

Génération de graphismes vectoriels avec Php

Code: visu-gen

Originaux

[url: http://tecfa.unige.ch/guides/tie/html/visu-gen/visu-gen-html](http://tecfa.unige.ch/guides/tie/html/visu-gen/visu-gen-html)

[url: http://tecfa.unige.ch/guides/tie/pdf/files/visu-gen.pdf](http://tecfa.unige.ch/guides/tie/pdf/files/visu-gen.pdf)

Auteurs et version

- [Daniel K. Schneider](#)
- Version: 0.7 (modifié le 6/3/06)

Prérequis

[Module technique précédent: xml-tech](#)

[Module technique précédent: svg-intro](#)

[Module technique précédent: php-intro](#)

[Module technique précédent: php-html](#)

[Module technique précédent: php-libs](#)

Module technique précédent: php-xml

Module technique suppl.: svg-xslt (Alternative: Visualisation avec XSLT !)

Abstract

Petite introduction à la génération de graphismes vectoriels (SVG, VRML, etc.) avec PHP et/ou XSLT. Surtout une petite introduction à la visualisation de données.

- Attention: Comprendre les exemples nécessite certains prerequis (voir ci-dessus)

A faire

- beaucoup ...
- par exemple: lire un template SVG et le modifier avec PHP/DOM, un exemple X3D,

Objectifs

- Introduire les techniques de base pour la visualisation

1. Table des matières détaillée

1. Table des matières détaillée	3
2. Introduction: contenus exotiques avec PHP	4
2.1 Le Mime type et les entêtes dans les fichiers	4
A.Définition du mime type	4
B.Les entêtes de vos fichiers	5
Exemple 2-1:Exemple complet avec SVG	5
C.XHTML avec d'autres namespaces	6
2.2 Principes de base de la visualisation	7
3. PHP-SVG avec des librairies SVG	8
3.1 phpHtmlLib	8
Exemple 3-1:Exemple SVG-phpHtmlLib simple	9
Exemple 3-2:Exemple SVG-phpHtmlLib SVGXYLineGraph	10
3.2 SVG Class de Killian	11
Exemple 3-3:Exemple d'animation avec SVG class	11
4. Alignement d'éléments SVG en cercle	13
4.1 Trigonométrie	13
4.2 Alignement en cercle avec PHP	15
Exemple 4-1:Génération d'éléments SVG autour d'un cercle	15
Exemple 4-2:XML vers SVG et visualiser autour d'un cercle	16
4.3 Alignement en cercle avec XSLT et un brin de PhP	17
Exemple 4-3:Génération d'éléments SVG autour d'un cercle	17
5. PHP VRML	18
Exemple 5-1:Simple exemple de génération VRML	18

2. Introduction: contenus exotiques avec PHP

2.1 Le Mime type et les entêtes dans les fichiers

Lorsque vous produisez d'autres contenus que HTML avec PHP, il faut veiller à deux choses:

1. Votre serveur doit indiquer à votre client de quel type de fichier il s'agit (indiquer le "Mime Type")
2. Votre fichier doit aussi contenir les autres déclarations nécessaires

A. Définition du mime type

- Cette instruction qui modifie le message HTTP du serveur doit intervenir tout au début du fichier ! (donc éviter tout echo, print, ou code en dehors de php avant)

Exemple SVG

```
header("Content-type: image/svg+xml");
```

Exemple PNG

```
Header("Content-type: image/png");
```

Exemple VRML

```
Header("Content-type: model/vrml");
```

B. Les entêtes de vos fichiers

- Il s'agit des lignes 2 (et 3, ...) dans votre fichier.

