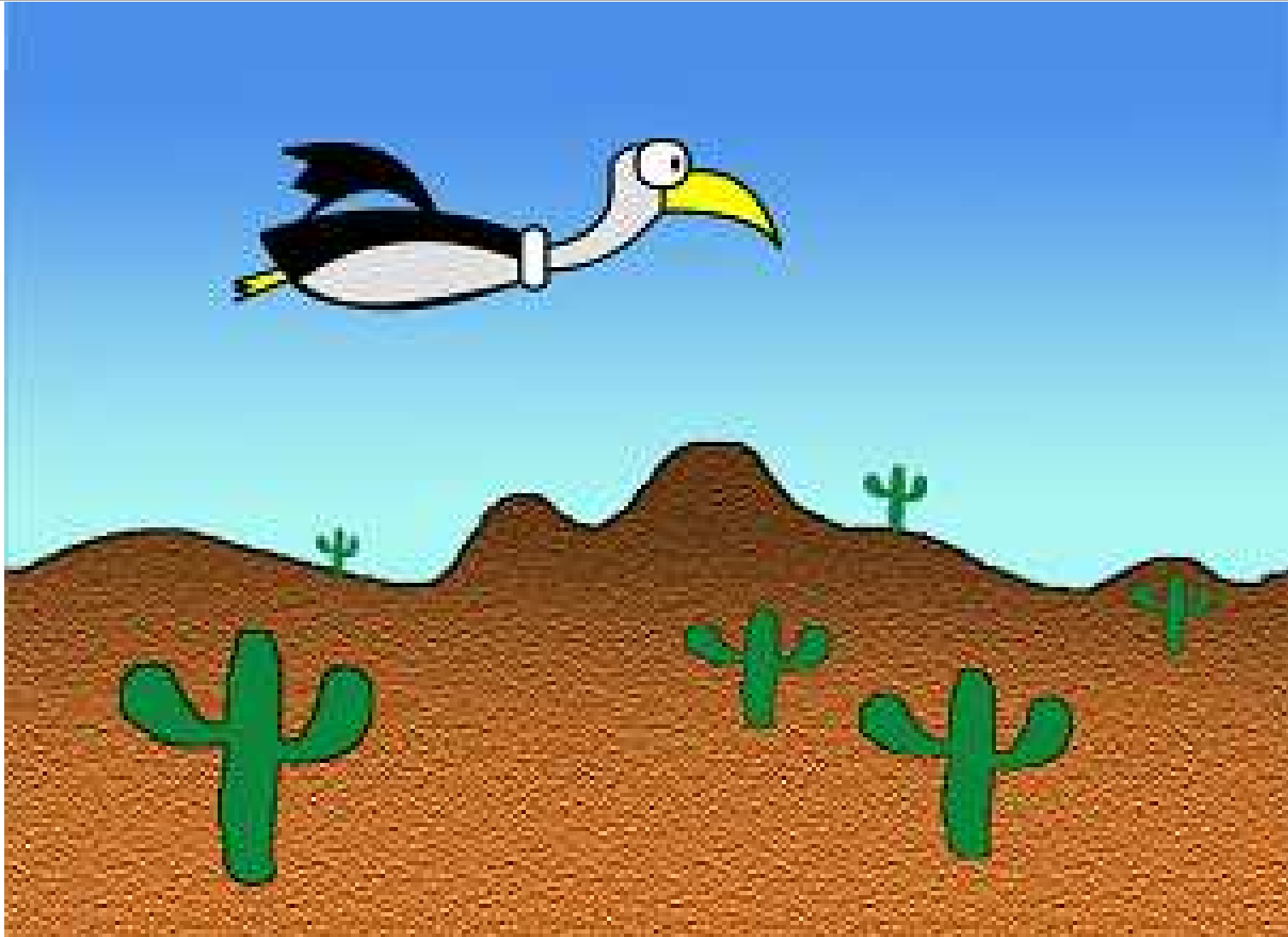
Luke Welling  
luke@omniti.com  
<http://lukewelling.com>

- Used MySQL and PHP since last century
- More than a decade of web development experience in a range of languages
- 2 years concentrating on building ugly charts



- Many applications seek to show quantitative data in a visual way
- Depending on the audience for the application, the degree that ugly matters will vary
- The default (and sometimes all) output from many popular libraries is ugly
- ... but there is a simple reason for that

