Chap. 3 : Les variables (var)

I. Comment fonctionne un ordinateur ?

On a vu que pour fonctionner un ordinateur à besoin de programmes. Ces programmes sont composés de suite d'instructions (d'ordres) simples que l'on mémorise dans la mémoire centrale de l'ordinateur. Les ordres (instructions) placés en mémoire sont réalisés par le microprocesseur qui est la partie active de l'ordinateur.
Dans l'exemple ci dessous, on donne 4 ordres à l'ordinateur :

a = 2 ; // la variable a prend la valeur 2.
b = 3 ; // la variable b prend la valeur ___.
a = b ; // la variable a prend la valeur de ___. Donc a à pour valeur ___ .
trace(a) ; // on affiche la valeur de a dans l'aire de sortie (Output). Donc il s'affiche ____ .dans cette aire

Pour optimiser la place dans la mémoire et la vitesse de traitement du microprocesseur ont type les variables suivant leur contenu.

II.La notion de type

Un programme gère des informations de natures diverses. Ainsi, les valeurs telles que 123 ou 2.4 sont de type numérique tandis que "Sti2d" est un mot composé de caractères.

La notion de type permet de différencier ces données. Ainsi, un type de données décrit le genre d'informations qu'une variable peut contenir.

Sous AS3, il existe trois catégories de données : logique, numérique et caractère.

II.1.Catégorie logique

La catégorie logique est représentée par le type **Boolean** (Booléen). Les valeurs logiques ont deux états : true (vrai ou 1) ou false (faux ou 0).
Les valeurs booléennes sont le plus souvent utilisées dans les instructions effectuant des comparaisons pour contrôler le déroulement d'un programme (a < b , a == b , a >b , ...).

II.2.Catégorie numérique

Cette catégorie contient trois types distincts : Number, int, uint.

- Le type **Number** est utilisé pour décrire les nombres réel de 4.9406564584124654e-324 à 1.79769313486231e+308.
- Le type **int** (integer) permet l'utilisation de nombres entiers positifs ou négatifs de -2147483648 à 2147483647.
- Le type **uint**, quant à lui, correspond aux entiers uniquement positifs (uint : unsigned int) de 0 à 4294967295.

L'utilisation des différents types numériques permet de gérer au mieux l'espace mémoire utilisé par l'animation. En effet, un nombre réel prend plus d'espace mémoire qu'un entier.

Remarque : Le type uint est essentiellement utilisé pour décrire les couleurs des objets à dessiner.
Exemples : *var couleur1 : uint = 0xFF0000 ; //couleur rouge.*
var couleur2 : uint = 0x00FF00 ; //couleur verte.
var couleur3 : uint = 0x0000FF ; //couleur bleue.

II.3.Catégorie caractère

Le type de données représentant la catégorie caractère est le type **String**. Un String est une chaîne de caractères, autrement dit, une suite de caractères (suite de lettres, chiffres et signes de ponctuation, par exemple).
La déclaration d'une variable contenant des caractères s'effectue en plaçant la suite de caractères entre des guillemets droits simples ou doubles. De cette façon, la chaîne de caractères est traitée comme étant un mot et non comme représentant le nom d'une variable.
Par exemple, dans l'instruction :

var maVariableString : String = "Bonjour";

Grâce aux guillemets, le terme "Bonjour" est considéré comme une chaîne et non comme une variable. Le terme *maVariableString* correspond au nom de la variable contenant la suite des caractères *Bonjour*.

III.Applications

Pour étudier les différents types, nous allons créer un nouveau projet intitulé *App sur les variables*.

La scène (stage) du projet sera composé de 3 objets TextField :

- texte de l'énoncer : *texteEnoncer ;*
- texte de saisi des textes : *texteEntrer ;*
- texte d'affichage de la réponse : *texteSortie ;*

