

# Cartographie 3

Éric Guichard

14 mars 2022

## Possibles

- QGis, logiciel de cartographie
  - Philcarto, pour Windows
  - Illustrator, Inkscape.
  - Mapinfo, arcview, arcGis.
  - Artique : <https://www.artique.com>, etc.
  - Magrit : <http://178.128.206.148/modules>.
  - Géoportail : <https://www.geoportail.gouv.fr>.
  - python.
  - Perl, (D3.)js, etc. StrawberryPerl sur Windows.
- Rappel du projet : acquérir une **culture numérique**.

# Formats

## Privateurs ou binaires (illisibles)

- shp
- propres aux logiciels précités.

## Ouverts (lisibles)

- txt, csv, etc.
- svg,
- de la forme tikz
- etc. Garder une cohérence : ce qui relèvera d'un identifiant, de coordonnées...

Ces formats sont aussi associés à des fonctions (Béziers plutôt que polygones, etc.).

## Simplicité ?

- Certaines solutions d'une simplicité déconcertante
- *a priori* seulement !
- Le diable se cache dans les détails
- et dans le *design*,
- sans compter la question des projections.

**Dans tous les cas** (sauf cartes esthétiques : non automatisées) :

- articuler (joindre) au moins deux fichiers : un pour les données (pop d'un dpt, etc.), un pour le fond de carte (ou l'un de ses calques) ;
- beaucoup de traitements, d'essais, d'erreurs ;
- construction d'une culture graphique ;
- et d'une culture épistémologique.

## Exemple SVG

Lisible, orienté web, interactif.

- en-tête,
- calques : `<g>... </g>`
- fin : `</svg>`
- encapsulable en du html.

**Attention** à la syntaxe, exigeante : c'est du xml.

## En-tête

On le recopie.

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>  
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"  
"http://www.w3.org/TR/SVG/DTD/svg10.dtd">  
<svg xmlns="http://www.w3.org/2000/svg"  
xmlns:xlink="http://www.w3.org/1999/xlink"  
viewBox=" 0 0 3000 3000" width="100%" height="100%" >
```

La viewBox (et non pas viewbox!) est modulable.

De même pour les proportions et les options (homothétie, rotations, etc.).

## Extrait d'un calque de surfaces

```
<g id ="DepSurface">  
...  
<path id="D14" fill="blue" stroke-width="1" stroke="rgb(0,0,0)"  
d="M1174 838 L1176 833 L1173 832 L1171 828 L1169 829 L1164 827 ... z"/>  
<path ...  
</g>
```

Commenter oralement :

- M, L, etc., id.
- z"/> ou "> </path>
- On n'oublie pas de « fermer » le polygone.

## Cadre

```
<g id ="Cadre" fill="rgb(0, 0, 250)" fill-opacity="0.02"  
stroke="rgb(0, 0, 220)" stroke-width="1">  
<rect x="5" y="5" width="2990" height="2990" />  
</g>
```

- Beaucoup de coordonnées cartésiennes.
- Attention au sens des axes  $x$  et  $y$ .



## Points et cercles

```
<g id="cercles">  
...  
<circle id="D95C" cx="1519.5" cy="824.5" r="20" stroke="red"  
stroke-width="2" fill="none" opacity="0.8"> </circle>  
<circle ...  
</g>
```

- Faire en sorte que l'*id* du point ait un lien avec celle de la surface associée.
- Points pour rappel : `r=0`, `stroke=""`, etc.
- `opacity` non nécessaire : ici pour rappel des possibles.

## D'abord voir le svg

Dossier <http://barthes.enssib.fr/cartom1>

- De base : FranceFondDpt-cours.svg
- Avec des centres : France4ok.svg

Expliciter cette construction artificielle des centres.

### Opérations élémentaires :

- Changer la couleur des départements, des cercles, leur rayon.
- Déplacer un centre.

## Un script élémentaire

Si le numéro du dpt est supérieur à 50, le colorier en vert.

**Attention** : comme pour les tutoriels en ligne, on fait un copier/coller sans tout comprendre. Le script est commenté en cours.

```
open (F,"France4ok.svg");
open (F2,">Francebase.svg");
while (<F>)
{
  if (/<path id="\(D\d+)\\"/)
  {
    $dpt=$1;
    $dpt=~s/D//g;
    $dpt+=0;
    if ($dpt>50)
      { s/blue/green/;}
  }
  print F2;
}
close (F); close (F2);
```

## Un script plus subtil

Le fond de carte (`France4ok.svg`) et le fichier de *données* : *mortssimples*, qui a l'allure suivante :

D01	0.47
D02	0.91
D03	0.69
...	

Morts *covid* du département pour 10 000 habitants (10 mars 2022).  
Ci après le script sur deux diapositives.

## Script partie 1

```
open (M,"mortssimples"); #données
open (F,"France4ok.svg"); # fond de carte
open (F2,">Francemorts.svg"); #résultat
while (<M>)
  {
  chop;
  ($dpt,$pcmorts)=split(/\t/);
  $covid{$dpt}=$pcmorts;
  }
close (M);
# On a mémorisé les informations de chaque dpt
```

On va s'occuper des changements de couleur des dpts.

## Script partie 2

```
while (<F>)
{
  if (/<path id="(D\d+)\\"/)
  {
    $dpt=$1;
    if ($covid{$dpt}<.65) # Premier seuil
      { s/blue/white/;}
    elsif ($covid{$dpt}<.85)
      { s/blue/orange/; }
    else
      { s/blue/purple/; }
  }
  print F2;
}
close (F); close (M2);
```

## Généralisations

- Trouver les populations des départements et les intégrer dans les cercles.
- Passer des morts aux confirmés.
- Choisir d'autres dates.

### Synthèse

- Travail sur des sources variées : trier, recoder, adapter, normaliser, etc.
- Beaucoup de *manipulations* : d'où les scripts, programmes, bibliothèques et... travaux collectifs.
- Force des outils/commandes Linux ou analogues.
- `convert` `sgv` vers `JPG` fonctionne !
- Un lent apprentissage : culture numérique, plus en rapport avec les langues et les mots qu'avec les maths.