# Programmers Can't Draw



- With that in mind, I am not really going to talk about
  - visual aspects of attractive charts
  - Guidelines for representing data graphically
- I am going to talk about technical aspects
  - Output formats
  - Libraries
  - Flexibility/customization
  - Development time

# The Bottom Line

- There is no "one true way"
- Tradeoff between flexibility and effort
- Output format and quality is an important, but unrelated consideration



- [Yahoo Finance \(beta\)](#)
- [Google Finance \(beta\)](#)
- [Google Analytics \(wow, not beta\)](#)

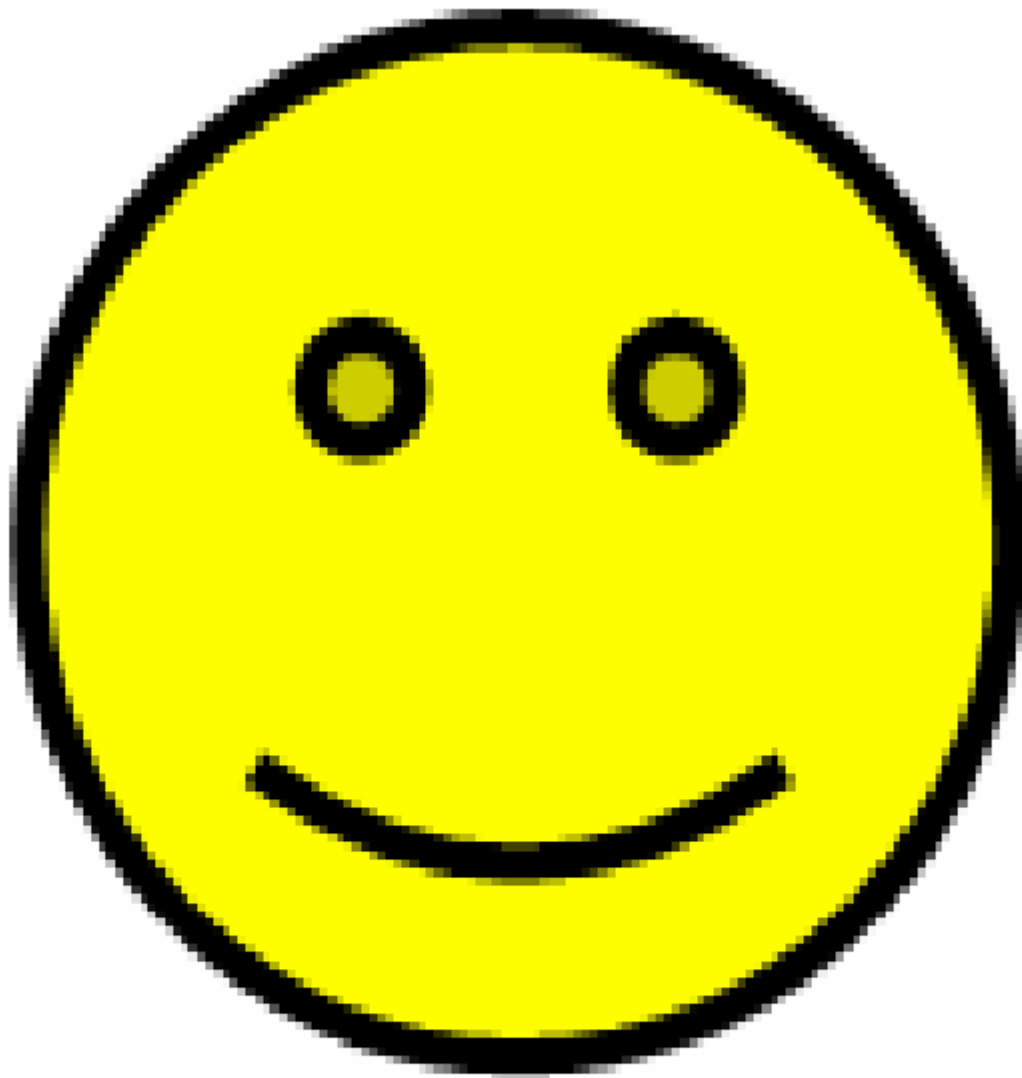
# Does it Matter?