Et d'un Sprite (objet graphique) pour valider la réponse : *boutonValider.*

Le code à écrire pour créer la scène est le suivant :

```
public class Main extends Sprite
	{	//Déclaration des variables (objets) globales
		//zone de la question :
		private var texteEnoncer : TextField = new TextField();
		//zone de saisie du texte :
		private var texteEntrer : TextField = new TextField();
		//zone d'affichage du texte :
		private var  texteSortie : TextField = new TextField();
		//Bouton validé reponse à la question:
		private var boutonValider : Sprite = new Sprite();

		public function Main():void
		{
			if (stage) init();
			else addEventListener(Event.ADDED_TO_STAGE, init);
		} //accolade de la méthode Main

		private function init(e:Event = null):void
		{	removeEventListener(Event.ADDED_TO_STAGE, init);

			// Point d'entrée :
			// organisation de la scène
			texteEnoncer.width = 400;
			texteEnoncer.height = 40;
			texteEnoncer.defaultTextFormat = new TextFormat("arial", 20)  ;
			texteEnoncer.text = "l'énoncer";
			addChild(texteEnoncer);
```

```
            //tester

            texteEntrer.y = texteEnoncer.y + texteEnoncer.height;
            texteEntrer.width = 400;
            texteEntrer.height = 40;
            texteEntrer.defaultTextFormat = new TextFormat("arial", 20)  ;
            texteEntrer.text = "";
            addChild(texteEntrer);
            //tester
            texteEntrer.border = true;
            //tester
            texteEntrer.type =  TextFieldType.INPUT ;
            //tester

            texteSortie.y = texteEntrer.y + texteEntrer.height + 10;
            texteSortie.width = 400;
            texteSortie.height = 100;
            texteSortie.multiline = true;
            texteSortie.wordWrap = true;
            texteSortie.text = "Texte de sortie";
            addChild(texteSortie);
            texteSortie.defaultTextFormat = new TextFormat("arial", 20)  ;
            texteSortie.text = "Texte de sortie";
            //tester

            boutonValider.graphics.beginFill(0xFF0000);
            boutonValider.graphics.drawRoundRect(0, 0, 200, 40, 20, 20);
            boutonValider.y = texteSortie.y + texteSortie.height +20;
            boutonValider.buttonMode = true;
            addChild(boutonValider);
            //tester
            boutonValider.addEventListener(MouseEvent.CLICK, question1);
            //tester


    }//accolade de la méthode init

    private function question1(e:MouseEvent):void
    {       //Déclaration des variables de lamethode question1
            var base : Number ;
            var hauteur : Number ;
            var hypotenuse : Number;

            base = 3.2 ;
            hauteur = 4.5 ;

            // affiche l'énoncer
            texteEnoncer.text = "L'hypothénuse d'un triangle rectangle";

            //calcul de l'hypoténuse
            // compléter l'instruction ci-dessous pour calculer hypoténuse :
            hypotenuse =  0

            // affiche la valeur de l'hypotenuse dans le textfield de sortie
```

```
            texteSortie.text = "L'hypothénuse est " + hypotenuse;

        }//accolade de la méthode question1

    }// Accolade de la class Main
```

III.1.Etude des nombres

Travail à faire :

- Compléter la méthode *question1* pour que le programme effectue le calcul de l'hypoténuse d'un triangle.
  Formule mathématique : _____
  Traduction de la formule en instruction : _____
  Réponse affichée : _____
- Changer le type de l'hypotenuse en nombre entier : _____
  Réponse affichée : _____

Remarque : Pour convertir un type dans un autre type, il suffit de faire type(maVariable).
Exemple : int(maVariable) → converti maVariable en nombre entier.
Number(maVariable) → converti maVariable en nombre réel.
String(maVariable) → converti maVariable en chaine de caractère.

III.2.Etude des chaines de caractère

Vous allez compléter l'application pour que celle-ci dialogue avec l'utilisateur.
Le dialogue est le suivant :

Énoncer : Quel est votre nom ?
Entrer par l'utilisateur : Dupont
Énoncer : C'est noté. Et votre prénom ?
Entrer par l'utilisateur : Pierre
Énoncer : Êtes vous un homme ou une femme ?
Entrer par l'utilisateur : homme //*utiliser l'instruction if.*
Sortie : Parfait. Je suis enchanté de faire votre connaissance M. Pierre Dupont.

Énoncer : Est-ce indiscret de vous demander votre âge?
Entrer par l'utilisateur : non //*utiliser l'instruction if.*
Énoncer : Dans ce cas, quel âge avez-vous M. Dupont ?
Entrer par l'utilisateur : 20 //*utiliser l'instruction if.*
Sortie : Félicitation vous êtes jeunes !