Exemple SVG

```
print('<?xml version="1.0" encoding="iso-8859-1"?>' . "\n");
print('<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN" "http://
www.w3.org/TR/2001/PR-SVG-20010719/DTD/svg10.dtd">' . "\n");
```

Exemple VRML

```
echo "#VRML V2.0 utf8";
```

Exemple 2-1: Exemple complet avec SVG

[url: http://tecfa.unige.ch/guides/php/examples/svg-simple/](http://tecfa.unige.ch/guides/php/examples/svg-simple/)

[url: http://tecfa.unige.ch/guides/php/examples/svg-simple/standalone-hello.php](http://tecfa.unige.ch/guides/php/examples/svg-simple/standalone-hello.php)

```
<?php
header("Content-type: image/svg+xml");
print('<?xml version="1.0" encoding="iso-8859-1"?>' . "\n");
print('<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN" "http://www.w3.org/TR/
2001/PR-SVG-20010719/DTD/svg10.dtd">' . "\n");
echo "<svg xmlns='http://www.w3.org/2000/svg'>\n";

print ('<rect x="50" y="50" rx="5" ry="5" width="200" height="100"
style="fill:#CCCCFF;stroke:#000099"/>' . "\n");
print('</svg>' . "\n");
?>
```

C. XHTML avec d'autres namespaces

- Avec des navigateurs comme Firefox 1.5 + vous pouvez combiner du code XHTML et SVG plus Xlink (par exemple)
- Toutefois, pour que cela marche, le fichier doit être servi comme XML !
 - ... et de préférence en xhtml+xml
 - n'oubliez pas non plus d'indiquer le namespace svg dans la balise <html>
- à chaque fois que vous insérez du SVG, indiquez aussi le namespace

```
// header ("Content-type: application/xml");
header ("Content-type: application/xhtml+xml");
echo '<?xml version="1.0" encoding="ISO-8859-1" ?>';
echo '<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">';
echo '<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:svg="http://www.w3.org/2000/svg">';
.....
.... echo "<p> ceci est du html </p>";
....
echo "<svg xmlns='http://www.w3.org/2000/svg'
      xmlns:xlink='http://www.w3.org/1999/xlink'
      height='100' width='200' x='200' y='0'>";
    <circle fill='#183952' r='10' />
echo "</svg>
```

2.2 Principes de base de la visualisation

- Il est facile de générer des contenus encodés avec un format "texte" et dont les règles sont connues.
 - Exemples: toutes les applications XML comme XHTML, SVG, RDF, RSS.
 - Principe: on produit le contenu avec une instruction comme echo, print(), etc.
 - Il existe des bibliothèques écrites en PHP qu'on peut trouver sur Internet et qui vous faciliteront le travail.
- Pour générer des formats binaires comme PNG ou Flash il faut installer des extensions spéciales à PHP et ensuite travailler avec l'API fourni.
 - Exemple: Xlvi. Image Functions dans le manuel PHP: JPEG, PNG etc.
 - Dans ce texte, on en parlera pas ...

3. PHP-SVG avec des librairies SVG

Travailler avec une librairie améliore la flexibilité et le coût de maintenance de votre code. Si vous ne trouvez rien qui vous convient, faites la vôtre (au moins quelques fonctions utiles pour les opérations répétitives).

3.1 phpHtmlLib

[url: http://phphtmlib.newsblob.com/](http://phphtmlib.newsblob.com/) (site phphtmlib)

[url: http://tecfa.unige.ch/lib/php/phphtmlib/](http://tecfa.unige.ch/lib/php/phphtmlib/) (installation local à TECFA)

[url: http://tecfa.unige.ch/lib/php/phphtmlib/doc/](http://tecfa.unige.ch/lib/php/phphtmlib/doc/) (documentation à TECFA)

phpHtmlLib est une très grande librairie à fonctions multiples

- Du manuel: phpHtmlLib is a set of PHP classes and library functions to build, debug, and render XML, HTML, XHTML, WAP/WML Documents, and SVG (Scalable Vector Graphics) images as well as complex html 'widgets' for PHP 4.2 or better.
- Populaire, de grande qualité, "lourd", nécessite une certaine expertise pour interpréter le manuel

