

Create Charts Dynamically

Sunday, 24 February 2008

Charts are graphical information that represent numerical data and its relationship. Charts can be created using different softwares or spreadsheets, and they are often incorporated in presentations or slides to help audiences understand the data easily.

Google provides a free and great chart API that can be used to dynamically generate charts for websites. A complete guide describing the API and how to use it can be found in the developer's guide page.

Several types of charts can be created using the API. Some of them are line, bar, pie, and venn diagram. Each chart has attributes such as type, colors, size, title, and labels. The chart generated by the API is a PNG-format image. It can be embedded in a web page using tag. Due to URL length restriction, the data for the chart must be mapped into one of the encoding supported by the API, i.e. simple encoding, text encoding, or extended encoding.

When creating chart dynamically, it is convenience to translate real-life data into chart API programmatically rather than manually. An example to do this, using Javascript, can be found in the guideline page. Another example is to use Perl scripts, as shown below. The codes read the data from a text file in the following format:

The column is separated by a tab. The first row specifies the header, and the first column specifies the label of the data.

Data below was taken from Wikipedia page about Aceh and will be used for the demonstration. Assume that the data is stored in a text file in a correct format as mentioned above.

```
Sector (% Aceh GDP) 2003 2004 2005 2006  Agriculture and fisheries 17.0 20.0 21.4 21.2  Oil, Gas and Mining 36.1
30.4 26.2 24.9  Manufacturing Industries 20.2 18.3 15.9 14.3  Electricity and Water Supply 0.1 0.1 0.2 0.2  Building /
Construction 3.4 3.8 3.5 5.1  Trade, hotels and restaurants 11.2 12.0 14.3 15.0  Transport & Communication 3.3 3.8 4.8
5.2  Banking & other Financial 0.9 1.2 1.2 1.3  Services 7.8 10.4 12.7 12.9  Source: http://en.wikipedia.org/wiki/Aceh
```

A Script to Create Pie Chart

```
#!/usr/bin/perl -w
# Author: tfa@www.acehonline.com
# Create pie charts using Google Chart API
# http://code.google.com/apis/chart/
#

use strict;
use CGI;
my $cgi = new CGI;

# params
my $datapath = defined($cgi->param('datapath')) ? $cgi->param('datapath') : 'data';
my $datafile = defined($cgi->param('datafile')) ? $cgi->param('datafile') : 'chartdata.txt';
my $pagetitle = defined($cgi->param('pagetitle')) ? $cgi->param('pagetitle') : 'www.acehonline.com';
my $pietype = defined($cgi->param('pietype')) ? $cgi->param('pietype') : 'p3'; # p or p3
my $chartsize = defined($cgi->param('chartsize')) ? $cgi->param('chartsize') : '520x160'; # w x h
my $chartcolor = defined($cgi->param('chartcolor')) ? $cgi->param('chartcolor') : '00CC00';
my $chartbgclr = defined($cgi->param('chartbgclr')) ? $cgi->param('chartbgclr') : 'F3EDBF';
my $onlycol = defined($cgi->param('onlycol')) ? $cgi->param('onlycol') : 0;

# format datafile:
#
# header <tab> col1 <tab> ... colk
# label1 <tab> data11 <tab> ... data1k
# label2 <tab> data21 <tab> ... data2k
#

if(! -e "$datapath/$datafile"){
    print "Data file not found\n";
    exit(0);
}

# get label and data value from datafile
my $row1 = `head -1 $datapath/$datafile`; # get the header = 1st line
my @header = split('\t', $row1);
```

```

my @cols = `cut $datapath/$datafile -f1`; # get the label = 1st col
@cols = @cols[1..$#cols];
my $label = "";
foreach(@cols){
  chomp;
  s/\s+$/g;
  if($label eq ""){
    $label .= $_;
  }else{
    $label .= "|$_";
  }
}

print $cgi->header({-type=>'text/html', -charset=>'UTF-8'});
print $cgi->start_html({-title=>"$pagetitle"});

# get data values in each col
for my $i(1..$#header){
  next if(($onlycol > 0) && ($i != $onlycol));
  my $t = $i+1;
  my @vals = `cut $datapath/$datafile -f$t`;
  my $title = $vals[0]; # title is in the first row of each col
  chomp($title);
  my @data = ();
  my $maxval = 0;
  for my $k(1..$#vals){ # populate each val in the row to array data and get max value
    chomp($vals[$k]);
    push(@data, $vals[$k]);
    $maxval = $vals[$k] if($vals[$k] > $maxval); # get max val
  }
  my $chartdata = encoding(\@data, $maxval);
  print $cgi->img({-src=>"http://chart.apis.google.com/chart?cht=$pietype&chd=$chartdata" .
    "&chs=$chartsizesize&chl=$label&chtt=$title&chf=bg,s,$chartbgclr&chco=$chartcolor"});
}

print $cgi->end_html;

#___DONE___

sub encoding{
  my $dataval = shift;
  my $maxval = shift;
  my @encoding = qw(A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i
    j k l m n o p q r s t u v w x y z 0 1 2 3 4 5 6 7 8 9);
  my $chartdata = "s:";
  my $len = $#encoding;
  foreach(@$dataval){
    my $curval = $_;
    if($_ >= 0){
      my $pos = int($len * $curval / $maxval);
      $chartdata .= $encoding[$pos];
    }else{
      $chartdata .= ' ';
    }
  }
  return $chartdata;
}

```

Usage and Output

A simple encoding was used in the example above. The script reads a few parameters, e.g. the path of data file, the name of data file, and several optional parameters such as pie chart type, chart size, chart color, and background color. The following examples show the usages and outputs of the script.

1. Use onlycol parameter to generate a pie chart from a specific year (specified by the column):
<http://www.acehonline.com/cgi-bin/genpiechart.cgi?chartcolor=44AA00&chartsize=550x160&onlycol=3>
2. Remove parameter onlycol to display all charts in one page:
<http://www.acehonline.com/cgi-bin/genpiechart.cgi?chartcolor=44AA00&chartsize=550x160>
3. Use pietype=2 for two dimensional pie chart and chartcolor to specify specific colors of the chart:
<http://www.acehonline.com/cgi-bin/genpiechart.cgi?chartcolor=44AA00,775544,0044DD&pietype=p&onlycol=1>