Bibliothèque de cartographie

Variables globales à déclarer
Nombre de régions à traiter : nb_regions (USA : 51)

Données :
Structure [nom de la région , x du centre , y du centre , données statistiques …….]

Objets SVG prédéfinis :
Dans l'ordre :
Le rectangle contenant la figure avec comme id XXXX00
Les régions avec pour id XXXX01 XXXX02 ……
XXXX sera envoyé aux fonctions comme préfixe (USA : 'reg')

Les rectangles pour les couleurs de la légende avec pour id :
XYZ1 XYZ2 XYZ3 …….
XYZ sera envoyé aux fonctions comme préfixe (USA : 'rectleg')

Les textes des légendes avec pour id :
XYZT1 XYZT2 …….XYZTn n déclaré comme nb_legendes
Objets cachés au départ

Un objet texte comme titre avec pour id : YUZG (USA : 'titre')

Eventuellement des symboles (disques, camemberts ou autres)

Affichage du titre de la carte
Paramètres : evt nom : id de l'objet texte valeur : chaîne à afficher

```
function titre(evt,nom,valeur)
{var svgdoc=evt.getTarget().getOwnerDocument();
obj=svgdoc.getElementById(nom);
child=obj.getFirstChild();child.setData(valeur)}
```

Exemple d'appel :
```
titre(evt,"titre","Population totale (1999)");
```

Remplissage des rectangles de la légende
Paramètres : evt prefixe : id générique
couleur(tableau : les couleurs) Autant de couleurs que de rectangles à remplir
opacité (tableau : les valeurs d'opacité) Autant d'opacités que de rectangles à remplir

```
function legendage(evt,prefixe,couleur,opacite)
{num=couleur.length;var svgdoc=evt.getTarget().getOwnerDocument();
for (i=1;i<=num;i++)
{obj=svgdoc.getElementById(prefixe+i.toString());
      obj.getStyle().setProperty("fill",couleur[i-1]);
      obj.getStyle().setProperty("fill-opacity",opacite[i-1])}}
```

Exemple d'appel
```
legendage(evt,"rectleg",["#cccccc","#cccccc","#cccccc"],["1.0","0.5","0.1"]);
```

Afficher la légende correspondante et cacher les autres
Paramètres : evt prefixe : id générique
etat tableau de l'état des légendes (0 : cachée ; 1 : visible)

```
function texte_legende(evt,prefixe,etat)
{nb_leg=etat.length;var svgdoc=evt.getTarget().getOwnerDocument();
for (i=1;i<=nb_leg;i++)
{obj=svgdoc.getElementById(prefixe+i.toString());
if (etat[i-1]==0) {obj.getStyle().setProperty("visibility","hidden")} else
{obj.getStyle().setProperty("visibility","visible")}}}
```

Exemple d'appel

```
texte_legende(evt,"texleg",[1,0,0,0]);
```

Remplissage des régions avec une même couleur avec des opacités différentes ou des couleurs différentes

Paramètres : evt prefixe (id générique des régions)
col1 col2 (données, si col2=-1, on prend col1 sinon on fait col1/col2)
q (quantiles : minimum, puis 1er quantile, puis …… et enfin maximum ;)
couleur (tableau : couleurs de remplissage) opacite (tableau : opacités de remplissage)
Variable globale : nb_regions

```
function degrades(evt,prefixe,col1,col2,q,couleur,opacite)
{nb_q=couleur.length;var svgdoc=evt.getTarget().getOwnerDocument();
for(i=1;i<=nb_regions;i++)
{ if (i<=9) {region=prefixe+"0"+i} else {region=prefixe+i};
if (col2!=-1) {valeur=donnees[i][col1]/donnees[i][col2]} else
{valeur=donnees[i][col1]};
obj=svgdoc.getElementById(region);
for (j=1;j<=nb_q;j++)
{if ((valeur>=q[j-1])&&(valeur<q[j]))
     {obj.getStyle().setProperty("fill",couleur[j-1]);
      obj.getStyle().setProperty("fill-opacity",opacite[j-1])}}}}
```

Exemples d'appel
Population des états :

```
degrades(evt,"reg",3,-1, [0,1500000,8000000,50000000],
["#cccccc","#cccccc","#cccccc"] , ["0.1","0.5","1.0"])
```

Densité

```
degrades(evt,"reg",3,4,[0,20,100,50000],["#ff0033","#ff0033","#ff0033"],
["0.1","0.5","1.0"])
```