Exemple 3-1: Exemple SVG-phpHtmlLib simple

<http://tecfa.unige.ch/guides/php/examples/svg-phphtmllib/> (phphtmllib-simple.*)

```
$phphtmllib = $_SERVER["DOCUMENT_ROOT"] . "/lib/php/phphtmllib";
include_once("$phphtmllib/includes.inc");
include_once($phphtmllib."/widgets/svg/SVGDocumentClass.inc");

//0. Create a page object with dimensions
$svgpage = new SVGDocumentClass(800,600);

//1. add a nice rectangle
$rect = svg_rect(50,50,600,400,"none","black",3);

//2. add polyline
$line = "50,375 150,375 150,325 250,325 250,375 350,375 350,250 450,250
450,375 550,375 550,175 600,175";
$zoliline = svg_polyline($line,"none","blue","2");

//3. Add an ellipse
$oeuf = svg_ellipse(300,300,200,100,"red","black","1");

//4. Add these objects to the page object
$svgpage->add($rect);
$svgpage->add($oeuf);
$svgpage->add($zoliline);

//5. render the whole thing
print $svgpage->render();
```

Exemple 3-2: Exemple SVG-phpHtmlLib SVGXYLineGraph

<http://tecfa.unige.ch/guides/php/examples/svg-phhtmlib/> (SVGXYLineGraph.*)

- Cet exemple utilise un widget pour faire des "line graphs"
- Pour la documentation voir la classe `SVGXYLineGraph` (y compris méthodes héritées)

```
.....
$width = 500;
$height = 500;

$svgdoc = new SVGDocumentClass("100%", "100%");
$graph = new SVGXYLineGraph("Popularity/usage/activity of TECFA's PHP
examples", $width, $height);
$graph->set_x_title("X-Axis Year");
$graph->set_y_title("Popularity");

//add three lines with different colors
$graph->add_line("0,1,2.3,4.2,6,8", "1,2,2.7,0.3,6,1", "red");
$graph->add_line("0,1,4.1,6", "0,4,2,3", "blue");
$graph->add_line("0,1,2,3,4,5,7", "0,4,3,1,7,8,10", "black");

//add the line graph widget to the document.
$svgdoc->add( $graph );

print $svgdoc->render();
```

3.2 SVG Class de Killian

[url: http://www.phpclasses.org/browse.html/package/457.html](http://www.phpclasses.org/browse.html/package/457.html)

- Cette classe semble être très bien faite (avec un petit volume)
- Mais elle vient sans aucune documentation (il faut regarder le code)

Exemple 3-3: Exemple d'animation avec SVG class

[url: http://tecfa.unige.ch/guides/php/examples/svg-class](http://tecfa.unige.ch/guides/php/examples/svg-class) (svgclass-example.php)

```
// *** Define the path to the SVG class dir. ***

define("SVG_CLASS_BASE", "./ori/");
// Include the class files.
require_once(SVG_CLASS_BASE."Svg.php");

// Create an instance of SvgDocument. All other objects will be added to this
// instance for printing. Also set the height and width of the viewport.
$svg =& new SvgDocument("400", "400");

// Create an instance of SvgGroup.
// Set the style, transforms for child objects.
$g =& new SvgGroup("stroke:black", "translate(200 100)");

// Add a parent to the g instance.
$g->addParent($svg);
```

```
// Create and animate a circle.
$circle = new SvgCircle("0", "0", "100", "stroke-width:3", "");
$circle->addChild(new SvgAnimate("r", "XML", "0", "75", "", "3s", "freeze"));
$circle->addChild(new SvgAnimate("fill", "CSS", "green", "red", "", "3s",
                                "freeze"));

// Once the circle is created move it right and down // DKS addition
$circle->addChild(new SvgAnimate("cx", "XML", "", "200", "4s", "3s",
                                "freeze"));
$circle->addChild(new SvgAnimate("cy", "XML", "", "200", "4s", "3s",
                                "freeze"));

// Make the circle a child of g.
$g->addChild($circle);

// Create and animate some text.
$text = new SvgText("0", "0", "SVG is cool",
                   "font-size:20;text-anchor:middle;", "");
$text->addChild(new SvgAnimate("font-size", "auto", "0", "20", "", "3s",
                               "freeze"));

// Make the text a child of g.
$g->addChild($text);

// Send a message to the svg instance to start printing.
$svg->printElement()
```