Énoncer : Savez vous que je suis excellent en calcul mental? Proposer moi un nombre s'il vous plait.
Entrer par l'utilisateur : 337
Énoncer : Je le note. Proposer moi un autre grand nombre s'il vous plait.
Entrer par l'utilisateur : 3456
Sortie : 337 fois 3456 font 1164672 !

Énoncer : Je suis capable de compter très vite !
Énoncer : Jusqu'à combien voulez-vous que je compte?
Entrer par l'utilisateur : 5
Sortie : 1 2 3 4 5 //*utilisation d'une boucle for*

Sortie : Merci pour ce bon moment. A bientôt Pierre

III.3.Aides

a. Syntaxe de l'instruction if (ctrl+B)

L'écriture de l'instruction if-else obéit aux règles de syntaxe suivantes :

```
if (condition) //si la condition est vraie
        {
                //faire plusieurs instructions;
        } //fait
else //sinon (la condition ci-dessus est fausse)
        { /
                //faire plusieurs instructions;
        } //fait
```

1. Si la condition située après le mot-clé if et placée obligatoirement entre parenthèses est vraie, alors les instructions placées dans le bloc défini par les accolades ouvrante et fermante immédiatement après sont exécutées.
2. Si la condition est fausse, alors les instructions définies dans le bloc situé après le mot-clé else sont exécutées.

De cette façon, un seul des deux blocs peut être exécuté à la fois, selon que la condition est vérifiée ou non.

b. Syntaxe de l'instruction for (ctrl+B)

La boucle for s'écrit de cette manière :

```
for (initialisation; condition; incrément)
        {
                plusieurs instructions;
        }
```

Les termes Initialisation, Condition et Incrément sont des instructions séparées obligatoirement par des points-virgules (;). Ces instructions définissent une variable, ou indice, qui contrôle le bon déroulement de la boucle. Ainsi :

- Initialisation permet d'initialiser la variable représentant l'indice de la boucle (exemple : i = 0, i étant l'indice). Elle est la première instruction exécutée à l'entrée de la boucle.
- Condition définit la condition à vérifier pour continuer à exécuter la boucle (exemple i < 10). Elle est examinée avant chaque tour de boucle, y compris au premier.
- Incrément est l'instruction qui permet de modifier le résultat du test précédent en augmentant ou diminuant la valeur de la variable testée. L'incrément peut être augmenté ou diminué (exemple : i = i + 1). Cette instruction est exécutée à la fin de chaque tour de boucle.

Pour notre application, on aura

```
for (var i:int = 0; i < nombreEntrer ; i++)
        {
                leTexte = leTexte + " " + i;
        }//fin de l'accolade de for
```

IV.A retenir

• L'instruction : *var positionEnX : Number = 2;*
définit une variable nommée positionEnX de type numérique, contenant la valeur 2.

• L'instruction : *var mot:String = "Bonjour";*
déclare une variable nommée mot contenant la suite de caractères "Bonjour". Cette variable est de type String.

• La commande : *trace(" x ------> " + positionEnX);*
affiche dans la fenêtre de sortie le contenu de la variable positionEnX précédé du commentaire x ------>.

• Les instructions : *a = 1;*
a = 3;
sont exécutées de haut en bas. Ainsi, la valeur initiale de a (1) est effacée et écrasée par la valeur 3.

• La suite des instructions : *n = 4;*
*p = 5*n+1;*
place la valeur 4 dans la variable n. L'expression 5 * n + 1 est ensuite calculée.
Le résultat est enfin rangé dans la variable représentée par p.

• L'instruction : *a = a + 1;*
est une instruction d'incrémentation. Elle permet d'augmenter de 1, la valeur contenue dans la variable a.

• La suite des instructions : *tmp = a;*
a = b;
b = tmp;
a pour résultat d'échanger les contenus des variables a et b.

• Les instructions : *var mot1:String = "Le mystère ";*
var mot2:String = "de la chambre jaune";
var titre:String = mot1 + mot2;
placent la chaîne de caractères contenue dans mot2 à la suite de mot1, dans la chaîne nommée titre.

• Dans l'instruction : *var phrase :String = "Agent secret " + 707;*
ActionScript convertit le chiffre 707 en chaîne "707" et l'ajoute à la fin de la suite de caractères "Agent secret ". La variable phrase contient la chaîne "Agent secret 707".