



sketch_200506a ▼

Introduction à Processing

<https://www.processing.org>

Septembre 2021

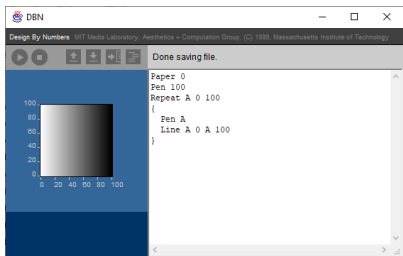
Design by Numbers

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

Date : 1999-2001

Lieu : MIT Media Lab

John Maeda



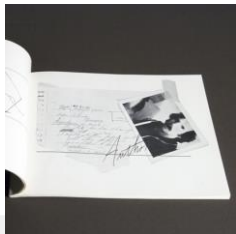
Visible Language Workshop

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

Date : 1975

Lieu : MIT

Muriel Cooper



Processing

<http://www.processing.org>

Date : Printemps 2001

Lieu : MIT Media Lab

Ben Fry / Casey Reas



Processing 3



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





sketch_200506a ▾

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_200506a ▾

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 OS X, and Windows. Select your choice to download the software below.



Processing

4.0 beta 1 (9 août 2021)

Mac OS X

Linux 64-bit

Windows 64-bit

[GitHub](#)

[Report Bugs](#)


[Wiki](#)

[Supported Platforms](#)

Read about the changes in 4.0. The list of revisions covers the differences between releases in detail.

Stable Releases

We need your help!



Help us continue with your generosity!

