

[Fichier](#)[Modifier](#)[Sketch](#)[Dépanner](#)[Outils](#)[Aide](#)

sketch_220919a | Processing 4.0.1



Java ▼

sketch 220919a ▼

Introduction à Processing

<https://www.processing.org>

Septembre 2022



sketch_220919a

Historique

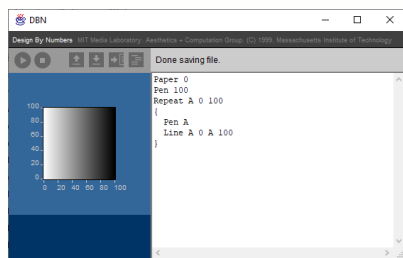
Design by Numbers

<http://dbn.media.mit.edu>

Date : 1999-2001

Lieu : MIT Media Lab

John Maeda



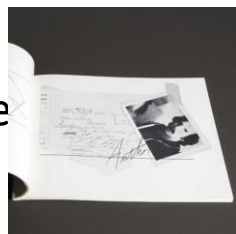
Visible Language Workshop

<http://museum.mit.edu/150/115>

Date : 1975

Lieu : MIT

Muriel Coope



Processing

<http://www.processing.org>

Date : Printemps 2001

Lieu : MIT Media Lab

Ben Fry / Casey Reas



p5.js



Processing 4

Wiring

<http://wiring.org.co>

Date : 2003

Lieu : IDII

Hernando Barragán



Arduino

<http://www.arduino.cc>

Date : 2005

Lieu : IDII

Massimo Banzi





Qu'est ce que Processing ?

Processing : « *an electronic sketchbook for developing ideas* »,

« *language that was created to develop visually oriented applications with an emphasis on animation and providing users with instant feedback through interaction* »

Processing est en réalité **Java** [et donc on peut développer facilement pour Android] mais il existe d'autres implémentations [en javascript P5.js, *Python*, ...] respectant [à peu près] la même syntaxe

Il s'exécute sur MacOS, windows, Linux [dont raspberry 😊] en 64 [ou 32] bits




sketch 220919a

Download / Processing.org

processing.org/download

Download

Processing is available for Linux, Mac, and Windows. Select your choice to download the software below.



Processing

4.0.1 (August 9, 2022)

MacOS (Intel 64-bit)

MacOS (Apple Silicon)

Windows (Intel 64-bit)

Linux (Intel 64-bit)

Linux (Raspberry Pi 32-bit)

Linux (Raspberry Pi 64-bit)

GitHub Report Bugs Wiki Troubleshooting

Stable Releases

4.0.1 (August 9, 2022) [MacOS \(Intel 64-bit\)](#) / [Linux \(Intel 64-bit\)](#) / [MacOS \(Apple Silicon\)](#) / [Linux \(Raspberry Pi 32-bit\)](#) / [Windows \(Intel 64-bit\)](#) / [Linux \(Raspberry Pi 64-bit\)](#)

Donate



sketch_220919a

Des ressources

- **Hello Processing :**
<https://hello.processing.org/editor>
- **Référence du langage :**
<https://processing.org/reference>
- **The Coding Train** [chaîne Youtube & Discord de Daniel Shiffman] : <https://thecodingtrain.com>
- **Fiches :** <https://github.com/truillet/processing>



Qu'est ce que Processing ?

1
2
3
4 Arduino a été développé à la base à partir de
5 Processing.

6 Il est donc simple de passer de l'un à l'autre pour
7 prototyper des systèmes interactifs logiciels et
8 matériels :

- 9 • **Processing.org** pour la partie interface et visualisation,
- 10 • **arduino** pour la partie matérielle et gestion de capteurs.