- It is not just a matter of aesthetics
- Subjective assumptions about authority



(from theonion.com)

- Raster Based
  - GIF,
  - PNG,
  - JPEG
- Vector Based
  - Canvas
  - SVG
  - Flash
  - PDF



Raster

# All Raster Formats

- GIF, PNG, JPEG, BMP, TIFF, whatever
- All the same limitations and advantages, really, only the compression varies
- Predictable efficient output on all browser and virtually any device
- Targeted at only one resolution
- Fixed display size
- Ugly if zoomed
- Ugly if printed
- Ugly if your library does not handle anti aliasing well (for lines and for text)

**UGLY**

- Scalable Vector Graphics
- Vector based
- Great deal of flexibility if you (or a library) are willing to write the code to do it
- Fairly good Firefox support
- Needs an uncommon plugin (with an uncertain future) to work in IE which makes it a non-starter for most applications
- Scalable
- Printable
- Manipulate (alter, zoom, animate) with JavaScript

- Not directly available in IE
- From Apple (who reserve IP rights)
- Great deal of flexibility if you (or a library) are willing to write the code to do it
- Not able to manipulate existing elements with JavaScript (but can redraw)
- Printable

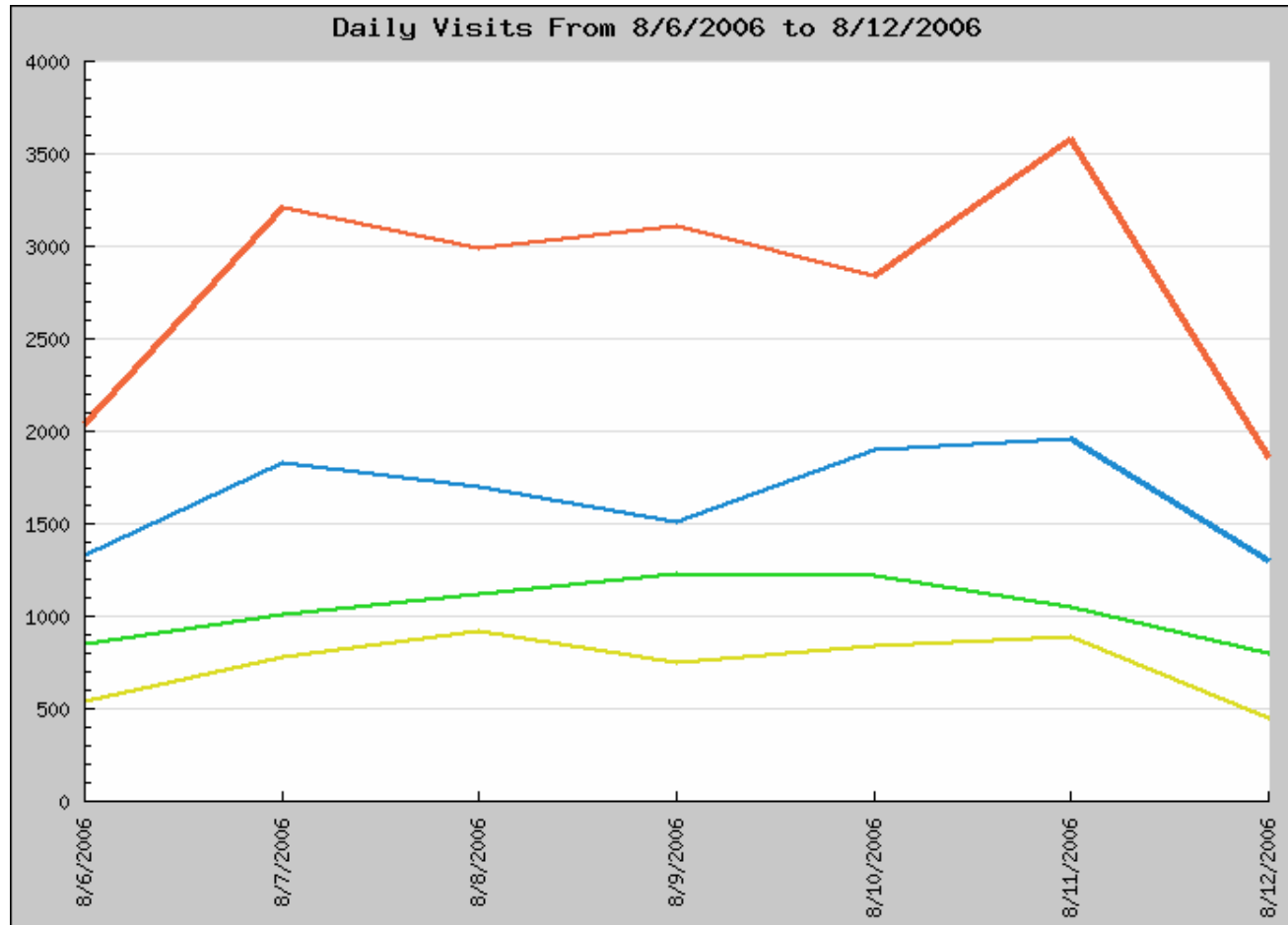
- Open Standard
- From Adobe
- Common, well known, but not universal plugin required
- Big plugin with long startup times for many people
- Requires some tweaking if you want your PDF to open in an iframe without toolbars and page navigation
- Prints well
- Wide variety of generation tools available