Donate

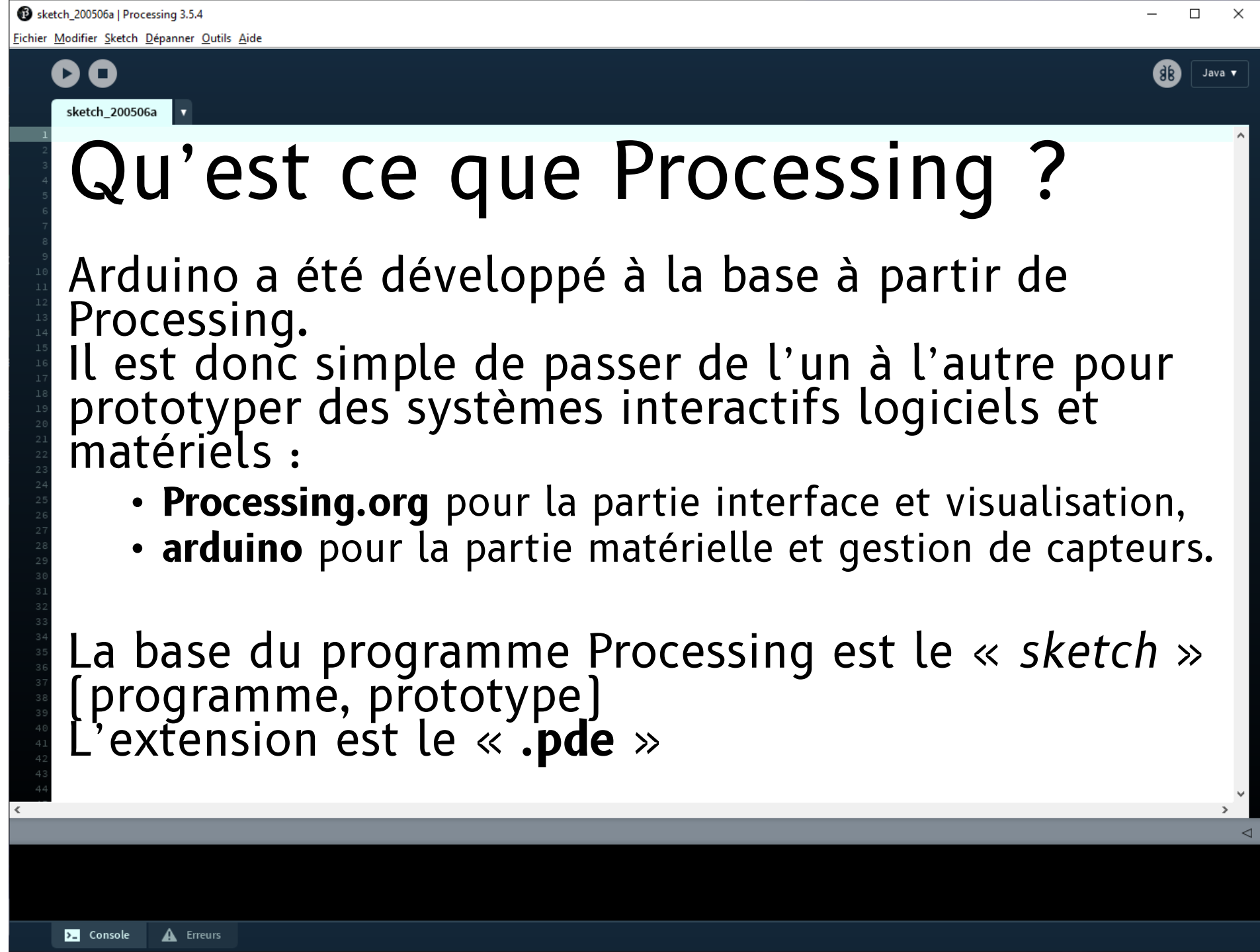
Console Erreurs



sketch_200506a ▼

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>





sketch_200506a

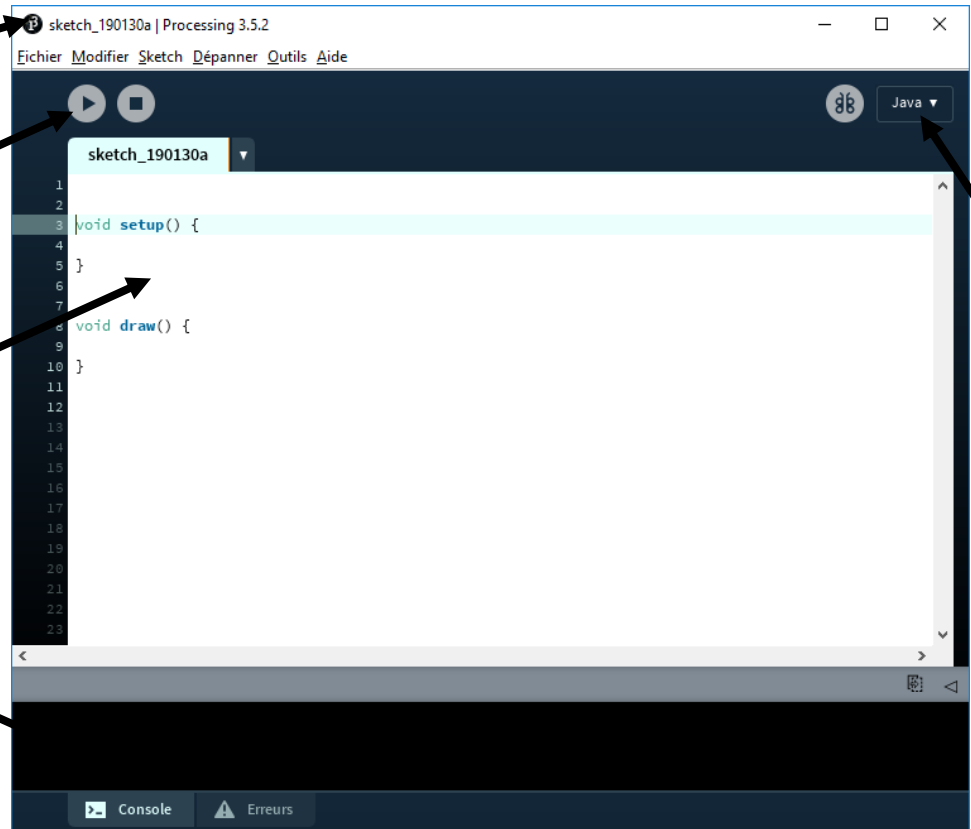
Editeur

Nom du sketch

Lancement du sketch

code

Sorties textuelles
+ erreurs



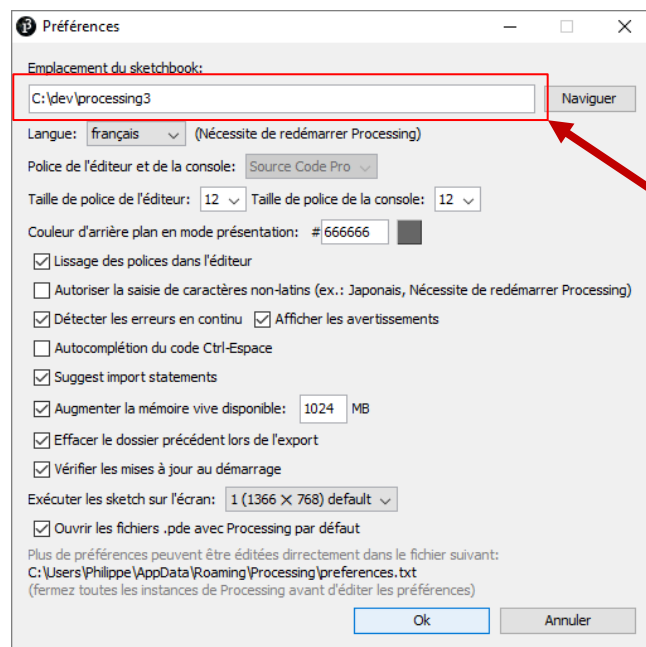
Mode
« en cours »



sketch_200506a

Structure

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



P sketch_160114a | Processing 3.0.1

Fichier Modifier Sketch Dépanner Outils Aide

Nouveau	Ctrl+N
Ouvrir...	Ctrl+O
Ouvrir récent	>
Sketchbook...	Ctrl+Maj+K
Exemples...	Ctrl+Maj+O
Fermer	Ctrl+W
Enregistrer	Ctrl+S
Enregistrer sous...	Ctrl+Maj+S
Exporter...	Ctrl+Maj+E
Aperçu avant impression	Ctrl+Maj+P
Imprimer...	Ctrl+P
Préférences...	Ctrl+Virgule
Quitter	Ctrl+Q

16



sketch_200506a

Structure

options

Préférences

Emplacement du sketchbook:
 Naviguer

Langue: français (Nécessite de redémarrer Processing)

Police de l'éditeur et de la console: Source Code Pro

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

Interface scale: ☒ Automatic 100% (Nécessite de redémarrer Processing)