4. Alignement d'éléments SVG en cercle

(chapitre à développer)

4.1 Trigonométrie

- Pour placer des éléments autour d'un cercle il faut faire un peu de trigonométrie

radius = hypoténuse

x = côté adjacent

y = coté opposé

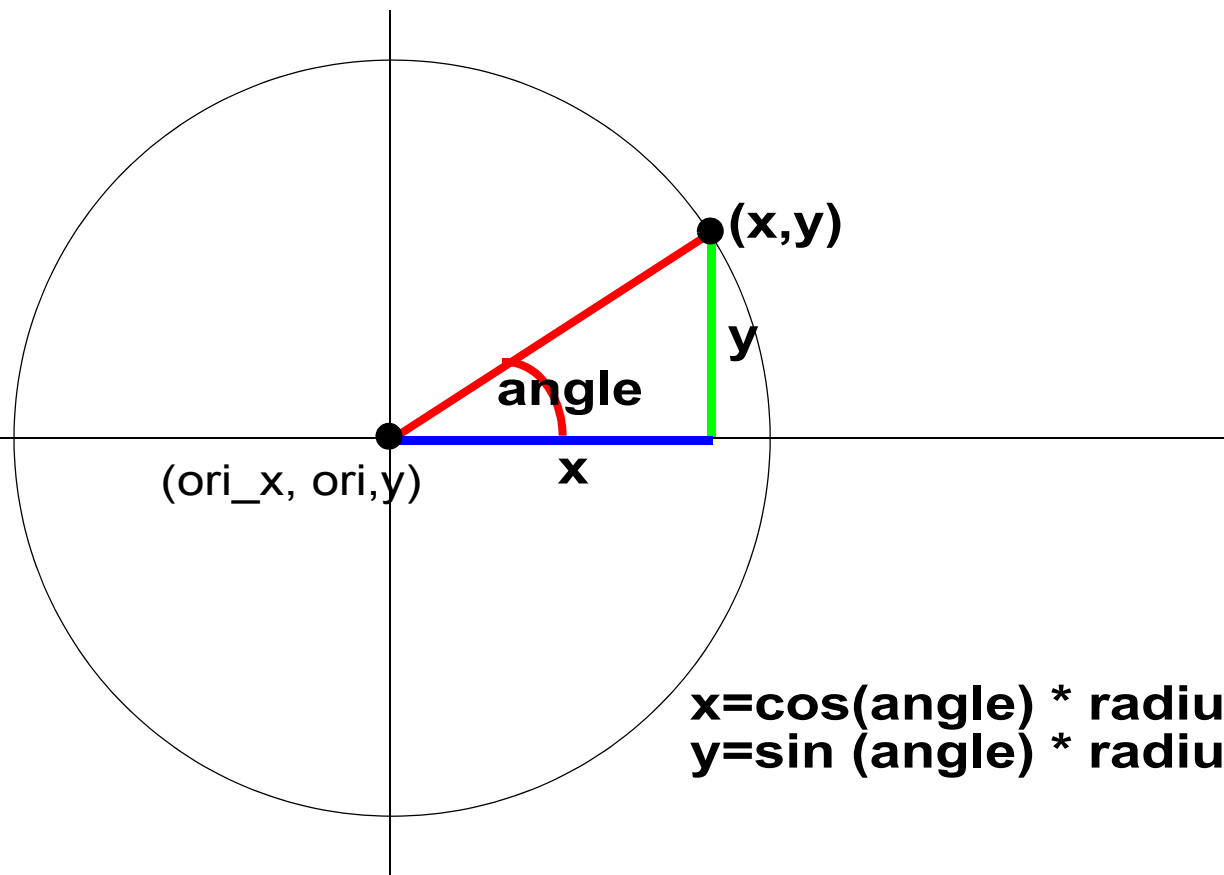
$\sin(\text{angle}) = y/\text{radius}$

$y = \sin(\text{angle}) * \text{radius}$

$\cos(\text{angle}) = x/\text{radius}$

$x = \cos(\text{angle}) * \text{radius}$

$\tan(\text{angle}) = y / x$



Formules:***Les angles sont exprimés en radians basé sur la constante Pi (3.142)***

