

DIFFICULTÉS ★★☆☆ **PRÉPARATION** ★★☆☆ **POSITION DU TEMPLATE:** RIGHT

DIFFICULTÉS ★★☆☆

PRÉPARATION ★★☆☆

POSITION DU TEMPLATE: RIGHT

NOM DE L'ENTITÉ GÉOGRAPHIQUE

Titre (texte)

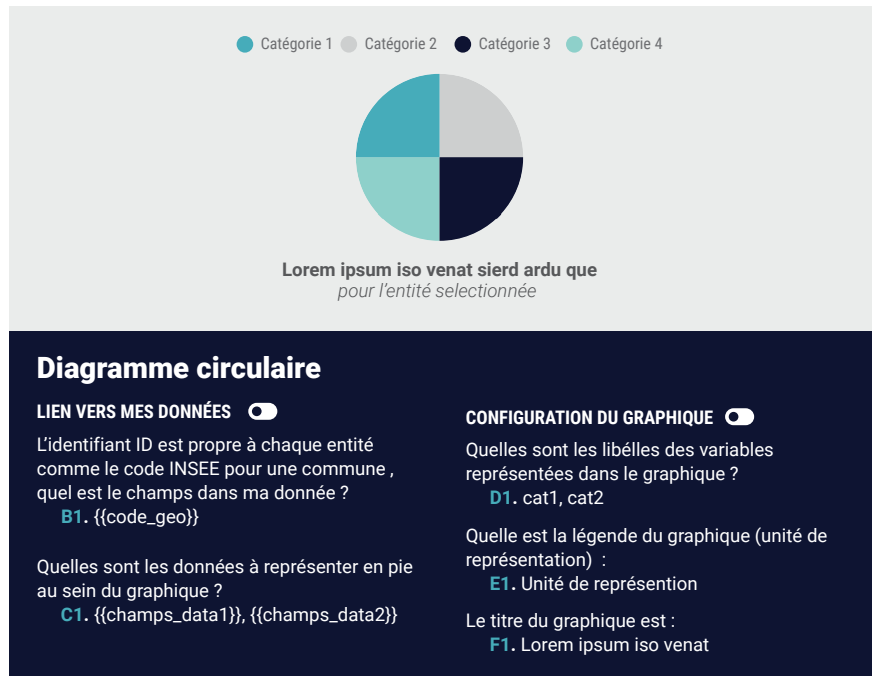
LIEN VERS MON TITRE ☒

Le titre est **propre** à chaque entité géographique, quel est le champs dans ma donnée ?

A1. {{nom_geographique}}

OPTIONS

POUR CHAQUE ÉLÉMENT, IL
EST NÉCESSAIRE DE CHOISIR
UNE OPTION PARMI CELLES
PROPOSÉES



- Ouvrir le fichier `template.graph_pie.mst` à l'aide d'un éditeur de texte (*Notepad++*)
- À la ligne 5, insérez l'option **A1** pour configurer le titre : `<h3 class="title-feature ">A1</h3>`
- À la ligne 10, insérez l'option **B1** pour configurer l'identifiant du graphique et injecter le code géographique de la donnée :
`<canvas class="chart" id="myChartPie1_B1" data-codegeo="B1"></canvas>`
 --> L'option B1 étant présente dans plusieurs partie du code, il est préférable d'effectuer un RECHERCHER/REPLACER en remplaçant code_geo par B1
- À la ligne 10, insérez tous les champs définis dans l'option **C1** pour ajouter les données au graphique :
`<canvas data-mychartPie1-1="{{C1_champsdata1}}" data-mychartPie1-2="{{C1_champsdata2}}"></canvas>`
 --> Ajoutez autant de data-mychartPie1-X que vos variables [CAT1, CAT2, CAT3, CAT4...] qui composent le graphique
- À la ligne 13, insérez l'option **F1** pour configurer le titre du graphique : `<div class="title-chart">F1</div>`
- À la ligne 95, insérez les libellés des variables définis dans l'option **D1** :
`var data = {myChartPie1_attr: {labels:["D1 CAT1","D1 CAT2"]}};`
 --> Ajoutez autant de libellés entre "" que vos variables [CAT1, CAT2...] qui composent le graphique
- À la ligne 95, insérez dans le datasets les données des variables définies dans l'option **C1** :
`var data = {myChartPie1_attr: data: {datasets:[{parseFloat("${myChartPie1_{{code_geo}}").attr("-data-mychartPie1-1")),parseFloat("${myChartPie1_{{code_geo}}").attr("data-mychartPie1-2"))}]}}};`
 --> Ajoutez autant de parseFloat("\${myChartPie1_{{code_geo}}").attr("data-mychartPie1-X") séparé par une virgule que vos variables [CAT1, CAT2, CAT3, CAT4...] qui composent le graphique
- À la ligne 110, insérez l'option **E1** pour configurer la légende du graphique : `label: "E1",`
- Pour personnaliser la couleur du template (titre), insérez le code #HEX de votre couleur à la ligne 26 :

```
root {
  --mycolor: #1D1B35;
}
```
- Pour personnaliser la couleur du graphique, insérez les codes RGBA de vos couleurs à partir de la ligne 113 :

```
datasets: [{
  backgroundColor: [
    "rgba(130,202,200,0.9)", //COULEUR DE LA CATÉGORIE 1 --> Ajoutez autant de couleurs que
    "rgba(29,28,51,0.9)" //COULEUR DE LA CATÉGORIE 2 vos variables [CAT1, CAT2, CAT3, CAT4...]
  ],
  hoverBackgroundColor: [
    "rgba(130,202,200,1)", //COULEUR DE LA CATÉGORIE 1 AU SURVOL
    "rgba(29,28,51,1)" //COULEUR DE LA CATÉGORIE 2 AU SURVOL
  ],
}]
```

LES ÉLÉMENTS DOIVENT ÊTRE PRÉSENTS DANS LA COUCHE GÉOGRAPHIQUE UTILISÉE EN LIEN DANS LE MVIEWER

TEMPLATE
TUTORIEL
5.2

DÉVELOPPER SES TEMPLATES



IMPORTANT !

Pour que le template fonctionne, pensez à paramétrer votre fichier de configuration `config.xml` comme ceci :

```
<layer id="xx" name="xx"
  infoformat="application/vnd.ogc.gml"
  ... >
  <template url="montemplate.mst"></template>
</layer>
```

+ NOTE

SI VOUS SOUHAITEZ DUPLIQUER LE GRAPHIQUE DANS LE TEMPLATE, VEAUX À MODIFIER LE NOM DES VARIABLES DU GRAPHIQUE CI-DESSOUS VIA UN RECHERCHER/EMPLACER :

```
myChartPie1_{{code_geo}}  
data-myChartPie1  
graph_myChartPie1  
myChartPie1_attr
```

POUR OBTENIR LES CODES COULEURS,
AIDEZ-VOUS DU **SELECTEUR DE COULEUR**
DISPONIBLE EN LIGNE
[https://htmlcolorcodes.com/fr/
selecteur-de-couleur/](https://htmlcolorcodes.com/fr/selecteur-de-couleur/)