11
12
13
14
15
16
17
18 La base du programme Processing est le « *sketch* »
19 [programme, prototype]

20 L'extension est le « **.pde** »
21
22
23
24
25
26
27



Java ▼

Editeur

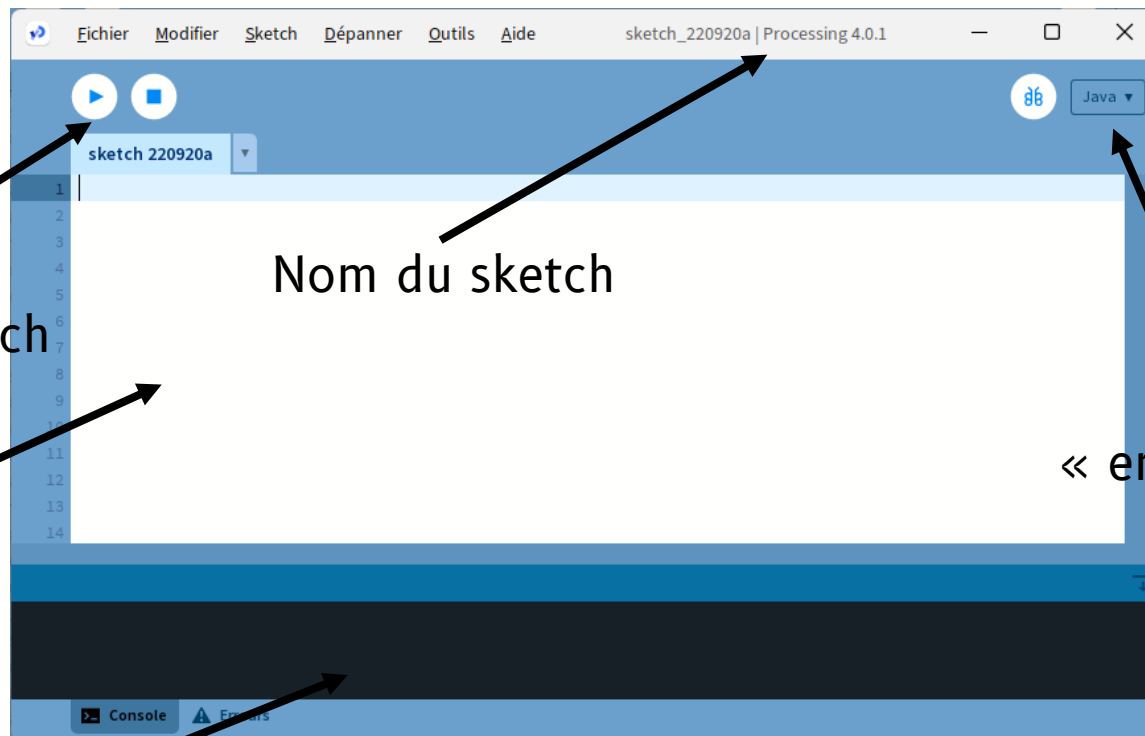
Lancement du sketch

code

Sorties textuelles
+ erreurs

Nom du sketch

Mode
« en cours »

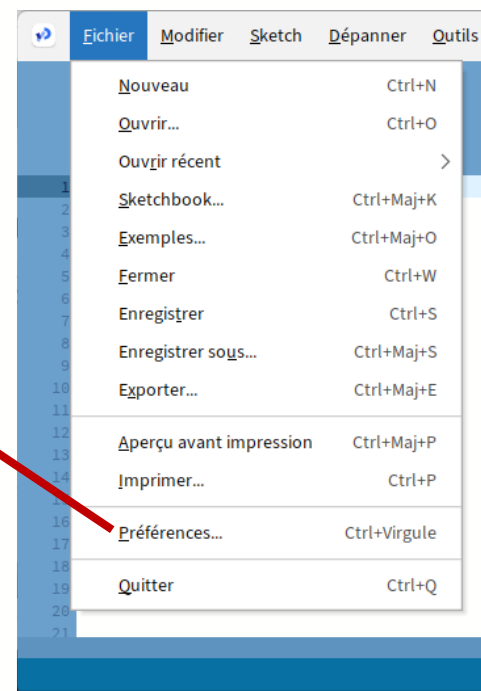
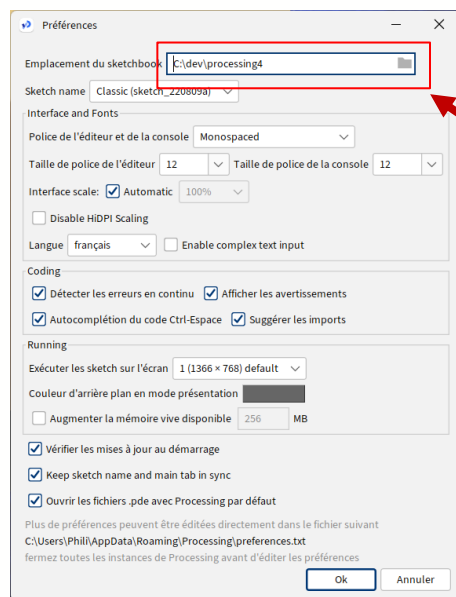