- En informatique on utilise les radians (au lieu des degrés)
- Une cercle = 360 degrés = 2 Pi, 180 degrés = Pi, 90 degrés = Pi/2

Formule: $\text{rad} = \text{deg} / 180 * \text{Pi}$

Calculer la position x d'un élément en connaissant l'angle et le radius:

- Pour chaque élément à placer on incrémente l'angle: $i * \text{angle}$
- Pour les dessins vectoriels, il faut ajouter la position du cercle (ori_x)
- Pour rendre la fomule plus flexible: on ajoute l'angle de départ, on travaille avec un arc (partie d'une circle), et on utilise $(i \text{ div } n)$ qui permet de faire un 2ème tour.

$x = \text{ori_x} + \cos(\text{arc} / n_els * (i \% n_els) + \text{start_angle}) * \text{radius}$

ori_x = coordonnée x du cercle

arc = angle total de dessin, par ex. pour 10 éléments sur 360 degrés:

n_els = nombre d'éléments

$\text{angle} = \text{arc} / n_els = \text{Pi} * 2 / 10 = 6.283 / 10 = 0.63$

i = numéro de l'élément ($i \% n_els$ est égal à i dans ce contexte)

start_angle = angle de départ sur le cercle (ou commencer à dessiner)

Pour calculer la position y d'un élément en connaissant l'angle et le radius:

$y = \text{ori_y} + \sin(\text{arc} / n_els * (i \% n_els) + \text{start_angle}) * \text{radius}$

4.2 Alignement en cercle avec PHP

Exemple 4-1: Génération d'éléments SVG autour d'un cercle

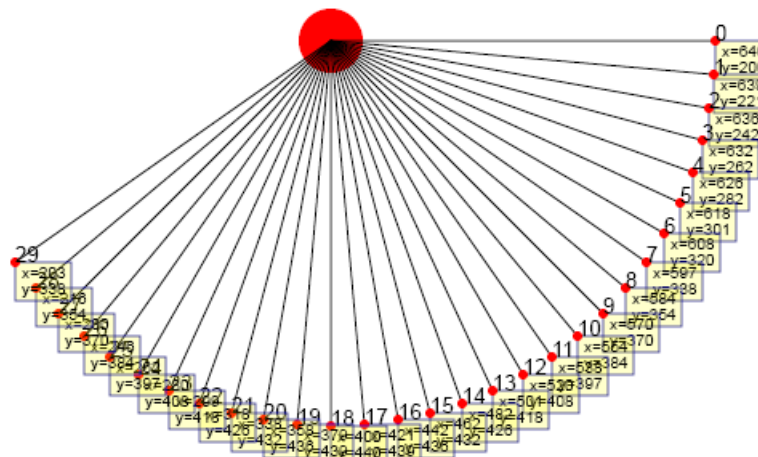
[url: http://tecfa.unige.ch/guides/svg/ex/objects-in-circles/](http://tecfa.unige.ch/guides/svg/ex/objects-in-circles/) (répertoire et source)

[url: elements-on-circle-with-php.php](#) (placer des rectangles autour d'un cercle)

[url: elements-on-arc-with-php.php](#) (même chose, mais sur un arc arbitraire)

(documentation à faire, voir le code source pour le moment)

Note pour le code: En graphisme par ordinateur (SVG et autres), l'origine (0,0) est en haut à gauche et y incrémente vers le bas, donc forcément le dessin commence à droite et vers le bas (au sens de la montre).



