

► Bienvenue !

► Semaine 1 :  
Prise en main

► Semaine 2 :  
Modèle internet

► Semaine 3 :  
Modèle Pair-à-  
pair


▼ **Semaine 4 :**  
**Visualisation**  
**de données**

Présentation de la  
semaine


Visualisation de  
données

Jimbo capteur et  
visualisation

Quiz 04

Quiz Echéance le avril  
30, 2019 at 23:30 UTC 

**TP 04 : Évaluation  
par les pairs**

Travaux pratiques  
Echéance le avril 24,  
2019 at 23:00 UTC 

► La suite de  
l'aventure

► Corrigés des TP

► Bugs/Questions  
techniques FUN  
?

## TP 4 : ESPACE DE DÉPÔT DES TRAVAUX ET ÉVALUATION PAR LES PAIRS

### Statut

Vous avez terminé ce devoir. Visualisez votre note et les détails de votre évaluation.

▼ **Votre réponse**  **COMPLET**

▼ **Évaluer les pairs**  **4 TERMINÉ**

◀ **Votre note: 10 sur 10**

### La question pour cette section

À partir des consignes ci-dessus, vous devez vous inspirer des exemples de la semaine et du cours des semaines passées pour réaliser un programme Processing que vous viendrez coller ici.

Bon courage !

### Votre réponse

/\* TP n°4 du MOOC "Fabriquer un objet connecté"

Ce dernier TP est centré autour de Processing, qu'on utilise ici. Après avoir vu comment récupérer des données formatées en JSON sur le web (ici envoyées par des ESP32), on va décliner ici leur interprétation graphique.

Puisque pour les besoins du TP on n'utilise pas un vrai JSON variable, il faudrait changer manuellement l'URL, pour tester les deux modes d'affichage.

Or, Processing est assez puissant et permet des interactions avec son clavier par exemple.

Donc ici, la personne qui aura la gentillesse de me relire, n'aura pas à s'embêter à reprendre le code, il lui suffira d'appuyer sur n'importe quelle touche de son clavier pour tester les deux modes d'affichage (une pression sur le clavier = une commutation du mode d'affichage).

On verra également que Processing peut faire de l'animation.

\*/

## Jimbo

```
codes couleur)
int noir = 0;
int blanc = 255;
int vert = #8ae234;
int rouge = #ef2929;

// définition d'une variable pour l'URL
String valeurURL="good"; //on commencera donc le programme avec
l'URL "http://api.jimbo.wtf/luftdaten/good" (cf.infra)

// une petite surprise
int size = 20; //indice : cela concerne l'animation

// dans le setup on définit quelques fonctions de base
void setup(){
size(400, 300, P3D); // les deux premiers champs servent à déterminer
la taille de l'espace utilisé. Le paramètre P3D permet à Processing
d'utiliser un rendu 3D (https://processing.org/tutorials/p3d/). Ce sera
nécessaire pour animer le texte.
noStroke(); // ici j'enlève les contours des formes ( en l'occurrence les
ellipses) qui seront dessinées (surtout avec un rendu 3D couplé à une
requête internet qui entraine une perte de résolution graphique : vous
pouvez tester en enlevant cette ligne)
}

void draw(){
background(blanc); // je définis ici la couleur de fond

// on déclare une variable String qui viendra stocker l'url de la
ressource à atteindre
String url = "http://api.jimbo.wtf/luftdaten/"+valeurURL ;//String url
stocke donc une URL qui variera en fonction de valeurURL (rappel, on
commence avec "good", notamment pour les besoins de la fonction
GET)
// on analyse le fichier JSON reçu
JSONObject json = loadJSONObject(url);
// on extrait l'élément JSON appelé "pm2.5"
float pm = json.getFloat("pm2.5");

//si la qualité de l'air est inférieure à 25
if(pm < 25){ //c'est donc la variable pm, qui correspond à la donnée
"pm2.5" du JSON (cf. supra), qu'on compare à l'entier 25
fill(vert); // la forme qui suit sera de couleur verte
ellipse(110, 137, 40, 40);//un objet rond est dessiné. Les deux
premières coordonnées définissent la position du centre de ce rond
(par rapport au point d'origine situé en haut à gauche). Les deux
derniers champs définissent le rayon (en hauteur et en largeur, pour
dessiner des ellipses...)
```

affiché sera donc = pm). Les deux entiers suivants définissent la position du texte : 89 pixels à droite, 140 pixels en bas.  
fill(noir); // le prochain objet sera noir  
textSize(12); // il s'agit d'un texte dont je définis ici la taille (nécessaire en mode 3D, sinon facultatif)  
text("La qualité de l'air est bonne", 140, 140); // j'écris le texte demandé, en le positionnant selon les deux derniers entiers entre parenthèses.