- Almost universally installed plugin available for many browsers and platforms
- Proprietary standard – proprietary players
- Good commercial generation tools, variable quality free generation tools
- Good support for manipulation, interaction, animation

- Vector Markup Language
- XML based
- Microsoft, and therefore ignored outside IE
- Manipulate through JavaScript

- Browser detection, and then SVG for FireFox and VML or Flash for IE is a viable option
- Means doing everything at least twice though, so you need a library to abstract it for you
- Dojo does SVG/VML
- eZ Components does SVG/Flash/PNG

- JpGraph
- PDF (various)
- Dojo Charting
- Plotkit
- EZ Charting
- Prebuilt Flash Widgets
- DIY Flash
- Flash on the fly (Ming)



- [Demo](#)

```
<?php
require 'SampleData.php';
require_once("jpgraph/src/jpgraph.php");
require_once("jpgraph/src/jpgraph_line.php");

$data = new SampleData();

$graph = new Graph(700,500);
$graph->SetMargin(40,20,30,70);

//Specify the X-scale explicitly
//let the Y-scale be auto-scaled
$graph->SetScale("intlin",0,0,0,6);
$graph->title->Set($data->title1.' ' . $data->title2);
```

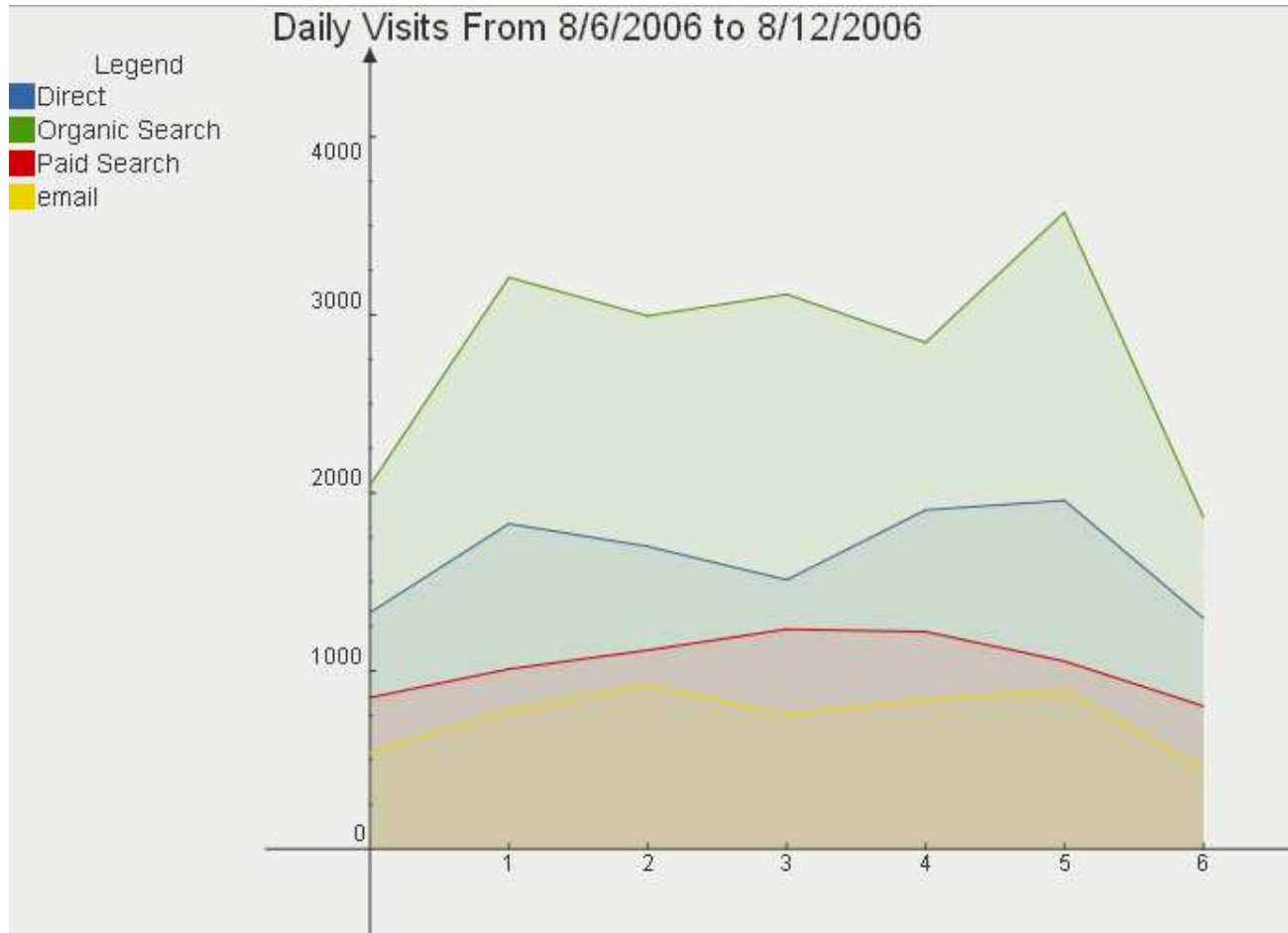
```
// values can only be integers so we need
// to "format" them as dates
$xvalues = range(0,6);
$graph->xaxis->setLabelAngle(90);
$graph->xaxis->
    setLabelFormatCallback('dateLabel');
function dateLabel($index)
{
    global $data;
    return $data->labels[$index];
}
```

```
// Add each line
for($i=0; $i<4; $i++)
{
    $line = 'line'.$i;

    $$line = new LinePlot($data->series[$i], $xvalues);
    $$line->SetWeight(2);
    $$line->SetColor('#'.$data->colours[$i]);
    $graph->Add($$line);
}
// finally draw the graph
$graph->Stroke();
?>
```

- Not very popular for this application
- I don't have a preferred library, but there are any number of tools to convert to PDF, many to directly generate PDFs and a few to generate PDF charts
- Possibilities include:
  - An application you like (Excel, Crystal Reports, ...) printing to PDF
  - Directly generating PDFs in PHP (pdf\_lib)
  - Directly generate PDF chart
- [Demo](#) (Excel and PDF-Chart-Creator)
- [Tweaked Display Demo](#) (TweakPDF)

- SVG/Flash/PNG output means "universal" compatibility
- Default output is attractive
- BSD licence
- PHP based, so convenient to hack



