



PROCESSING
 VIEW ALL & PRINT
Présentation
Introduction
Exemples d'utilisations
L'installation de Processing
Les bases de Processing
Dessiner
L'espace de dessin
Les formes
Les couleurs
Le texte
Dessiner plus
Les images
Les styles de bordures
La typographie
Les transformations
Programmer
Les variables
Les conditions
Les répétitions
Les listes
Les méthodes
Les objets
Les commentaires
Animer
La méthode draw
La ligne de temps
L'animation d'un objet



Les bases de Processing

Processing propose à la fois un environnement de création complet et un ensemble de fonctionnalités supplémentaires qui viennent enrichir les possibilités du logiciel. Cet environnement permet d'écrire des programmes (appelés sketch dans Processing), de les convertir en fichiers autonomes, de les publier ainsi que d'identifier et de corriger les erreurs. Il est simple et contient les fonctions essentielles à la programmation tout en restant accessible.

Processing est basé sur le langage Java. C'est cette syntaxe qui sera utilisée lorsque vous allez programmer. Processing vous facilite la maîtrise de ce langage en se chargeant de manière transparente des opérations relativement complexes comme gérer les fenêtres, le son, la vidéo, la 3D et bien d'autres choses encore. Il propose une grande palette de fonctionnalités prédéfinies qui simplifient la création de logiciels graphiques et la rendent accessible aux personnes qui désirent simplement créer graphiquement sans avoir de notions complexes de programmation et de mathématiques.

Ce chapitre vous présente les bases de l'interface de Processing et les notions minimales de la syntaxe Java à connaître pour bien débuter.

Actuellement l'interface de Processing est disponible uniquement en anglais. Malgré cela, son utilisation reste simple et il est probable que les versions à venir proposeront une traduction du logiciel en plusieurs langues dont le français.

