//////////////////////

# DESIGNING SYSTEMS

//////////////////////////////////////////////////////////////////////////

## BASES : Formes_01

//////////////////////

**Sommaire**

- Qu'est-ce que Processing ?
- Comment installer ?
- Notre premier Sketch.
- Introduction à l'interface.

**Concepts / Mots Clés**

design génératif, design d'interaction, applications digitales, interface

**Qu'est-ce que Processing ?**

Processing est à la fois un langage et un environnement de programmation pour des artistes & designers visuels travaillant avec des images 2D/3D, de l'animation et de l'interaction. Processing a été crée par Ben Fry et Casey Reas en 2001 pendant leurs études au MIT sous la direction de John Maeda.

Processing est entièrement Open Source, un logiciel libre. Il est devenu un outil de création fort, une alternative importante aux logiciels commerciaux, accessible à tous et soutenu par une communauté internationale et passionnée.

Processing est utilisé dans les domaines des arts visuels pour créer toutes sortes d'applications créatives que ce soient pour l'édition, le web ou pour des installations artistiques de grand taille. Il est à la fois un outil de travail et une manière d'aborder les différentes phases de création : conceptualisation jusqu'au rendu final.

**Comment installer Processing ?**

L'installation de Processing est assez simple et dépend de la plate-forme que vous voulez utiliser. Dans tous les cas, allez sur la page de téléchargement: http://processing.org/download/ et cliquez sur le nom de votre plate-forme.

## Notre premier sketch

À l'ouverture du logiciel, vous aurez une fenêtre (la zone d'édition) qui ressemble un peu comme un éditeur de texte avec un espace vide dans lequel vous tapez votre code.

Ecrivez le code suivant dans la zone d'édition :

```
ellipse(50,50,100,100);
```

Ensuite appuyez sur le button 'lecture' de Processing pour exécuter le code et visualiser le résultat.

## Introduction à l'interface

L'interface d'utilisation de Processing est composée de deux fenêtres distinctes :

- La fenêtre principale dans laquelle vous allez écrire votre code.
- La fenêtre de visualisation dans laquelle vous 'visualisez' votre code.

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

Vous pouvez ouvrir l'image, 'L'interface.jpg' dans le dossier DOCS pour voir les noms en anglais.

## Anatomie d’un programme en Processing

Pour mieux comprendre les notions de bases de la programmation lisez bien le bref article suivant : http://fr.wikipedia.org/wiki/Langage_de_programmation

Voir le sketch *PROGRAM ANATOMY / ANATOMIE D'UN PROGRAMME*

////////////////////////////////////

```
float dia; // Déclaration d'une variable globale.


// Bloque de code setup() { ... }
void setup() {

  size(400, 400); // Ceci est une ligne de commande ou une instruction.
  smooth();
  fill(0,0,255);
  noStroke();
  dia = 0; // L'affectation d'une valeur de départ à notre variable.

}


// Bloque de code draw() { ... }
void draw() {
  background(0);
  // Notez que notre fonction ellipse contient des valeurs constantes & la variable
'dia'.
  // Les valeurs ajoutées dans une fonction sont leurs 'arguments' ou 'paramètres'.
  // Il existe des fonctions qui ne prennent pas d'arguments, Ex. noStroke(); noFill();
  // Chaque valeur correspond donc à un argument/paramètre spécifique.
  // Voir le référence : https://processing.org/reference/ellipse_.html
  // ellipse(a,b,c,d);
  // l'argument a est égale à la position de notre ellipse sur l'axe x en pixels
  // l'argument b est égale à la position de notre ellipse sur l'axe y en pixels
  // l'argument c est égale à la largeur de notre ellipse en pixels
  // l'argument a est égale à la hauteur de notre ellipse en pixels
  ellipse(200, 200, dia, dia);

  // les commentaires sont souvent utilisés pour désavctiver une instruction.
  // Activez cette ligne et voir ce qu'il se passe.
  //ellipse(dia, dia, 50, 50);

  dia = dia + 0.1; // Mise-à-jour de la variable.
```

```
}


// Bloque de code fonction { ... }
void keyReleased() {

  if (key == 's' || key == 'S') { // Une structure conditionnelle.
    saveFrame("img_###.png");

  }
}
```

////////////////////////////// FIN

Ressources

http://fr.flossmanuals.net/Processing/

http://processing.org/learning/gettingstarted/

http://fr.wikipedia.org/wiki/Terminologie_informatique

http://fr.wikipedia.org/wiki/Langage_de_programmation