Remplissage des régions en couleurs différentes suivant la valeur exacte d'une donnée

Paramètres : evt prefixe col1 (valeur à représenter)
q (tableau : valeurs à représenter) couleur (tableau : couleurs de remplissage)
opacite (tableau : opacités)
Variable globale : nb_regions

```
function colorie(evt,prefixe,col1,q,couleur,opacite)
{nb_q=couleur.length;
var svgdoc=evt.getTarget().getOwnerDocument();
for(i=1;i<=nb_regions;i++)
{ if (i<=9) {region=prefixe+"0"+i} else {region=prefixe+i};
obj=svgdoc.getElementById(region);
for (j=0;j<nb_q;j++)
{if (donnees[i][col1]==q[j])
     {obj.getStyle().setProperty("fill",couleur[j]);
      obj.getStyle().setProperty("fill-opacity",opacite[j])}}}}
```

Exemple d'appel
Prévisions des élections USA

```
colorie(evt,"reg",5,["B","I","G"],["blue","gray","red"],["0.5","0.5","0.5"])}
}
```

Valeurs à représenter 'B' (Bush favori) 'G' (Gore favori) 'I' (indécis …)

Affichage de disques de dimension variable suivant une valeur ou un rapport de deux valeurs
Objet à déclarer :

```
<g id="cercles">
<circle id="cercle" cx="-3703" cy="0" r="0" style="fill:red;fill-opacity:0.5"
/>
</g>
```

Paramètres : evt col1 col2 (données, si col2=-1, on prend col1 sinon on fait col1/col2)
rmin rmax (rayons extrêmes du disque) couleur opa nom (id de l'objet 'circle' à cloner) nœud (nœud du DOM où seront créés les disques)
Variable globale : nb_regions

```
function disques(evt,col1,col2,rmin,rmax,couleur,opa,nom,noeud)
{var svgdoc=evt.getTarget().getOwnerDocument();
var target=svgdoc.getElementById(nom);
var  contents = svgdoc.getElementById (noeud);
if (col2!=-1) {valeur=donnees[1][col1]/donnees[1][col2]} else
{valeur=donnees[1][col1]};
mini=valeur;maxi=valeur;
for (i=2;i<=nb_regions;i++)
{if (col2!=-1) {valeur=donnees[i][col1]/donnees[i][col2]} else
{valeur=donnees[i][col1]};
if (valeur<mini) {mini=valeur};
if (valeur>maxi) {maxi=valeur}};
for (i=1;i<=nb_regions;i++)
{var newnode = target.cloneNode(false);
etiq=nom+i.toString();
newnode.setAttribute ('id', etiq);
newnode.setAttribute("cx",donnees[i][1].toString());
newnode.setAttribute("cy",donnees[i][2].toString());
if (col2!=-1) {valeur=donnees[i][col1]/donnees[i][col2]} else
{valeur=donnees[i][col1]};
rayon=rmin+Math.round((rmax-rmin)*(valeur-mini)/(maxi-mini));
newnode.setAttribute("r",rayon.toString());
newnode.getStyle().setProperty("fill",couleur);
newnode.getStyle().setProperty("fill-opacity",opa);
newnode = contents.appendChild (newnode)}}
```

Exemple d'appel

Nombre de grands électeurs aux USA

```
disques(evt,4,-1,5,25,"red","0.5","cercle","cercles");
```

Effacer des symboles (disques ou autres)

Paramètres : evt nom (tableau id génériques des objets à effacer)
nœud (nœud du DOM où ont été créés les objets)

```
function efface(evt,nom,noeud)
{nb_obj=nom.length;
var svgdoc=evt.getTarget().getOwnerDocument();
for (j=0;j<nb_obj;j++)
{for (i=1;i<=nb_regions;i++)
{cible=nom[j]+i.toString();obj=svgdoc.getElementById(cible);
var  contents = svgdoc.getElementById (noeud);
contents.removeChild (obj)}}};
```

Pour effacer les disques créés

```
efface(evt,["cercle"],"cercles")
```

Affichage de symboles de dimension variable suivant une valeur ou un rapport de deux valeurs

Objet à déclarer :

