

```
////////////////////
DESIGNING SYSTEMS
////////////////////
BASES : VARIABLES
////////////////////
```

Sommaire

- Les arguments
- Les valeurs constantes
- Les valeurs qui varient !

Concepts / Mots Clés

arguments, variables, données, types de données

Les arguments

Vous avez sans doute remarqué que la plupart des fonctions en Processing acceptent des valeurs en chiffres entre les parenthèses :

```
ellipse(30, 100, 50, 50);
```

Dans le langage de la programmation on appelle celles-ci les *arguments* ou les *paramètres*. Lorsqu'on recherche dans la référence de Processing nous trouvons une définition de la fonction. (Sélectionnez la fonction, faites un clique droit et choisissez «Find in reference» dans le menu déroulant.

Syntax	ellipse(a, b, c, d)	
Parameters	a	int or float: x-coordinate of the ellipse
	b	int or float: y-coordinate of the ellipse
	c	int or float: width of the ellipse
	d	int or float: height of the ellipse

a, b, c, d sont des *arguments* aussi appelés *paramètres*. Il existe également des fonctions qui n'acceptent pas d'arguments comme `noStroke()` / `noFill()`...

Les valeurs constantes

```
ellipse(30, 100, 50, 50);
```

Dans cette fonction, toutes les valeurs attribuées aux arguments sont des valeurs constantes. Nous dessinons donc un graphique statique. Il est possible de changer ces valeurs au fur et à mesure que notre programme est en marche et ainsi nous pouvons faire bouger ou même interagir avec notre cercle. Pour ce faire il va falloir créer ce qu'on appelle une variable.

Les valeurs qui varient !

Les variables permettent de stocker des informations - des données.

Ces données peuvent être des chiffres, des chaînes de caractères, des messages, des images, des couleurs. Il existe donc différents types de données. Ce qui est important à saisir, c'est qu'une variable est une donnée qui varie, que nous avons besoin de créer cette variable nous même et que cette variable est souvent utilisée dans différentes parties du programme.

Au début, les variables s'avèrent parfois un peu compliquées car la notion est abstraite mais avec un peu d'exercice vous verrez que c'est assez simple. Les variables sont omniprésentes dans la programmation et donc la notion des variables est importante à saisir dès le début sinon vous allez vous perdre très vite et vous allez manquer un concept très puissant de la programmation.

Il existe 3 étapes dans l'utilisation des variables.

Étape 1. Type de donnée

```
int / float / String / boolean / PFont...
```

On doit toujours désigner le type de données dans Processing. C'est du '*typage explicite*' - C'est-à-dire que nous indiquons nous mêmes le type lors de sa création.

Étape 2. Nom de la variable

```
posX, posY, maCouleur...
```

Nous pouvons écrire n'importe quel nom pour notre variable tant qu'il ne porte pas le même nom d'une fonction ou d'un mot clés existant dans le langage de Processing. Essayez de nommer vos variables avec un minimum de logique. Par exemple, si vous allez utiliser la variable pour varier la taille d'un rectangle, choisissez `taille` / `size` comme nom.

Étape 3. Attribution/Affectation

avec la symbole “=” suivi par une valeur

Finalement, nous donnons toujours une valeur initiale à notre variable. Souvent cette valeur n’est qu’une valeur de départ car comme vous avez bien compris, tout l’intérêt d’une variable est la possibilité de la faire varier !

Voici quelques exemples

```
// cette déclaration indique explicitement que
// la variable xPos est de type entier (int)
int xPos = 40;

//la variable theta est de type virgule flottant (float)
float theta = 30.35;

//la variable mesMots est de type caractères (String)
String mesMots = "Ceci n'est pas un pipe.";

//la variable font est de type PFont pour manipuler des fontes.
PFont font = createFont("Helvetica", 30);
```

Voir les sketches *Variables_01* / *Variables_02* / *Variables_03*.