- [Demo](#)

```
<?php

require_once
    "c:/htdocs/localhost/oscon07/ezc/Base/src/base.php";
require 'SampleData.php';

function __autoload( $className )
{
    ezcBase::autoload( $className );
}

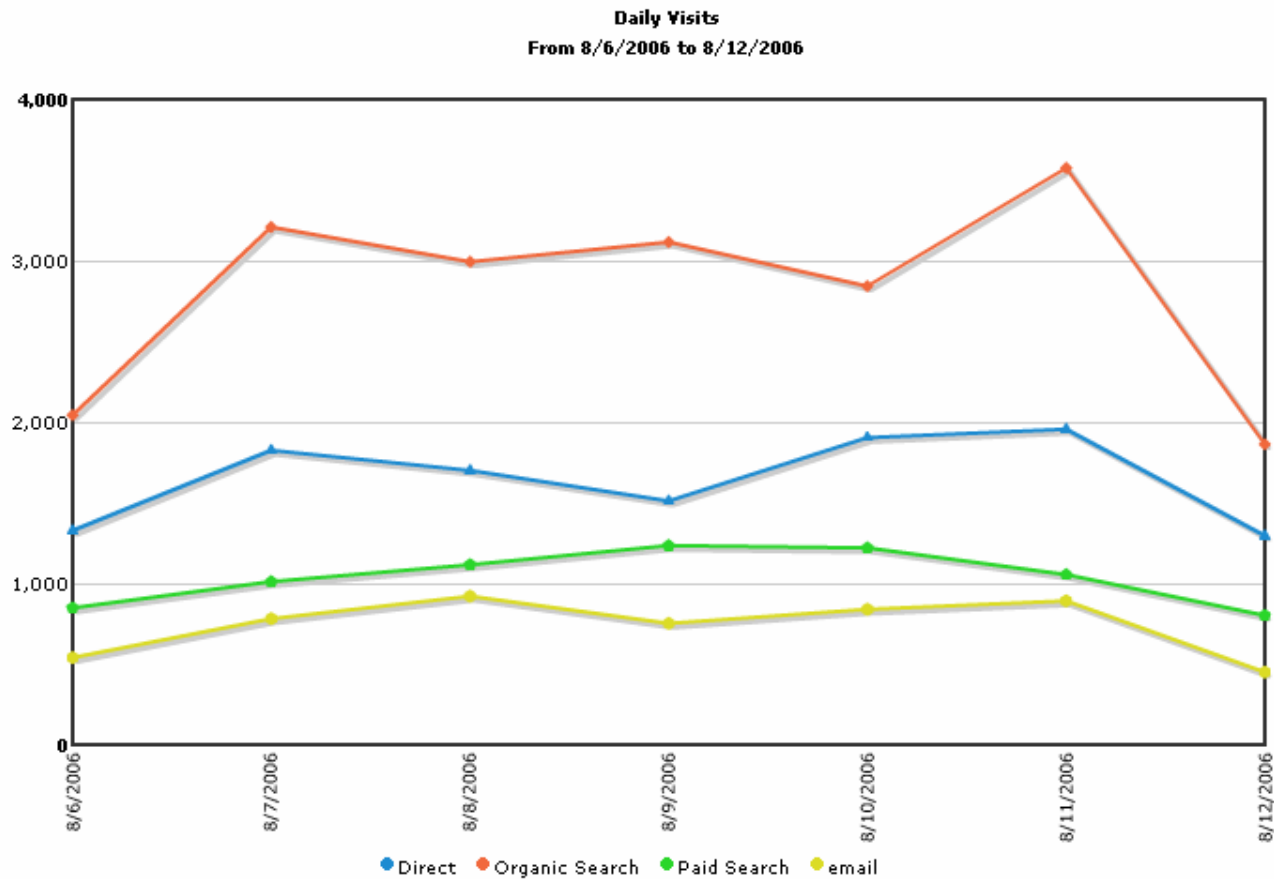
header( 'Content-Type: image/svg+xml' );

$data = new SampleData();
```

```
$chart = new ezcGraphLineChart();
$chart->title = $data->title1.' '.$data->title2;
$chart->palette = new ezcGraphPaletteTango();
$chart->options->fillLines = 230;
$chart->legend->title = "Legend";
$chart->xAxis->font->maxFontSize = 12;
$chart->yAxis->font->maxFontSize = 12;
$chart->title->font->maxFontSize = 20;
for($i = 0; $i<4; $i++)
{
    $chart->data[$i] = new ezcGraphArrayDataSet( $data-
        >series[$i] );
    $chart->data[$i]->label = $data->names[$i];
}
$chart->driver = new ezcGraphSvgDriver();
$chart->render( 700, 500, 'php://output' );
?>
```