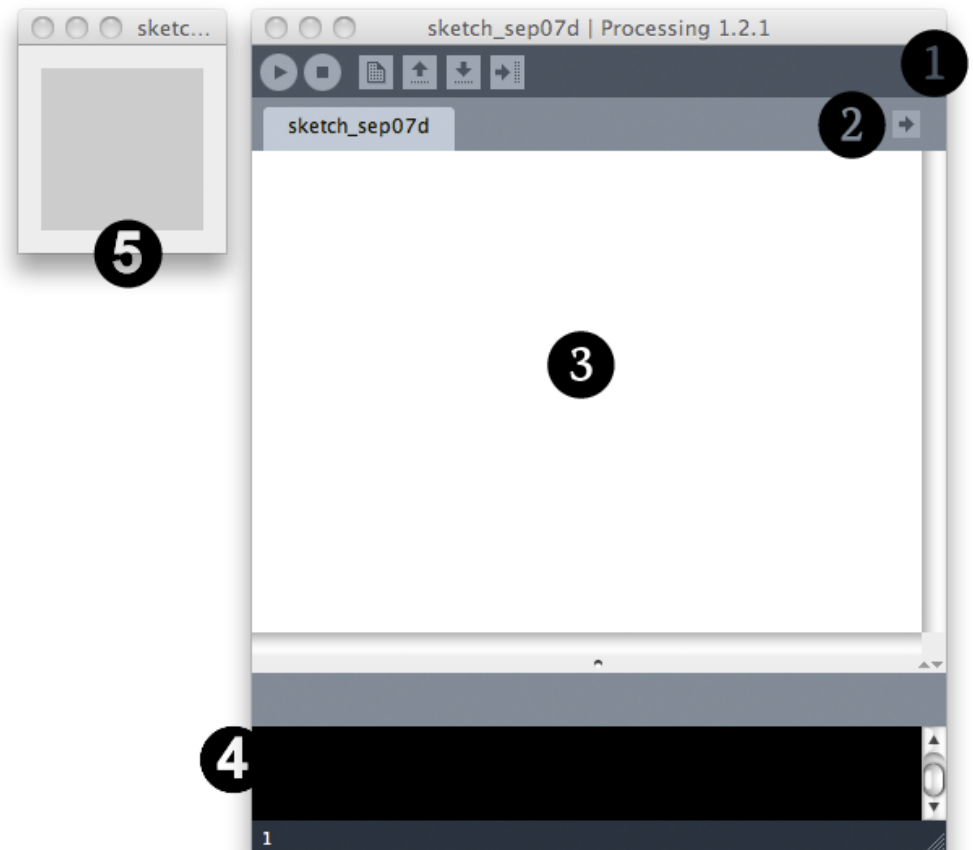
L'interface

L'interface d'utilisation de Processing est composée de deux fenêtres distinctes : la fenêtre principale dans laquelle vous allez créer votre projet et la fenêtre de visualisation dans laquelle vos réalisations apparaissent.

On trouve les éléments suivants dans l'interface :

1. Barre d'actions
2. Barre d'onglets
3. Zone d'édition
4. Zone de sortie
5. Fenêtre de visualisation
6. Barre de menu (non représentée sur la copie d'écran)

L'animation de plusieurs objets
La lecture du son
Sortir
L'exportation
L'impression
La vidéo
Interagir
Les événements clavier
Les événements souris
L'entrée microphone
L'entrée vidéo
Annexes
Les astuces
Les erreurs courantes
Les bibliothèques externes
La documentation en ligne
Arduino
A propos de ce manuel
Glossaire
Credits



Barre d'actions



Exécute votre sketch (votre programme).



Stop l'exécution de votre sketch.



Crée un nouveau sketch.



Ouvre un sketch existant.



Sauvegarde le sketch en cours.

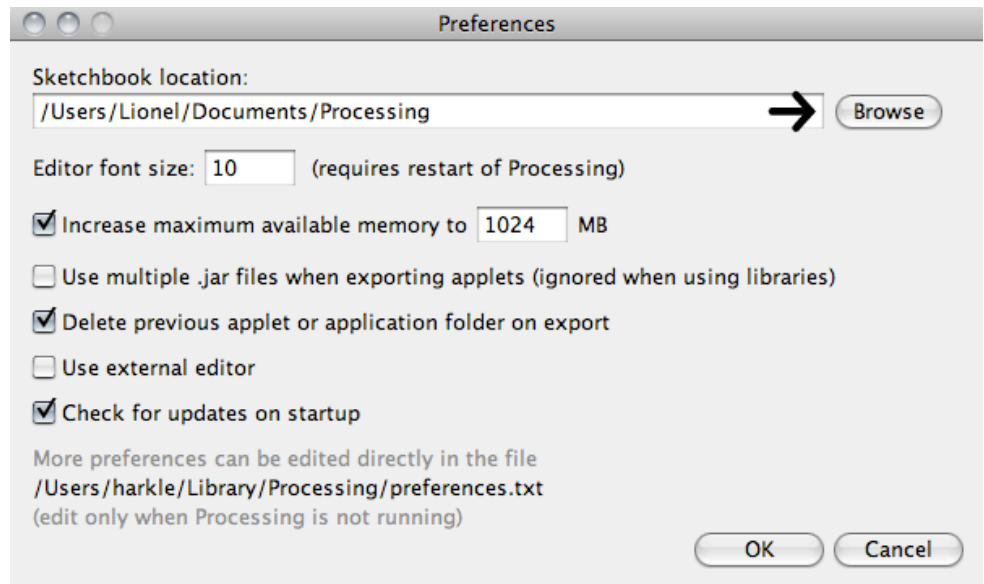


Exporte le sketch pour le web.

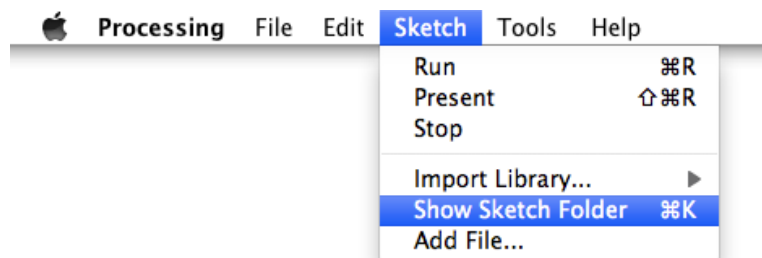
Le dossier de travail

C'est le dossier dans lequel seront enregistrés les sketch (les programmes) et les bibliothèques (des modules externes proposant des fonctionnalités supplémentaires). Par défaut ce dossier se nomme *Processing* et se trouve dans *Documents* (sous Mac) ou *Mes Documents* (sous Windows). Sous GNU/Linux, il est dans votre dossier personnel sous le nom de *sketchbook*.