Un exemplaire du symbole ici nommé ‘maison’, ce symbole peut être aussi compliqué qu’on veut, on ne pourra pas dans cet exemple changer la couleur des clones
Sa taille sera obtenue par une matrice, l’homothétie étant toujours centré en 0 ; 0, le symbole devra être centré en 0 ; 0.
La matrice s’appliquera à tous les éléments du symbole.
Au départ le symbole est caché.

```
<g id="symboles">
<g id="maison" transform="matrix(1 0 0 1 0 0)"
style="stroke:black;visibility:hidden;fill-opacity:0.7">
<path d="M-50 -50l100 0 -50 -30 z" style="fill:red" />
<path d="M-45 -50l90 0 0 100 -90 0 z" style="fill:yellow" />
<path d="M-15 50l30 0 0 -40 -30 0 z M-30 -10l20 0 0 -20 -20 0 z M30 -10l-20 0
 0 -20 20 0 z" style="fill:white" />
</g>
</g>
```

Paramètres : evt col1 col2 (données, si col2=-1, on prend col1 sinon on fait col1/col2)
rmin rmax (rayons extrêmes du disque) couleur opa nom (id de l’objet ‘circle’ à cloner) nœud (nœud du DOM où seront créés les disques)
Variable globale : nb_regions

```
function symboles(evt,col1,col2,dmin,dmax,opa,nom,noeud)
{var svgdoc=evt.getTarget().getOwnerDocument();
var target=svgdoc.getElementById(nom);
var  contents = svgdoc.getElementById(noeud);
if (col2!=-1) {valeur=donnees[1][col1]/donnees[1][col2]} else
{valeur=donnees[1][col1]};
mini=valeur;maxi=valeur;
for (i=2;i<=nb_regions;i++)
{if (col2!=-1) {valeur=donnees[i][col1]/donnees[i][col2]} else
{valeur=donnees[i][col1]};
if (valeur<mini) {mini=valeur};
if (valeur>maxi) {maxi=valeur}};
for (i=1;i<=nb_regions;i++)
{var newnode = target.cloneNode(false);
etiq=nom+i.toString();
newnode.setAttribute ('id', etiq);
if (col2!=-1) {valeur=donnees[i][col1]/donnees[i][col2]} else
{valeur=donnees[i][col1]};
taille=dmin+Math.round(10*(dmax-1)*(valeur-mini)/(maxi-mini))/10;
matrice="matrix("+taille+" 0 0 "+taille+" "+donnees[i][1].toString()+"
"+donnees[i][2].toString()+")";
```

```
newnode.getStyle().setProperty("visibility","visible");
newnode.setAttribute("transform",matrice);
newnode = contents.appendChild (newnode)}}
```

Exemple d'appel
```
symboles(evt,5,-1,1,4,"0.5","maison","symboles")
```

Pour les effacer, même procédure que pour les disques
```
efface(evt,["maison"],"symboles")
```

Affichage de camemberts

Déclarer les objets à cloner : des objets 'path' où nous utiliserons l'arc de cercle.
Il faut déclarer autant d'objets que de parts de camemberts ….leur id doit être XYZ1 XYZ2 ….

```
<g id="symboles" style="visibility:visible;stroke:gray;fill-opacity:0.5">
<path id="pie1" d="M0 0" style="fill:#cc9933" />
…….
</g>
```