//si la qualité de l'air est supérieur à 25  
if(pm >= 25){ //même principe que précédemment, on utilise la variable pm pour comparer à notre seuil de tolérance au particules fines qui est de 25  
fill(rouge); // le prochain objet sera rouge  
ellipse(100, 140, 40+(pm/2), 40+(pm/2)); //ici je change un peu la définition du cercle, c'est à dire que je fais évoluer la taille de son rayon en fonction du niveau de particules avec la formule 40[taille minimum du cercle]+(pm/2)[augmentation du rayon selon un ratio de pm divisé par 2 (pour pas que le rayon soit trop grand)].

fill(noir); //mon prochain texte sera noir  
textSize(size); //c'est ici qu'on rentre dans l'animation. En l'occurrence c'est la taille du texte suivant qu'on va faire varier selon size.  
text("Alerte particules fines", 145, 145, -size + size \* sin(radians(frameCount\*10))); //ici je définis le texte, sa position (145 et 145), puis son évolution grâce au rendu 3D. Alors je vous rassure, la formule n'est pas de moi, elle vient d'ici :  
<https://forum.processing.org/two/discussion/11806/text-scale-animation> (cf. le très court code de la 1ere réponse)

}  
}

// c'est ici qu'on facilite le test grâce à la fonction keyPressed() :  
[https://processing.org/reference/keyPressed\\_.html](https://processing.org/reference/keyPressed_.html)  
void keyPressed() {  
if (valeurURL == "good") { //si la valeur de l'URL fini par "good"  
valeurURL = "bad"; //alors on permute à "bad" lorsqu'on appuie sur le clavier. C'est pour ça qu'il est important de commencer ce programme avec une valeur de base définie (que j'ai choisi ="good"). C'est également nécessaire pour la fonction GET.  
}  
else {  
valeurURL = "good"; //dans tous les autres cas, c'est à dire quand valeurULR="bad", on permute sur "good".  
}  
}

/\*

Commentaire technique : je précise qu'appuyer sur une touche du clavier ne permute pas les modes d'affichages, il permute l'URL qui est requêtée. L'affichage change donc en fonction du JSON de cette dernière.

Commentaire : j'avoue être resté sur ma fin avec ce dernier TP qui consistait simplement à faire de la mise en page graphique. Alors pour combler ma frustration, j'en ai profité pour découvrir Processing. Et j'avoue m'être bien amusé. C'est finalement un outil de visualisation assez puissant.

Je vous recommande les liens suivants :

<https://fr.flossmanuals.net/processing/animation-dun-objet/>

ou encore le code de la 1ere réponse du lien suivant

[https://forum.processing.org/two/discussion/comment/80145/#Comment\\_80145](https://forum.processing.org/two/discussion/comment/80145/#Comment_80145) (qui montre qu'on peut aussi interagir avec la souris).

Bonne fin de MOOC

\*/

## Votre fichier chargé

## Évaluations de votre réponse

### ▼ Qualité du code

5 / 5 POINTS

#### • NOTE MÉDIANE DES PAIRS. - 5 POINTS

### Code fonctionnel et efficace ⓘ

PAIR 1 - CODE FONCTIONNEL ET EFFICACE

Tout fonctionne comme demandé avec les mots clés indiqués et le code est même allé plus loin (texte inclus dans le voyant, clignotement, etc)

PAIR 2 - CODE FONCTIONNEL ET EFFICACE

Le code ne compile pas, mais il y a un travail de recherche et une volonté d'aller plus loin que les consignes alors je valorise davantage cela que la fonctionnalité du code

PAIR 3 - CODE FONCTIONNEL ET EFFICACE

Pas mal du tout l'amélioration du code, je suis assez fan et en effet tu m'as permis d'être un peu plus curieux, je vais creuser tout ça! merci à toi!!

PAIR 4 - CODE FONCTIONNEL ET EFFICACE

Bravo, vous parvenez à faire mieux que ce qui a été demandé!

- NOTE MÉDIANE DES PAIRS. - 5 POINTS

### Code bien documenté ⓘ

PAIR 1 - CODE BIEN DOCUMENTÉ

Tout est bien expliqué et très bien structuré, notamment avec le recours à une fonction et à des effets 3D au niveau graphique et avec des noms de variables explicites pour les couleurs.

PAIR 2 - CODE BIEN DOCUMENTÉ

RAS

PAIR 3 - CODE BIEN DOCUMENTÉ

Super bien documenté!!

PAIR 4 - CODE BIEN DOCUMENTÉ

Excellent

#### ▼ Commentaires additionnels sur votre réponse

- PAIR 1

Animation graphique obtenue intéressante. Par contre, je n'ai pas réussi à faire passer le voyant du vert au rouge en cliquant sur la souris. Ce n'est pas grave car ce n'était pas demandé mais cela m'aurait intéressé. Au bout du compte, je ne vois pas bien pourquoi le sous-programme keypress() ne s'exécute pas.

PAIR 2

PAIR 3

super chouette, c'est agréable de voir qqn s'amuser comme ça!

PAIR 4

C'est mon premier MOOC, et j'apprécie beaucoup votre état d'esprit positif en partage des infos dans votre code. Merci pour votre enthousiasme.

#### ▼ Faire un retour sur l'évaluation par les pairs

##### Statut

Votre retour a été soumis. Il sera accessible à aux membres de l'équipe d'enseignement lorsqu'ils examineront les registres du cours.



[Aide et Contact](#)

[Politique de confidentialite](#)

[Conditions générales d'utilisation](#)

[Mentions légales](#)

