

# JS 01 : Structure d'un programme

## Objectifs

- Comprendre les généralités de la programmation et savoir créer un premier programme.
- Connaître les règles et les conventions de syntaxe et de présentation d'un programme.
- Comment déclarer des variables et des constantes, comment les utiliser dans le corps d'un programme.
- Ajouter des commentaires à vos programmes

## Les variables

Les données en mémoire sont stockées dans des variables. Il peut par exemple s'agir de données saisies par l'utilisateur ou de résultats obtenus par le programme, intermédiaires ou définitifs.

Pour employer une image, une variable est une boîte, que le programme (l'ordinateur) va repérer par une étiquette. Pour avoir accès au contenu de la **boîte**, il suffit de la désigner par son étiquette.

Les noms de variables peuvent être choisis librement par le programmeur (sauf parmi les mots-clés). La liste des mots-clés se trouve dans la documentation en ligne, mais dans un problème, une variable est choisie pour jouer un rôle. Il est souhaitable que l'auteur de l'algorithme s'en souvienne en choisissant des noms évocateurs et en respectant, autant que possible, l'unicité du rôle pour chaque variable.

Les caractères suivants peuvent être utilisés :

- lettres majuscules : de A à Z
- lettres minuscules : de a à z (les minuscules sont considérées comme des caractères différents des majuscules)
- chiffres, de 0 à 9
- les caractères trait souligné (underscore) : \_ et arobase : @

Exemple : nom des variables à retenir dans un problème de calcul de moyenne de notes :

Exemple : nom des variables à retenir dans un problème de calcul de moyenne de notes :

- `totalNotes` : total des notes d'un(e) élève
- `nombreNotes` : nombre des notes de cet(te) élève

JS 01

Réalisation : Guillaume DELACROIX Formateur AFPA

10 janvier 2023

**Afpa**

- `moyenneNotes` : moyenne des notes de cet(te) élève

Remarquez le style de nommage des variables : le premier mot est entièrement en lettre minuscules, le second mot commence par une majuscule; on appelle ça la convention *camelCase*.

Une variable est typée, c'est-à-dire qu'elle possède **un type** : elle peut représenter, par exemple, un nombre (entier ou décimal) ou une chaîne de caractères (nous verrons plus tard d'autres types).

Les chapitres suivants vous donneront plus de détails sur l'ensemble des types de données en Javascript.

Une variable peut être déclarée à tout moment à l'intérieur du corps du programme. Cependant, la déclaration et l'initialisation d'une variable doivent impérativement précéder la première utilisation de celle-ci. Il est également et généralement préférable de déclarer les variables en début de bloc (juste après l'accolade ouvrante) pour une question de lisibilité.

Chaque déclaration de variable est construite sur le modèle suivant :

```
Type nomDeLaVariable ;
```

Une variable peut être initialisée lors de sa déclaration :

```
type nomDeLaVariable = valeurInitiale;
```

Les variables ne sont visibles que dans le bloc d'instructions - bloc défini par des accolades { et } - dans lequel elles sont définies.

Comme toutes les instructions du langage Javascript, chaque déclaration de variable DOIT absolument être terminée par un point-virgule ;.

Le point-virgule ne constitue pas un séparateur, mais plutôt un terminateur d'instructions.

# Les instructions

Une instruction est une ligne de traitement.

Le caractère point-virgule ; est un terminateur. TOUTES les instructions doivent se terminer par un ;.

Les différents identificateurs sont séparés par un séparateur.

Les séparateurs peuvent être indifféremment l'espace, la tabulation et le saut de ligne.

# Les commentaires

Les commentaires dans un programme servent :

- à donner aux développeurs des indications sur le fonctionnement d'un programme
- à désactiver (temporairement) des blocs de code

**Les parties de code mises en commentaires ne sont pas exécutées.**

- Commentaires sur une seule ligne : on utilise les signes // :

```
// Demande le nom de famille
```

```
var nom = prompt("Entrez votre nom");
```

Dans cet exemple, on fournit une information complémentaire au code (1ère ligne); le code de la seconde ligne sera exécuté (demande à l'utilisateur d'entrer un nom).

- Commentaires sur plusieurs lignes : les caractères compris entre /\* et \*/ ne seront pas interprétés. On utilise cette notation quand il y a plusieurs lignes à commenter.

```
/* Demande le nom de famille
```

```
var nom = prompt("Entrez votre nom");
```

```
*/
```

Dans cet exemple, les 2 lignes sont en commentaires, cette fois le code de la seconde ligne ne sera **pas exécuté**.

# Les conventions d'écriture

Ces conventions ne sont pas des contraintes de syntaxe des langages informatiques. Elles n'existent que pour en faciliter la lecture et font partie d'une sorte de norme implicite que tous les bons développeurs s'obligent à respecter :

- Une seule instruction par ligne. Même si tout un programme peut-être écrit sur une seule ligne.
- Les délimiteurs d'un bloc { et } doivent se trouver sur des lignes différentes et être alignés sur la première colonne de sa déclaration.
- A l'intérieur d'un bloc {} les instructions sont indentées (décalées) par un caractère tabulation ou un espace.
- A l'intérieur d'un bloc {} la partie déclaration des variables et la partie instructions sont séparées par une ligne vide.