sketch_220919a

Structure

- Les « sketches » [programmes] sont localisés dans le répertoire « *préférences* »





sketch_220919a ▾

Structure

options



Préférences

Emplacement du sketchbook C:\dev\processing4

Sketch name Classic (sketch_220809a) ▾

Interface and Fonts

Police de l'éditeur et de la console Monospaced ▾

Taille de police de l'éditeur 12 ▾ Taille de police de la console 12 ▾

Interface scale: ☒ Automatic 100% ▾

☐ Disable HiDPI Scaling

Langue français ▾ ☐ Enable complex text input

Coding

☒ Détecter les erreurs en continu ☒ Afficher les avertissements

☒ Autocomplétion du code Ctrl-Espace ☒ Suggérer les imports

Running

Exécuter les sketch sur l'écran 1 (1366 × 768) default ▾

Couleur d'arrière plan en mode présentation

☐ Augmenter la mémoire vive disponible 256 MB

☒ Vérifier les mises à jour au démarrage

☒ Keep sketch name and main tab in sync

☒ Ouvrir les fichiers .pde avec Processing par défaut

Plus de préférences peuvent être éditées directement dans le fichier suivant
C:\Users\Phil\AppData\Roaming\Processing\preferences.txt
fermez toutes les instances de Processing avant d'éditer les préférences

Ok Annuler



Java ▼

Structure

- un sketch est **composé de** :
 - Au moins un fichier « **.pde** » [cela peut être plus - un par classe objet-].
Le fichier principal doit avoir le même nom que le répertoire du sketch

ELIPSE (C:) > dev > processing3 > boutons_img >

Nom

Modifié le

Type

Taille

code

05/12/2015 18:02

Dossier de fichiers

data

06/12/2015 21:18

Dossier de fichiers

boutons_img.pde

06/12/2015 21:47

Processing Source...

1 Ko

Button.pde

06/12/2015 21:47

Processing Source...

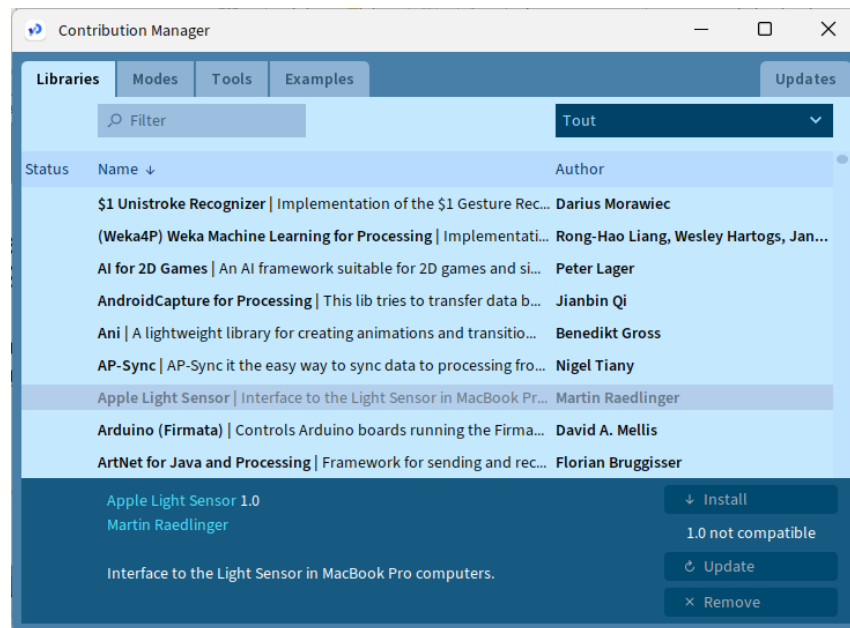
3 Ko



Structure

- Et de **librairies externes** [Outils | Ajouter un Outil | onglet *Libraries*]

