



sketch 220919a ▼

Introduction à Processing

<https://www.processing.org>

Février 2023



Fichier

Modifier

Sketch

Dépanner

Outils

Aide

sketch_220919a | Processing 4.0.1



Java ▼

sketch_220919a

Historique

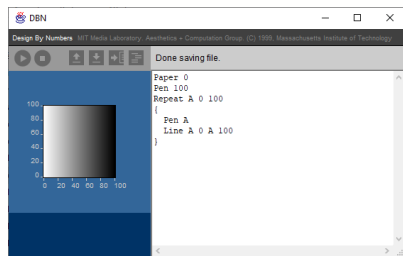
Design by Numbers

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

Date : 1999-2001

Lieu : MIT Media Lab

John Maeda



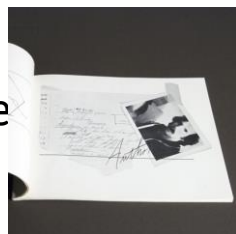
Visible Language Workshop

<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 2



Java ▼

Download / Processing.org

processing.org/download

Processing Foundation

Processing

p5.js

Processing Android

Processing Python

Processing

Download

Documentation

Learn

Teach


About

Donate

Search

Download

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



Processing

4.1.2 (January 16, 2023)

Linux (Raspberry Pi 32-bit)	Linux (Raspberry Pi 64-bit)
Linux (Intel 64-bit)	MacOS (Apple Silicon)
MacOS (Intel 64-bit)	Windows (Intel 64-bit)

GitHub

Report Bugs

Wiki

Troubleshooting

Stable Releases

4.1.2 (January 16, 2023) [Linux \(Raspberry Pi 32-bit\)](#) / [Linux \(Raspberry Pi 64-bit\)](#) / [Linux \(Intel 64-bit\)](#) / [MacOS \(Apple Silicon\)](#) / [MacOS \(Intel 64-bit\)](#) / [Windows \(Intel 64-bit\)](#)

3.5.4 (January 17, 2020) [MacOS](#) / [Linux \(Intel 64-bit\)](#) / [Windows \(64-bit\)](#) / [Windows \(32-bit\)](#)

We need your help!



Help us continue with your generosity!

Donate



Java ▼

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 ?

arduino a été développé à partir de Processing.
Il est donc simple de passer de l'un à l'autre pour
prototyper des systèmes interactifs logiciels et
matériels :

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

La base du programme Processing est le « *sketch* »
[programme, prototype]

L'extension est le « **.pde** »



Java ▼

Editeur

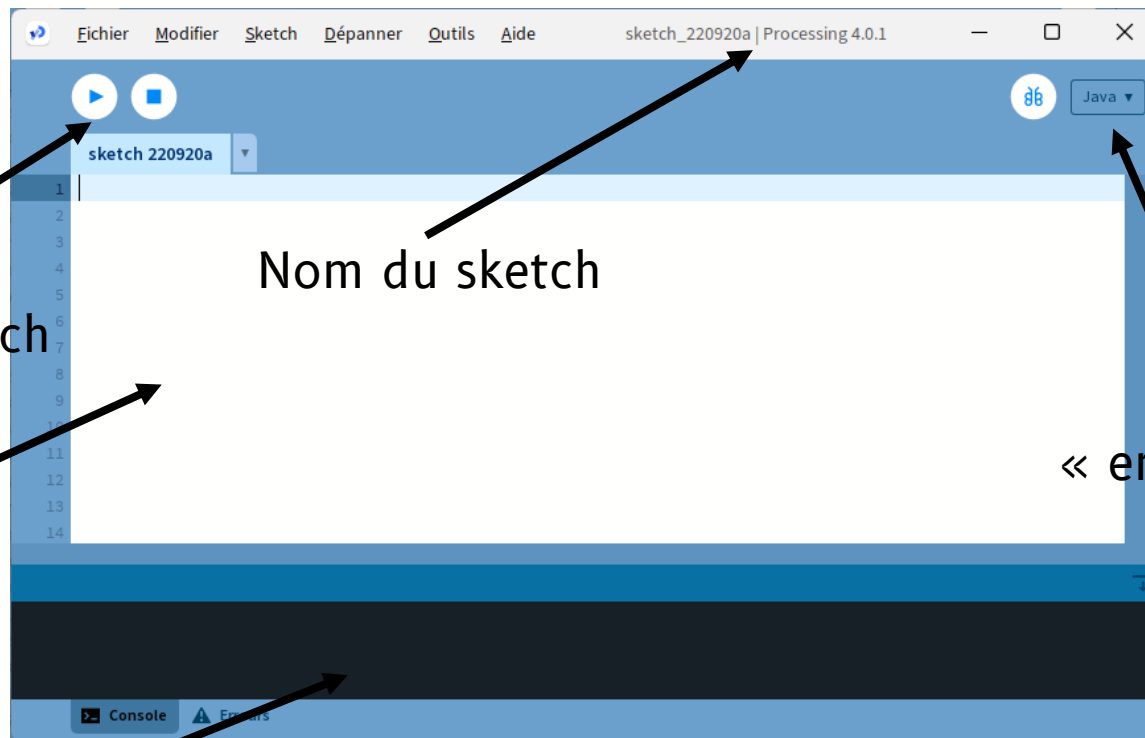
Lancement du sketch

code

Sorties textuelles
+ erreurs

Nom du sketch

Mode
« en cours »

