Fiche n°2: Structure Générale d'un Programme
Les types de données

Algorithmique	Langage JavaScript
// Commentaire	/* *Commentaire* *Ne pas ajouter de commentaire en fin de ligne (une partie ne sera pas visible) mais plutôt avant la ligne de code.* *Seule exception où on utilise la fin de ligne: les commentaires après les déclarations de variable.*/
	// donner un titre à votre programme <script>
programme Exemple	
	// Constante **const** Max = 100 ;
CONST Max ←100	
	// Variables
VAR x, y : ENTIER nom : chaîne	**var** x, y ,nom;
	ou
DEBUT	**let** x, y ,nom; *// la portée est différente*
	/*
/* Corps du programme principal */	*Corps du programme principal*
	/
FIN	</script>

Les déclarations sont ici faites sur une seule ligne avec le préfixe `var` qui n'est pas une annonce de type. En JavaScript, le type a priori n'existe pas, les variables sont associées à un type lors des affectations de contenu.

Il faut se poser la question de savoir si la variable doit être disponible tout au long du script, ou si elle doit être limitée à la fonction dans laquelle elle a été créée. C'est ce que l'on appelle la portée des variables.

La chaîne de caractères doit être encadrée par des guillemets ou des apostrophes.
Exemple :
"ma constante texte" ou 'ma constante texte' sont des constantes chaînes de caractères.

Les variables booléennes acceptent les valeurs true ou false.

Infinity
Cette constante correspond à un nombre infini, supérieur à 1.7976931348623157E+10308 ou inférieur à -1.7976931348623157E+10308

Opérateurs booléens :

Algo	>	<	>=	<=	=	<>	et	ou	non
JavaScript	>	<	>=	<=	==	**!=**	**&&**	||	**!**

Les mots réservés sont des mots correspondant généralement à des objets, propriétés ou méthodes déjà utilisés par JavaScript et qui ne peuvent, donc, pas recevoir de valeurs. Voici un tableau qui récapitule ces mots interdits. Certains de ces mots sont d'ores et déjà interdits, d'autres, même s'il est encore possible de les utiliser, sont déconseillés car les prochaines versions de JavaScript ne les accepteront plus.

abstract	float	short
boolean	for	static
break	function	super
byte	goto	swich
case	if	synchronized
char	implements	this
class	import	throw
const	in	throws
continue	instanceoff	transient
debugger	int	true
default	interface	try
delete	long	typeoff
do	native	var
double	new	void
else	null	volatile
enum	package	while
export	private	with
extends	protected	