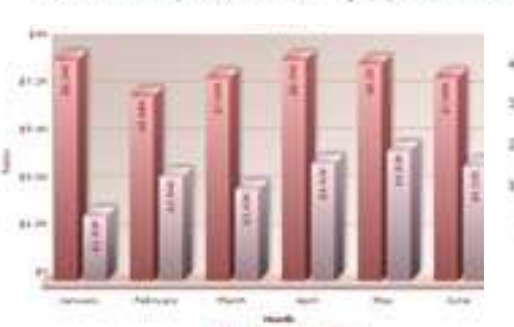
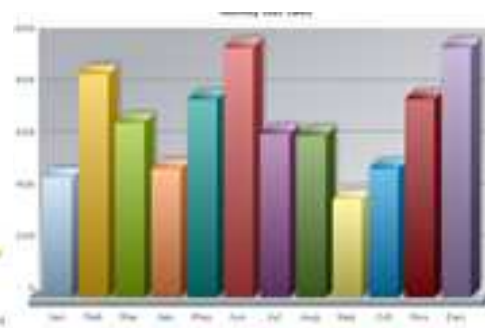
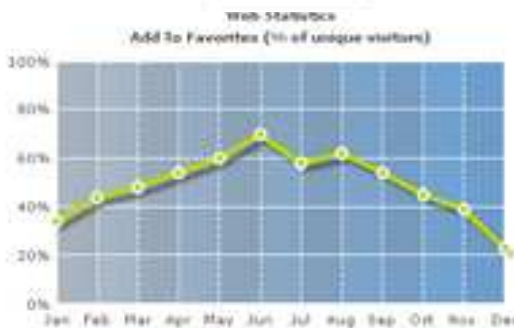
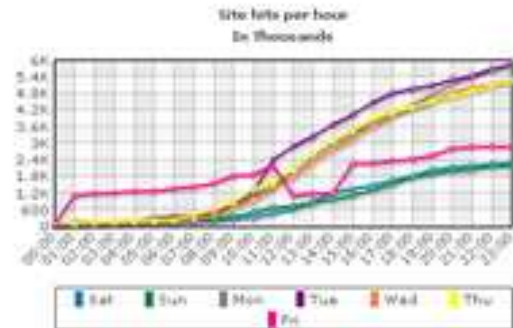
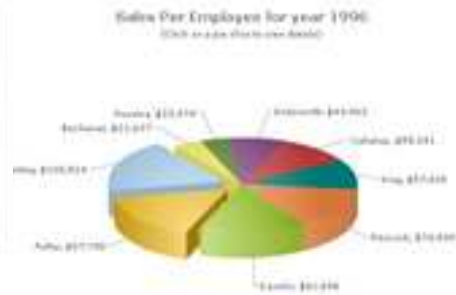
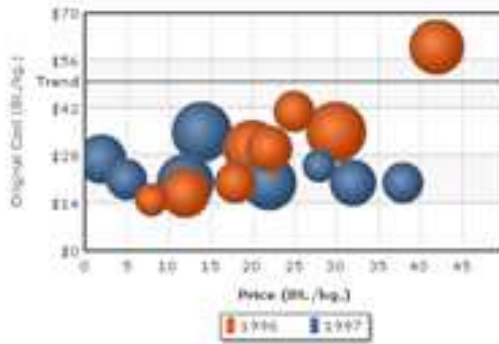
- VML/SVG output is a compelling idea
- Fairly immature at the moment
- JavaScript based

- In this case FusionCharts



- [Demo](#)

# Fusion Charts



- If you are aiming for very interactive charts with scrollable, zoomable time periods and similar features, you could look at Adobe Flex.
- It requires a moderate amount of work, but provides extensive built in functionality and great flexibility
- Commercial
- Their [demo](#)

- Right at the complete flexibility and hard work end of the spectrum.
- Ming is a PHP extension to generate Flash movies dynamically
- No built in charting support
- Experimental extension
- Free

```
<?php
$movie = new SWFMovie();
$movie->setDimension(500, 300);

$font = new SWFFont("Arial.fdb");
$text = new SWFText();
$text->setFont($font);
$text->setColor(0, 0, 0x33);
$text->setHeight(20);

// moving text to a zero coordinate does not work
$text->moveTo(1, 1);
$text->addString("1, 1");

$text->moveTo(1, 300);
$text->addString("1, 300");

$text->moveTo(430, 1);
$text->addString("430, 1");

$text->moveTo(430, 300);
$text->addString("430, 300");
```