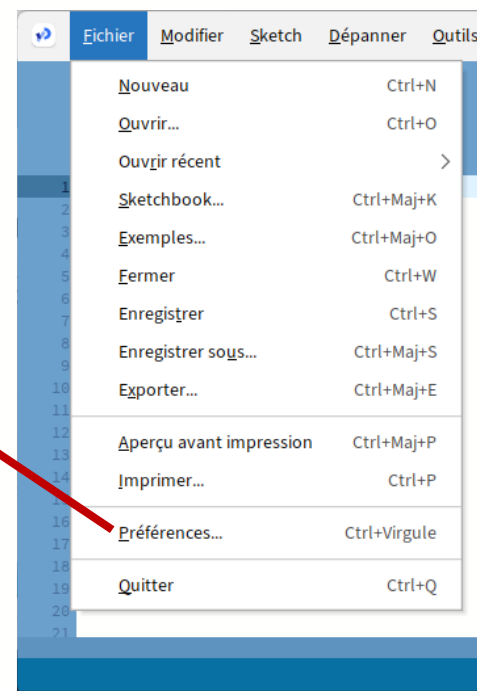
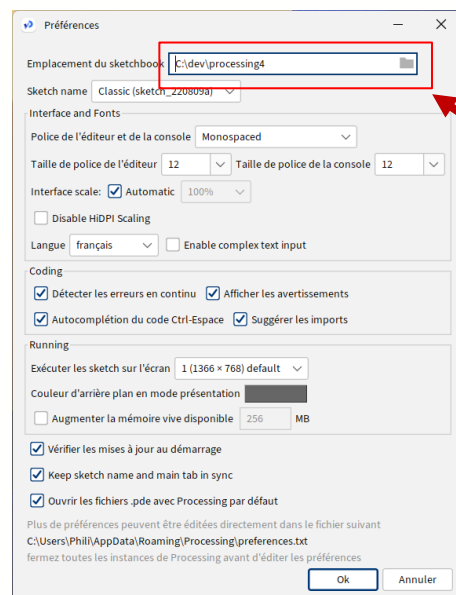




sketch_220919a

Structure

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





sketch_220919a ▾

Structure

options



Préférences

Emplacement du sketchbook

Sketch name

Interface and Fonts

Police de l'éditeur et de la console

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

Interface scale: ☒ Automatic

☐ Disable HiDPI Scaling

Langue ☐ 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

Couleur d'arrière plan en mode présentation

☐ Augmenter la mémoire vive disponible 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



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
 - Plus d'autres répertoires

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

Rechercher dans : boutons_i... 🔍

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

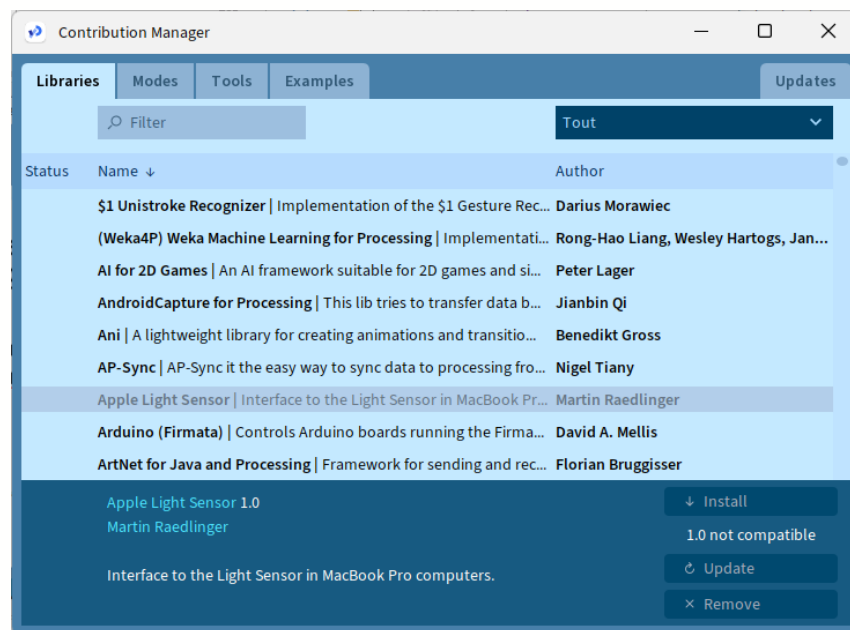


sketch_220919a

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);  
}
```

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

sketch_220919a | Processing 4.0.1



Java ▼

sketch_220919a ▼

Un premier exemple



