



Les variables

Une variable est une donnée que l'ordinateur va stocker dans l'espace de sa mémoire.

42	491	33.145	5	« Salut tout le monde ! »	true
x	y	angle	i	message	vie

Une variable est définie par deux **éléments** :

Un **nom** qui sert à désigner un emplacement dans la mémoire centrale et permet d'y accéder facilement.

Un **type** qui détermine la taille de la variable et la façon dont est traduite la valeur en code binaire.

Exemples : **int** nombre entier **float** nombre à virgule.

type de donnée	taille
boolean	1 bit
byte	8 bits
char	16 bits
int	32 bits
float	32 bits
color	32 bits

Déclaration et affectation

On déclare une variable en indiquant d'abord le type de données qui va être stocké suivi par le nom de la variable. On affecte une valeur à une variable en utilisant le caractère = . En programmation le caractère = affecte la valeur à droite du = au nom de la variable qui se trouve à gauche.

```
int x ; // on déclare une variable x qui contiendra un entier
float y ; // on déclare une variable y qui contiendra un décimal
boolean b ; // on déclare une variable b qui contiendra un booléen
x=50 ; // on affecte la valeur 50 à x
y=12.6 ; // on affecte la valeur 12.6 à y
b=true ; // on affecte la valeur true à b
```

On peut déclarer plusieurs variables du même type en une seule instruction.

```
int x,y,z ; // on déclare trois variables x,y et z qui contiendront des entiers
```

int nombre entier

Essayer ce programme pour comprendre :

```
int entier;
entier = 3;
int x = 0;
println(entier);
line (x,x,100,100);
x=50;
line (x,0,x,100);
```

float nombre avec décimales.

double nombre avec décimales mais plus de précision que float.

Essayer ce programme pour comprendre :

```
float decimal;
double long_decimal;
decimal = PI;
long_decimal = PI;
println( "La valeur de pi est " + decimal);
println( "La valeur de pi est " + long_decimal);
```

boolean variable qui contient deux états : vrai(true) ou faux(false).

Essayer ce programme pour comprendre :

```
boolean vraifaux;
int x = 8;
vraifaux = true;
println(vraifaux);
vraifaux = ( x == 7);
println(vraifaux);
```

char variable qui contient une lettre.

Essayer ce programme pour comprendre :

```
char lettre1, lettre2;
lettre1 = &apos;A&apos;;
lettre2 = &apos;a&apos;;
println("lettre1 contient "+ lettre1);
println("lettre2 contient "+ lettre2);
```

string variable qui contient du texte.

Essayer ce programme pour comprendre :

```
String texte;
texte = "Bonjour!";
println("Le texte contient "+ texte);
```

color variable qui contient une couleur.

Essayer ce programme pour comprendre :

```
color blanc = color (255,255,255);
color noir = color (0,0,0);
fill(blanc);
rect(0,0,25,25);
fill(noir);
rect(25,25,100,100);
```

Le nom des variables

On peut choisir le nom des variables mais il faut respecter certaines règles.

t	~	Un nom de variable court est conseillé pour une variable avec peu d'espérance de vie (quelques lignes). Pour des variables utilisables tout au long du programme, il faut un nom plus explicite.
tempé	X	Il n'est pas possible d'utiliser des caractères spéciaux tels que les accents, les espaces, les @, \$ % etc... dans le nom de variable. Ceux-ci provoqueront une erreur.
interieurtemperature	~	Ce nom de variable est un peu long, des noms trop longs alourdiront le code à sa relecture et utiliseront plus d'espace mémoire. Ce n'est pas défendu mais peu conseillé.
int	X	int est un des noms réservés de Processing, il en existe quelques uns tels que for, if, float, true.... Ces mots ne peuvent pas être utilisés pour des variables et provoqueront une erreur.
1nttemp	X	S'il est autorisé d'utiliser des numéros dans le nom d'une variable, il n'est pas possible de commencer le nom d'une variable avec un numéro.
tempinter	~	Ce choix de nom semble judicieux pour une variable importante, cependant si le nom de la variable possède plus d'un mot, il est conseillé de capitaliser la première lettre des mots à partir du deuxième.
tempInter	✓	Ce choix facilite l'utilisation de la variable au cours du programme, il est explicite, concis et respecte les conventions d'écriture du code Java.

Les variables prédéfinies

Dans Processing, certaines variables sont déjà déclarées. Ce sont des variables prédéfinies.

Pour connaître la largeur et la hauteur de la fenêtre graphique, on utilise les variables `width` (largeur) et `height` (hauteur).

Essayer ce programme pour comprendre :

```
// Les variables width et height ont déjà été déclarées et contiennent 100.
println(width + "," + height);
size(200,300); // La fonction size va changer la taille de la fenêtre.
println(width + "," + height);
```

```
// Affiche la valeur de pi
println( "La valeur de pi est " + PI);
```

Arithmétiques

On peut modifier les valeurs d'une variable par des opérations comme l'addition, la multiplication...

Essayer ce programme pour comprendre :

```
int a = 3;
int b = 5;
int c = a+b;
println(" La valeur de c est "+ c);
a = b * 2;
println(" La valeur de a est "+ a);
c = a + b;
println(" La valeur de c est "+ c);
a = c/b;
println(" La valeur de a est "+ a);
```

Les expressions peuvent être utilisées directement dans des instructions.

Essayer ce programme pour comprendre :

```
int a = 100;
int b = -20;
line (a*2,50,a+b,a);
```

Essayer ce programme pour comprendre :

```
int a = 30;
line(a,0,a,height);
a = a + 40;
strokeWeight(4);
line(a,0,a,height);
```

Essayer ce programme pour comprendre :

```
float y = 20;
println(y);
line(0,y,width,y);
y = y * 1.6;
println(y);
line(0,y,width,y);
y = y * 1.6;
println(y);
line(0,y,width,y);
y = y * 1.6;
println(y);
line(0,y,width,y);
```