Couleur d'arrière plan en mode présentation: # 666666

☒ Lissage des polices dans l'éditeur

☐ Autoriser la saisie de caractères non-latins (ex.: Japonais, Nécessite de redémarrer Processing)

☐ Détecter les erreurs en continu ☒ Afficher les avertissements

☒ Auto complétion du code Ctrl-Espace

☒ Suggest import statements

☐ Augmenter la mémoire vive disponible: 256 MB

☒ Effacer le dossier précédent lors de l'export

☒ Vérifier les mises à jour au démarrage

Exécuter les sketch sur l'écran: 1 (1366 X 768) default

☒ Ouvrir les fichiers .pde avec Processing par défaut

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

Ok Annuler

sketch_200506a | Processing 3.5.4

FichierModifierSketchDépannerOutilsAide

sketch_200506a

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 >

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

Console

Erreurs

sketch_200506a

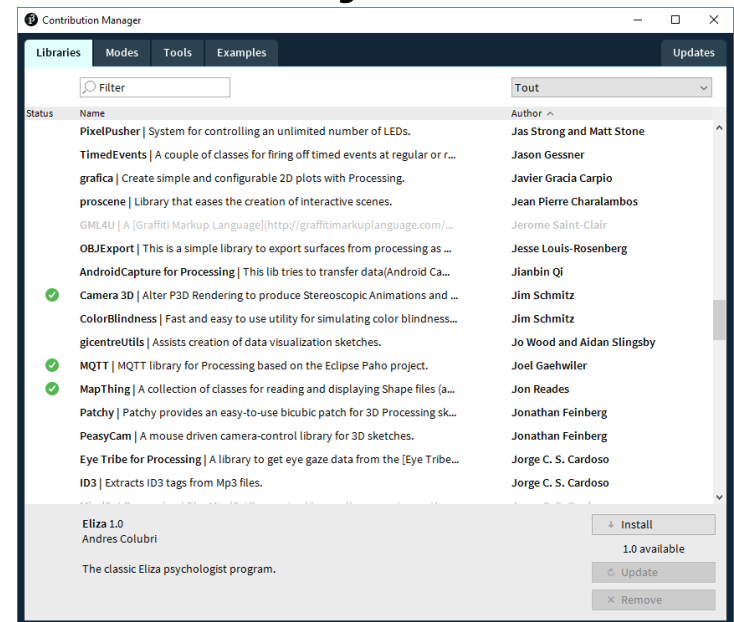
Structure

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

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



Video | GStreamer-based video library for Processing.