Paramètres : evt
reference (tableau des données à représenter en proportion par rapport à la dernière valeur)
Le nombre de parts de camembert est égal au nombre d'éléments de reference. La dernière part représente le complément à la valeur de référence.
rayon (du disque) nom (id générique) nœud (du DOM où créer les symboles)
Les couleurs sont données par l'objet cloné …
On pourrait les envoyer comme paramètre.

```
function camembert(evt,reference,rayon,nom,noeud)
{nb_parts=reference.length;
var svgdoc=evt.getTarget().getOwnerDocument();
var  contents = svgdoc.getElementById(noeud);
for(i=1;i<=nb_regions;i++)
{a='';xA=rayon+donnees[i][1];yA=donnees[i][2];angle=0;
for (k=1;k<nb_parts;k++)
{angle1=360*donnees[i][reference[k-1]]/donnees[i][reference[nb_parts-
1]];angle=angle+angle1;
xB=donnees[i][1]+rayon*Math.cos(angle*Math.PI/180);
yB=donnees[i][2]-rayon*Math.sin(angle*Math.PI/180);
if (angle1<=180)
{a=a+'M'+xA+' '+yA+' A '+rayon.toString()+' '+rayon.toString()+' 0 0 0 '+xB+'
'+yB+' L'+donnees[i][1].toString()+","+donnees[i][2].toString()+' z'} else
{a=a+'M'+xA+' '+yA+' A '+rayon.toString()+' '+rayon.toString()+' 0 1 0 '+xB+'
'+yB+' L'+donnees[i][1].toString()+","+donnees[i][2].toString()+' z'};
var target1=svgdoc.getElementById(nom+k.toString());
var newnode = target1.cloneNode(false);
etiq=nom+k.toString()+i.toString();newnode.setAttribute ('id',
etiq);newnode.setAttribute("d",a);
newnode = contents.appendChild (newnode);xA=xB;yA=yB;a=''};
angle1=360-angle;
xB=Math.round(donnees[i][1]+rayon);yB=Math.round(donnees[i][2]);
if (angle1<=180)
{a=a+'M '+xA+' '+yA+' A '+rayon.toString()+' '+rayon.toString()+' 0 0 0 '
+xB+' '+yB+' L'+donnees[i][1].toString()+","+donnees[i][2].toString()+' z'}
else
{a=a+'M '+xA+' '+yA+' A '+rayon.toString()+' '+rayon.toString()+' 0 1 0 '
+xB+' '+yB+' L'+donnees[i][1].toString()+","+donnees[i][2].toString()+' z'};
var target1=svgdoc.getElementById(nom+nb_parts.toString());
```

```
var newnode = target1.cloneNode(false);
etiq=nom+nb_parts.toString()+i.toString();newnode.setAttribute ('id',
etiq);newnode.setAttribute("d",a);
newnode = contents.appendChild (newnode)}}
```

Exemple d'appel (proportion des noirs, asiatiques et hispaniques par état aux USA)
Le nombre de noirs est la donnée 5, les hispaniques la donnée 7, les asiatiques la donnée 6 et la population totale de l'état est la donnée 3.
Nous aurons 4 parts de camembert, noirs, hispaniques, asiatiques et le reste de la population.

```
camembert(evt,[5,7,6,3],15,"pie","symboles")}}
```

Pour effacer ces camemberts, il faut effacer les 4 objets

```
efface(evt,["pie1","pie2","pie3","pie4"],"symboles")
```

Affichage de barres d'évolution en pourcentage

Déclarer les objets à cloner : un objet 'path' et un objet 'rect'

```
<g id="barres" style="visibility:visible;stroke:gray;fill-opacity:0.5">
<rect id="barre" x="0" x="0" width="0" height="0" style="fill:#cc9933" />
<path id="trait" d="M0 0" style="fill:#cc9933" />
</g>
```

Paramètres : evt col1 (donnée nouvelle) col2 (ancienne valeur) largeur (de la barre) haut (hauteur maximale) couleur (tableau de 2 couleurs, la première pour une augmentation, la seconde pour une diminution) opa (opacité pour la barre) nom (tableau des deux id génériques pour la barre et le trait) nœud (du DOM où créer les symboles)

```
function barres(evt,col1,col2,large,haut,couleur,opa,nom,noeud)
{ var svgdoc=evt.getTarget().getOwnerDocument();
var target1=svgdoc.getElementById(nom[0]);
var target2=svgdoc.getElementById(nom[1]);
var  contents = svgdoc.getElementById (noeud);
maxi=Math.abs((donnees[1][col1]-donnees[1][col2])/donnees[1][col2]);
for (i=2;i<=nb_regions;i++)
{if (Math.abs((donnees[i][col1]-donnees[i][col2])/donnees[i][col2])>maxi)
{maxi=Math.abs((donnees[i][col1]-donnees[i][col2])/donnees[i][col2])}};
for (i=1;i<=nb_regions;i++)
{var newnode = target1.cloneNode(false);
etiq=nom[0]+i.toString();
newnode.setAttribute ('id', etiq);
newnode.setAttribute("x",Math.round(donnees[i][1]-large/2).toString());
newnode.setAttribute("width",large.toString());
hauteur=
Math.round(haut*(donnees[i][col1]-donnees[i][col2])(maxi*donnees[i][col2]));
if (hauteur<0) {newnode.setAttribute("y",donnees[i][2].toString());
newnode.setAttribute("height",Math.abs(hauteur).toString());
newnode.getStyle().setProperty("fill",couleur[1])}
else
{newnode.setAttribute("y",Math.round(donnees[i][2]-hauteur).toString());
newnode.setAttribute("height",Math.abs(hauteur).toString());
newnode.getStyle().setProperty("fill",couleur[0])};
newnode.getStyle().setProperty("fill-opacity",opa);
newnode = contents.appendChild (newnode);
var newnode=target2.cloneNode(false);
etiq=nom[1]+i.toString();
newnode.setAttribute ('id', etiq);
```

```
chaine="M"+Math.round(donnees[i][1]-large).toString()+"
"+donnees[i][2].toString()
+" "+Math.round(donnees[i][1]+large).toString()+" "+donnees[i][2].toString();
newnode.setAttribute("d",chaine);
newnode = contents.appendChild (newnode)}}
```