```
Fichier Modifier Sketch Dépanner Outils Aide bubbles | Processing 4.0.1

void setup() {
  fullscreen(); // mode plein écran
  background(255); // on met le fond en blanc
  frameRate(15); // 15 images par seconde maximum
  noStroke(); // pas de contour
}

// boucle d'affichage
void draw() {
  fill(random(255), random(255), random(255), random(255)); // remplir avec des couleurs aléatoires R G B alpha (transparence)
  circle(random(width), random(height), random(80)); // cercle à position aléatoire de diamètre 80
}
```

L'enregistrement est terminé.

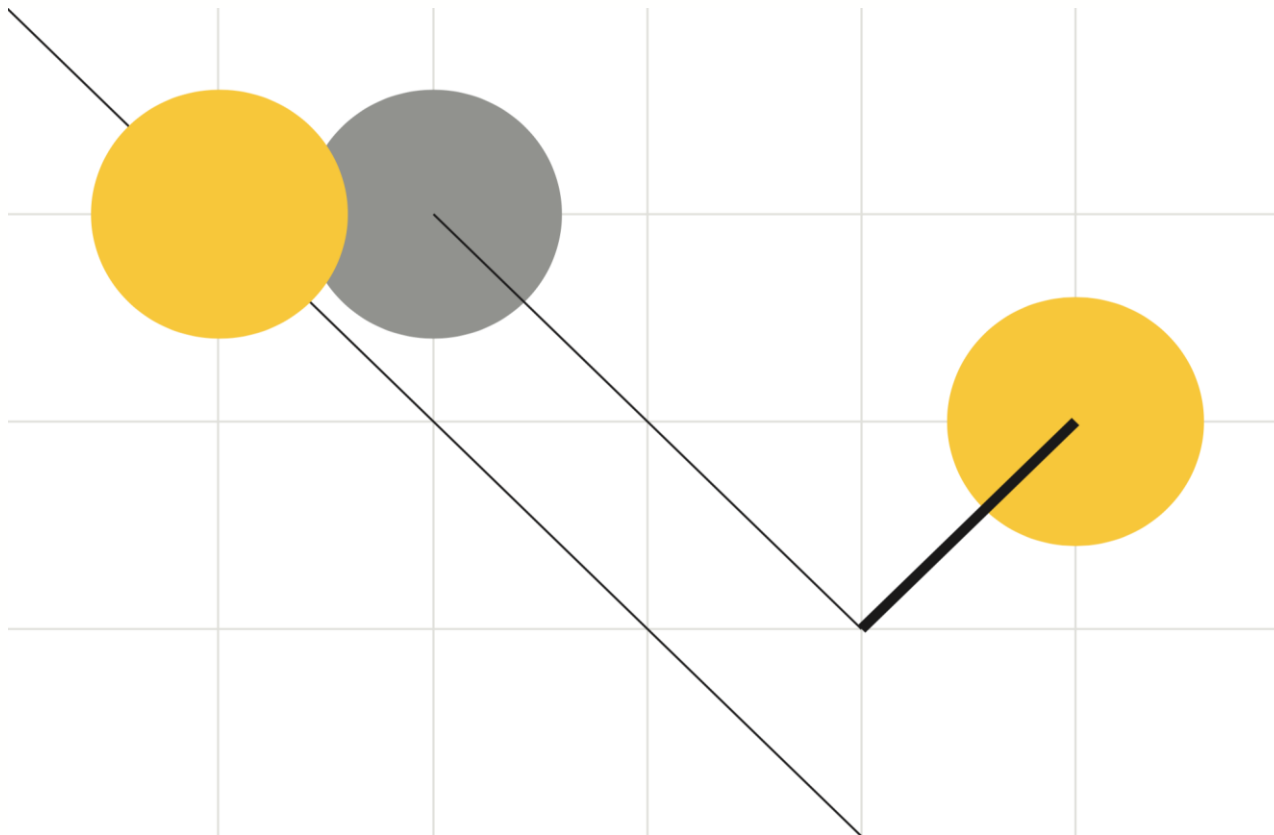
Console Erreurs



Sketch 220919a

Des exercices

- Utiliser des primitives graphiques





Sketch 220919a ▼

Des exercices

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

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27





Java ▼

sketch_220919a ▼

Evénements

- 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*