Certaines sont très utiles
comme **video** pour gérer
la webcam





sketch_220919a

Processing et IntelliJIDEA



- <https://github.com/ctruiet/ProcessingOnIntelliJIDEA>



Java ▼

sketch_220919a ▼

Trois fonctions de base à utiliser

- **settings** : exécuté avant setup pour choisir le rendu graphique, ... [utile notamment avec Eclipse & IntelliJ]
- **setup** : exécuté une seule fois au démarrage – permet d'initialiser les variables du programme
- **draw** : c'est la boucle de traitement et d'affichage exécutée « à l'infini » [*mainloop*]

```
void setup()  
{  
  size(200,200);  
  background(102);  
}
```

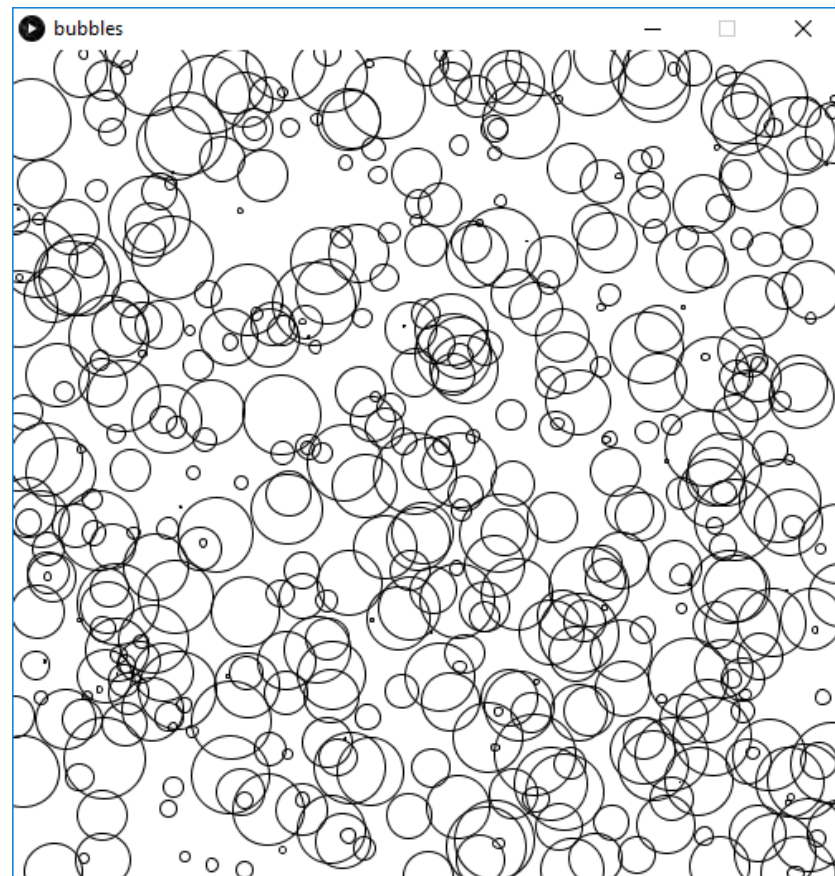


sketch_220919a

Un premier exemple

```
bubbles | Processing 3.3
Fichier Modifier Sketch Dépanner Outils Aide

bubbles
2 // variables du sketch (pseudos globales)
3 float x,y, diametre;
4
5 // setup n'est exécutée qu'une fois
6 void setup() {
7   size(500,500); // définit la taille de la fenêtre
8   background(255); // définit le fond de la fenêtre dessiné une fois (blanc)
9   smooth(); // anti aliasing
10  noFill(); // pas de remplissage des formes
11 }
12
13 // draw est exécutée à l'infini
14 void draw() {
15   x = random(width); // tiré au hasard entre 0 et la largeur de la fenêtre
16   y = random(height); // tiré au hasard entre 0 et la hauteur de la fenêtre
17   diametre = random(50);
18
19   // dessine un cercle - primitive pré-définie
20   ellipse(x,y, diametre, diametre);
21 }
22
23
24
25
26
27
28
29
30
L'enregistrement est terminé.
Console Erreurs
```