## Ming Example (2/2)

```
$movie->add($text);

$shape = new SWFShape();

$shape->setLine(2, 0, 0, 0x33);

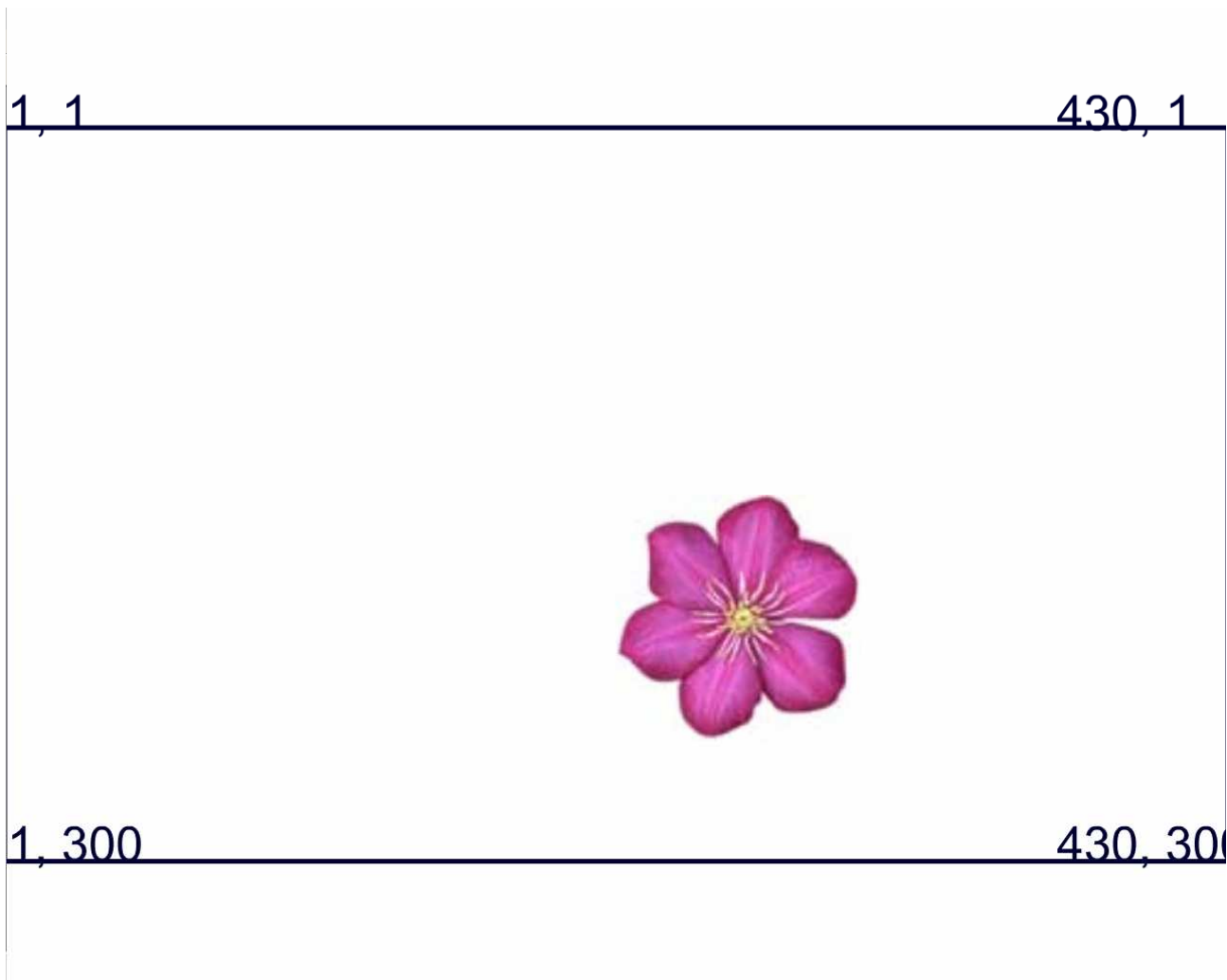
$shape->drawLineTo(500, 0);
$shape->drawLineTo(500, 300);
$shape->drawLineTo(0, 300);
$shape->drawLineTo(0, 0);

$movie->add($shape);

//place image in center
$image = new SWFBitmap(fopen('flower.jpg', 'r'));
$imageHandle = $movie->add($image);
$imageHandle->moveTo(250, 150);

header('Content-type: application/x-shockwave-flash');
$movie->output();
?>
```

# Ming "hello World"



- You need to work out what features you need and do a proof of concept in your chosen Library
- Everybody gives you lines, bars, pies, but not everybody gives you maps, gantts etc
- In the less flexible approaches, odd requirements (eg values counting down to zero, x-axis not at 0) might make some libraries non starters
- There is generally a trade-off between flexibility and work