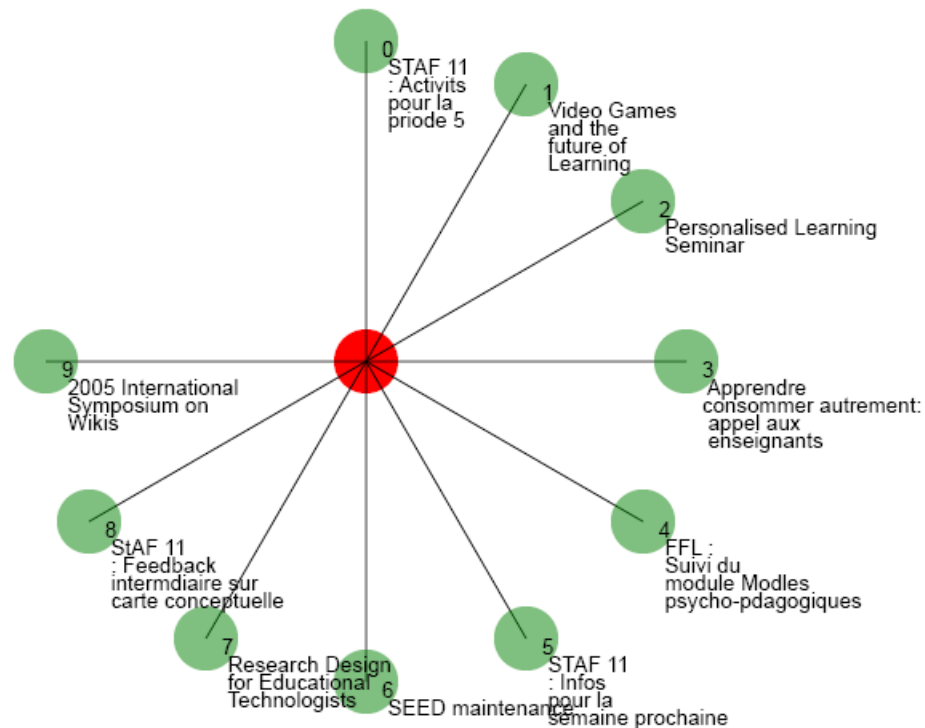
Exemple 4-2: XML vers SVG et visualiser autour d'un cercle

[url: http://tecfa.unige.ch/guides/svg/ex/objects-in-circles/](http://tecfa.unige.ch/guides/svg/ex/objects-in-circles/) (répertoire et sources)

[url: elements-on-arc-with-simple-xml.php](#) (lire des items RSS avec simple-xml)

[url: elements-on-arc-with-dom.php](#) (lire des items RSS avec DOM)

(documentation à faire, voir le code source pour le moment)



4.3 Alignement en cercle avec XSLT et un brin de PHP

Exemple 4-3: Génération d'éléments SVG autour d'un cercle

[url: http://tecfa.unige.ch/guides/svg/ex/objects-in-circles/](http://tecfa.unige.ch/guides/svg/ex/objects-in-circles/) (répertoire et source)

[url: elements-on-circle-with-xslt.php](#) (placer des rectangles autour d'un cercle)

[url: elements-on-circle-with-xslt.xsl](#) (feuille de style)

Fonctions trigonométriques en XSLT

- Il y en a pas (donc il faut soit les programmer avec JS, Java, ou en XSLT avec des tables etc., soit faire appel à PHP et donc faire le rendering en SVG du côté serveur)
- Pour que cela marche il faut enregistrer les fonctions php dans xslt:

```
$proc = new XSLTProcessor;  
// This allows to access ALL php functions within XSLT  
$proc->registerPHPFunctions();
```

Attention aux namespaces dans le fichier XSLT:

```
<xsl:stylesheet version="1.0"  
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"  
  xmlns:php="http://php.net/xsl"  
  xmlns:xlink="http://www.w3.org/1999/xlink">
```

(cf. aussi les transparents svg-xslt pour des détails !)

5. PHP VRML

Exemple 5-1: Simple exemple de génération VRML

- planter 100 arbres sans se fatiguer

[url: voir: /guides/php/exemples/vrml-temple/mixing/](#)

```
<? Header("Content-type: model/vrml");
    echo "#VRML V2.0 utf8"; ?>

// ici on définit un prototype pour arbre (code pas montré)

PROTO Tree ....

Transform {
    translation -5 0 -10
    children [
<?
for ($i=0; $i<10; $i++) {
    for ($j=0; $j<10; $j++) {
        echo "Tree { translation $i 0 $j }";
    }
}
?>
    ] }
```