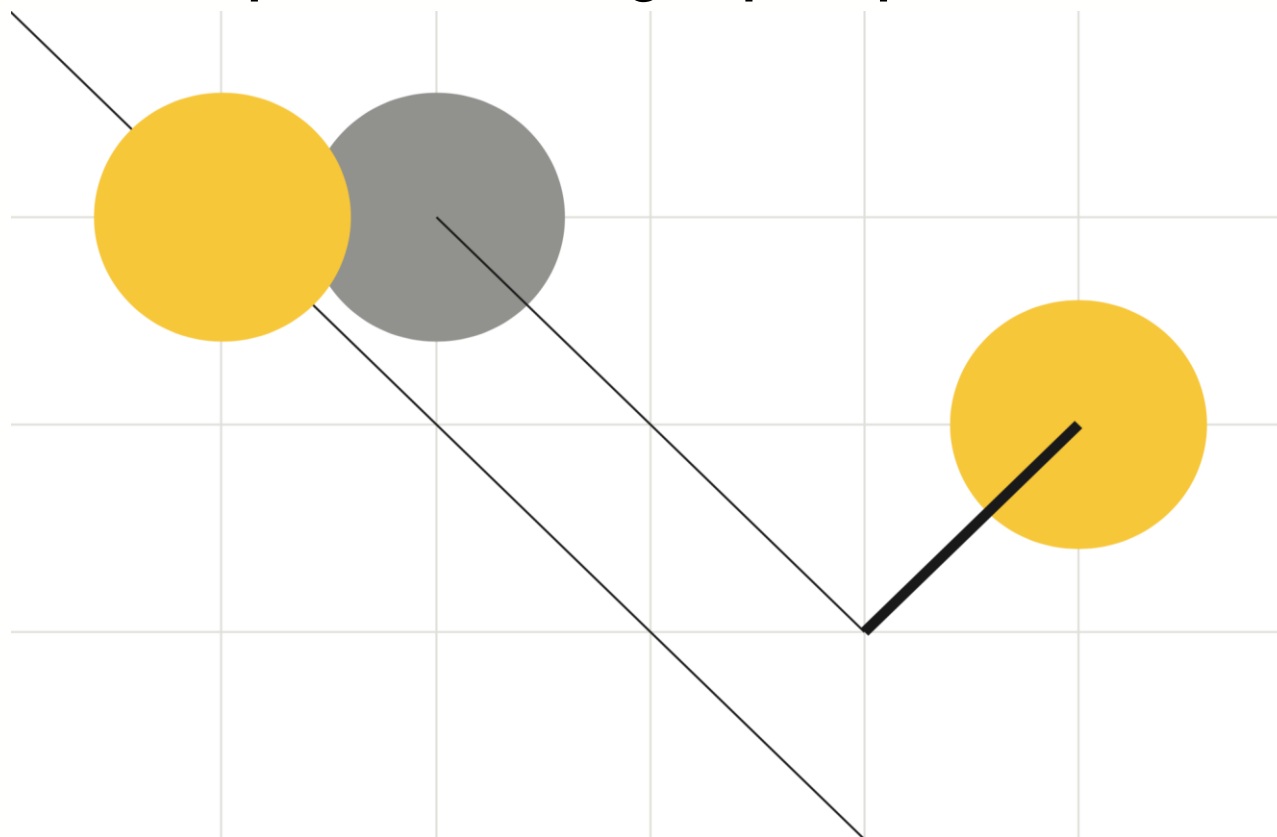




Sketch 220919a

Des exercices

- Utiliser des primitives graphiques





Java ▼

Sketch 220919a

Des exercices

- Utiliser des primitives graphiques
- Utiliser les événements clavier/souris



Java ▼

sketch_220919a ▼

Événements

- Les différents événements seront traités au travers de fonctions que vous aurez à écrire
 - **Souris** : `mousePressed()`, `mouseReleased()`, `mouseMoved()`, `mouseClicked()`, `mouseDragged()`
 - **Clavier** : `keyPressed()`, `keyReleased()`
 - **Vidéo** : `movieEvent()`
 - **Webcam** : `CaptureEvent()`
 - **Liaison Série** : `serialEvent()`



Java ▼

Sketch 220919a

Des exercices

- Utiliser des primitives graphiques
- Utiliser les événements clavier/souris
- Utiliser des polices de caractères/des images ...
- Utiliser la webcam
- *Récupérer et envoyer des données de/vers arduino*