The Processing Foundation



sketch_200506a ▾

Processing et IntelliJIDEA



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

sketch_200506a

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

```
void setup()  
{  
    size(200,200);  
    background(102);  
}
```
- **draw** : c'est la boucle de traitement et d'affichage exécutée « à l'infini » [*mainloop*]



sketch_200506a ▼

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()`



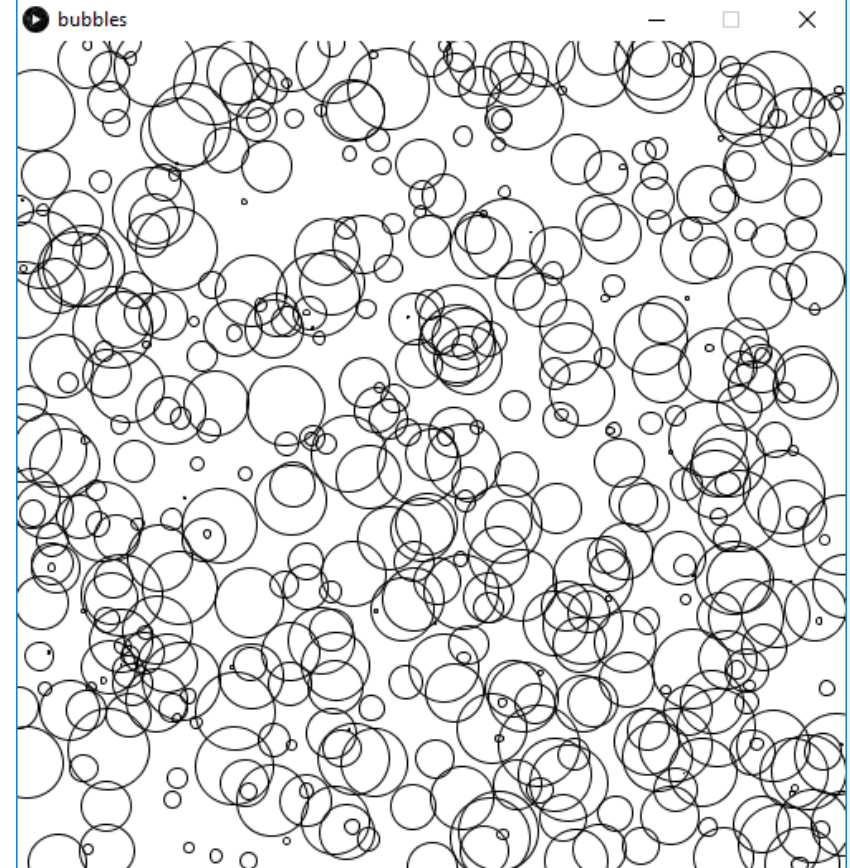
sketch_200506a ▾

Un premier exemple

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

bubbles ▾

1 // variables du sketch (pseudos globales)
2 float x,y, diametre;
3
4 // setup n'est exécutée qu'une fois
5 void setup() {
6   size (500,500); // définit la taille de la fenêtre
7   background(255); // définit le fond de la fenêtre dessiné une fois (blanc)
8   smooth(); // anti aliasing
9   noFill(); // pas de remplissage des formes
10 }
11
12 // draw est exécutée à l'infini
13 void draw() {
14   x = random(width); // tiré au hasard entre 0 et la largeur de la fenêtre
15   y = random(height); // tiré au hasard entre 0 et la hauteur de la fenêtre
16   diametre = random(50);
17
18   // dessine un cercle - primitive pré-définie
19   ellipse(x,y, diametre, diametre);
20 }
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37 L'enregistrement est terminé.
38
39
40
41
42 Console Erreurs
43
44
```





sketch_200506a ▼

Exercices à venir

- 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*