Exemple d'appel (évolution de la population française entre 1990 et 1999 par région)
La population en 1999 est la donnée 3, la population en 1990 la donnée 4
Nous aurons une barre rouge au-dessus du trait si la population a augmenté et une barre bleue en dessous si elle a diminué

```
barres(evt,3,4,300,300,["red","blue"],"0.5",["barre","trait"],"barres");
```

Pour effacer ces barres, il faut effacer les 2 objets
```
effaceplus(evt,["barre","trait"],"barres")
```

Affichage d'histogrammes

Déclarer les objets à cloner : des objets 'rect'
Il faut déclarer autant d'objets que de barres d'histogramme….leur id doit être XYZ1 XYZ2 ….

```
<g id="histo" style="visibility:visible;stroke:gray;fill-opacity:0.5">
<rect id="histo1" x="0" y="0" width="0" height="0" style="fill:#cc9933" />
…….
</g>
```

Paramètres : evt
reference (tableau des données à représenter en proportion par rapport à la dernière valeur)
Le nombre de barres est égal au nombre d'éléments de reference moins un.
La dernière valeur représente la valeur de référence. (n'est pas prise en compte si elle vaut –1)
large (des barres) haut (hauteur maximale des barres) couleur (tableau des couleurs)
nom (Tableau des id génériques) nœud (du DOM où créer les symboles)

```
function histogramme(evt,reference,large,haut,couleur,opa,nom,noeud)
{nb_histo=reference.length-1;
var  contents = svgdoc.getElementById(noeud);maxi=0;
for (i=1;i<=nb_regions;i++)
{for (j=0;j<nb_histo;j++)
{if (reference[nb_histo]==-1)
{valeur=Math.abs(donnees[i][reference[j]])}
else
{valeur=Math.abs(donnees[i][reference[j]]/donnees[i][reference[nb_histo]])};
if (valeur>maxi) {maxi=valeur}}};
for(i=1;i<=nb_regions;i++)
{for (j=0;j<nb_histo;j++)
{var target1=svgdoc.getElementById(nom[j]);
var newnode = target1.cloneNode(false);
etiq=nom[j]+i.toString();
newnode.setAttribute ('id', etiq);
newnode.setAttribute("x",Math.round(donnees[i][1]+(2*j-
nb_histo)*large/2).toString());
newnode.setAttribute("width",large.toString());
if (reference[nb_histo]==-1)
{valeur=Math.abs(donnees[i][reference[j]])}
else
{valeur=Math.abs(donnees[i][reference[j]]/donnees[i][reference[nb_histo]])};
hauteur=Math.round(haut*valeur/maxi);
if (hauteur>=0)
```

```
{newnode.setAttribute("y",Math.round(donnees[i][2]-hauteur).toString())}
else {newnode.setAttribute("y",Math.round(donnees[i][2]).toString())};
newnode.setAttribute("height",Math.abs(hauteur).toString());
newnode.getStyle().setProperty("fill",couleur[j]);
newnode = contents.appendChild (newnode)}}}
```

Exemple d'appel (proportion des noirs, asiatiques et hispaniques par état aux USA)
Le nombre de noirs est la donnée 5, les hispaniques la donnée 7, les asiatiques la donnée 6 et la population totale de l'état est la donnée 3.
Nous aurons 3 barres, noirs, hispaniques et asiatiques par rapport à la population totale.

```
histogramme(evt,[5,7,6,3],15,50,["maroon","red","yellow"],["histo1","histo2",
"histo3"],"histo")}}
```

Pour effacer ces histogrammes, il faut effacer les 3 objets
```
efface(evt,["histo1","histo2","histo3"],"histo")
```

Utilisation de cette librairie

Si nous avons ces procédures dans un fichier carto.js
Pour charger ces procédures depuis le fichier svg :
Mettre un objet SVG script
```
<script xlink:href="carto.js" />
```

Nous pouvons aussi mettre les données dans un fichier texte data.js
```
<script xlink:href="data.js" />
```

