

Les variables

Bien que seule elle soit inutile, elle est indispensable dans tout programme un tant soit peu poussé car elle permet, tel votre mémoire, de retenir des informations pour y accéder à nouveau plus tard.

Important : Si vous effectuez un calcul mais n'enregistrez pas le résultat vous serez obligé de refaire toutes les opérations de ce calcul pour avoir à nouveau le résultat.

Pour créer une variable et qu'elle nous renvoie vers une valeur, on effectue ce qu'on appelle *l'initialisation*, c'est-à-dire qu'on va écrire comment notre variable doit être nommée suivi d'une signe « = » et finalement de la valeur qu'on veut retenir en mémoire. Nous utiliserons le mot-clef « var » dans notre cas pour indiquer que ce qui suit est une variable mais en pratique cela dépendra du langage utilisé.

Ex : **var temp = 28**

Si on veut changer ultérieurement cette valeur, pas de soucis, on peut refaire la même chose avec simplement une autre valeur mais sans avoir besoin du mot-clef « var », la seule différence étant qu'on appelle ceci une *assignation* désormais.

Ex : **temp = 35**

On peut aussi créer une variable mais décider qu'on lui affectera une valeur plus tard. C'est ce qu'on appelle *la déclaration*. Dans ce cas-là il faudra également utiliser le mot « var » pour que le langage comprenne qu'on veut créer une variable et qu'on n'est pas en train de lui demander la valeur de la variable (Voir plus bas comment récupérer la valeur d'une variable).

Ex : **var temp**

Dans le cas ci-dessus on a bien créé une variable mais elle ne renvoie pour le moment aucune valeur. Si on décide à un moment qu'elle doit faire le lien avec une valeur on utilisera l'affectation comme au-dessus afin de lui en attribuer une.

Stocker des valeurs est une chose mais on comment les récupère-t-on ? Eh bien simplement en tapant le nom de la variable votre programme va « remplacer » ce nom par sa valeur. Regardez les exemples ci-dessous :

5 + 5 // L'ordinateur va arriver au résultat 10

OU

var x = 5

5 + x // On arrive également à 10

En fait on peut se représenter une variable comme une étiquette qui se pose sur un tiroir ou une boîte dans lequel/laquelle on met ce qu'on veut. Et de plus c'est très proche de la réalité car une variable c'est un « panneau d'indication » qui envoie vers l'adresse de ce qu'on appelle des « emplacements mémoires », c'est-à-dire des endroits où l'information est stockée et qui ont chacun une adresse spécifique (Ce sont ces composants électriques qui retiennent des charges tel qu'on l'a vu dans la partie sur l'encodage).

Mais les variables, nous, on n'y met pas que des chiffres ! Cela serait plutôt dur de communiquer sinon. Et donc on peut stocker différents types d'informations dans les variables. Le nombre de types dépend du langage mais on ne va découvrir maintenant que quelques-uns universels. Les voici :

Le type numérique : Pas de surprise ici, ce sont les nombres. Certains langages en ont plusieurs (un type pour les nombres entiers, un pour les nombres décimaux, ...) mais dans notre cas on n'y prêtera pas attention pour le moment.

Ex : 7, -13

Le type alphanumérique : C'est ce qu'on utilise pour tous les caractères, symboles et autres. Généralement on le repère facilement car il est entouré de guillemets (soit doubles "" soit simples '). Attention ! Il y a une différence entre "35" et 35 par exemple, le premier est une suite de caractères ou symboles ce qui veut dire qu'on ne peut pas calculer avec tandis que le second est une valeur numérique. Dans la plupart des langages ce type est appelé String. (Chaîne en anglais)

Ex : "j", "Bonjour", "38", "45 + 72" (ce dernier ne sera pas calculé et donc l'ordinateur ne renvoie pas de réponse)

Le type booléen : Ce type est un peu spécial car il ne peut contenir que 2 valeurs possibles, soit true soit false (Donc « vrai » & « faux »). C'est un type qui nous sera très utile plus tard pour ajouter de la logique à notre programme.

A noter que lorsque on utilise le signe « = » c'est une affectation et non une égalité ! C'est toujours la valeur à droite du signe « = » qui est attribuée à la variable à gauche du signe « = », rien à voir avec les mathématiques donc.

Un autre point est qu'on peut affecter une variable à une autre variable cependant ce qui se passe alors, du moins pour les types « simples » tel que ceux qu'on a vu, est que la valeur de la variable à droite du signe « = » est copiée dans un autre emplacement mémoire, ou dans d'autres termes « une autre boîte », puis que la variable à gauche du signe « = » ira pointer vers cet emplacement.

Ex : **var a = 2**

var b = a // b possède donc la même valeur que la variable a

a = 4 // a vaut maintenant 4 mais b vaut toujours 2 car ils ne partagent pas les mêmes « boîtes »

Quizz

1) Comment affecter le numérique 3 à une variable appelée « day » ?

2) Comment affecter le symbole 3 à une variable appelée « day » ?

3) Comment déclarer une variable ne pointant vers aucune valeur ?

4) A votre avis que contient chaque variable à la fin ?

```
var student_1 = "John"
```

```
var student_2 = student_1
```

```
student_1 = "Marc"
```

5) Si je veux afficher le calcul « $41 - 17 * 2$ » à l'écran sans que ce soit son résultat qui apparaisse, et que je veux le stocker dans ma variable qui porte le nom « myCalculation » que dois-je écrire ?