Pour modifier ce dossier, allez dans le menu *Files > Preferences*. Dans la boîte de dialogue qui apparaît, cliquez sur *Browse* pour choisir le dossier qui vous convient.



A tout moment, pour savoir quel est votre dossier de travail, sélectionnez, dans le menu *Sketch* > *Show Sketch Folder*. Cette option est également accessible via le raccourci ctrl-k sous Windows/Linux ou cmd-k sur Mac :



Bases du langage

Processing utilise le langage Java pour créer vos sketch. Ce langage, qui va être lu par votre machine afin de créer votre sketch, possède un certain nombre de règles de syntaxe qui, si elles ne sont pas respectées empêcheront l'exécution correcte du programme. Il y a aussi un certain nombre de concepts de base nécessaires à connaître.

Processing est sensible à la casse, il fait la différence entre les majuscules et les minuscules: **libre** est différent de **Libre** !

Le point virgule

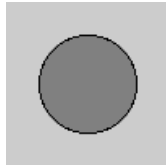
A la fin de chaque instruction (dessiner un cercle, effectuer un calcul, etc.) on doit mettre un « ; » afin de signaler à l'ordinateur la fin de l'instruction. Dans l'exemple ci-dessous, on utilise les caractères « // » pour insérer un commentaire qui sera ignoré lors de l'exécution.

```
//Dessine un cercle
ellipse(10,10, 10, 10);

//Crée une variable
int chiffre = 10 + 23;
```

Appels de méthodes

Processing propose un grand nombre de fonctionnalités prédéfinies appelées méthodes : dessiner un rectangle, définir une couleur, calculer une racine carrée, etc. Ces méthodes ont chacune un nom spécifique. Pour les invoquer, il suffit de taper leur nom en respectant les majuscules et minuscules et de coller des parenthèses après le nom : parfois on doit préciser certaines valeurs à l'intérieur des parenthèses (couleur, position, taille, etc.). L'exemple ci-dessous affiche un cercle gris.



```
fill(128);  
ellipse(50, 50, 60, 60);
```

Affichage dans la zone de sortie

La zone de sortie (la zone 4 dans le schéma de l'interface présenté plus haut) permet d'afficher du texte brut à des fins de test et de débogage (correction). Pour afficher quelque chose, il faut utiliser la méthode `println()`;

```
println("Salut tout le monde!");
```

```
println(1000);
```

Opérations arithmétiques

Processing permet d'effectuer des calculs mathématiques. Tout au long de l'utilisation de cet environnement, vous serez amené à calculer des valeurs. Ne vous inquiétez pas : l'ordinateur le fera pour vous. Les opérations d'addition, soustraction, multiplication et division peuvent être combinées. On peut utiliser des parenthèses pour définir l'ordre des opérations. **Attention aux nombres à virgule !** On utilise un point pour séparer les unités des décimales et non pas une virgule. Voici quelques exemple :

```
println(10 + 5);  
println(10 + 5 * 3); // 5*3 (soit 15) puis additionne 10  
println((10 + 5) * 3); // 10+5 (soit 15) puis multiplie 15 par 3  
println(10.4 + 9.2);
```

Cela va produire le résultat suivant :

Certaines opérations arithmétiques peuvent être contractées. Par exemple, `i++` donne le même résultat que `i = i + 1`. Et `x+=10` donne le même résultat que `x=x+10`.

Maintenant que vous connaissez les bases de l'interface de Processing, il ne vous reste qu'à apprendre à écrire du code qui fonctionne, afin de réaliser des projets qui dessineront et créeront ce que vous voulez. Nous allons vous accompagner pas à pas.