Un exemple complet : la population des USA
En-tête du fichier SVG
```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20000303 Stylable//EN"
"http://www.w3.org/TR/2000/03/WD-SVG-20000303/DTD/svg-20000303-stylable.dtd">
```

Début des objets SVG, définition des dimensions de la carte
```
<svg xml:space="preserve" width="800" height="550" >
```

Appel aux fichier de procédures et aux données
```
<script xlink:href="carto.js" />
<script xlink:href="data.js" />
```

Début du script propre à cet exemple
```
<script><![CDATA[
```

Variables globales indispensables aux procédures de carto.js
```
var nb_regions=51 ;
```

Autres variables globales
```
var legende="",num_choisi=0,cible0="reg00",old_style="";
```
Exemple de fonction utilisant au mieux les procédures
```
function poptot(evt)
{if (legende!="poptot")
{if (legende=="pie") {efface(evt,["pie1","pie2","pie3","pie4"],"pie")};
if (legende=="electeurs") {efface(evt,["cercle"],"cercles")};
if (legende=="densite") {efface(evt,["maison"],"symboles")};legende="poptot";
titre(evt,"titre","Population totale (1999)");
legendage(evt,"rectleg",["#cccccc","#cccccc","#cccccc"],["1.0","0.5","0.1"]);
texte_legende(evt,"texleg",[1,0,0,0,0,0]);
```

```
degrades(evt,"reg",3,-1,[0,1500000,8000000,50000000] ,
["#cccccc","#cccccc","#cccccc"], ["0.1","0.5","1.0"])}}
```

Fin du script
```
]]></script>
```

Les objets graphiques et la gestion des événements souris sur ces objets
Les régions
```
<g onmousemove="nommer(evt)" style="fill:#B7DDC8; stroke:black; stroke-
width:1;fill-opacity:1.0">
```
La région 00, un rectangle couvrant l'ensemble de la carte, dessiné en premier
```
<rect id="reg00" x="0" y="0" width="800" height="550" style="fill:white"/>
```
Les états
```
<path id="reg01" style="fill:#B7DDC8" d="M131 483 ……..z"/>
…….
</g>
```

Les rectangles de légende (nommés obligatoirement XYZ1 XYZ2 et XYZ3)
```
<rect id="rectleg1" x="500" y="75" width="60" height="15"
style="stroke:black;fill:none;fill-opacity:1"/>
```

Les textes de légende (nommés obligatoirement SQWS1 SQWS2 …..) leur nombre : nb_legendes
```
<text id="texleg1" x="530" y="70" style="visibility:hidden;text-
anchor:middle;font-size:8pt;font-family:Times Roman;fill:black">+ 8Mhab<tspan
x="590">1,5-8 Mhab</tspan><tspan x="650">-1,5Mhab</tspan></text>
```

Un exemple de 2 mis en exposant
```
<text id="texleg2" x="530" y="70" style="visibility:hidden;text-
anchor:middle;font-size:8pt;font-family:Times
Roman;fill:black">+100h/km<tspan dy="-4" style="font-size:6pt">2</tspan>
<tspan x="590" dy="4" style="font-size:8pt">20-99h/km<tspan dy="-4"
style="font-size:6pt">2</tspan></tspan><tspan x="650" dy="4" style="font-
size:8pt">-20h/km<tspan dy="-4" style="font-
size:6pt">2</tspan></tspan></text>
```

Le titre
```
<text id="titre" x="590" y="50" style="visibility:visible;text-
anchor:middle;font-size:8pt;font-family:Times Roman;fill:black">Les états des
Etats Unis</text>
```

L'objet à cloner pour les camemberts
```
<g id="symboles" style="visibility:visible;stroke:gray;fill-opacity:0.5">
<path id="pie1" d="M0 0" style="fill:#cc9933" />
<path id="pie2" d="M0 0" style="fill:#ff9900" />
<path id="pie3" d="M0 0" style="fill:#ffff00" />
<path id="pie4" d="M0 0" style="fill:#ffffff" />
</g>
```

Un texte, en cliquant sur celui ci, nous aurons la population des états
```
<g onclick="poptot(evt)">
<rect id="legende1" x="0" y="0" width="100" height="15"
style="fill:#cccccc;fill-opacity:0.5"/>
<text id="nom1" x="50" y="10" style="text-anchor:middle;font-size:8pt;font-
family:Times Roman;fill:black">Population des Etats</text>
</g>
```

La fin du fichier
```
</svg>
```