

Programmation générale

Les mots-clés et fonctions suivantes sont utilisés dans la plupart des langages de programmation. Le syntaxe varie peu d'un langage à l'autre, ce qui facilite l'apprentissage d'autres langages plus tard.

Opérateurs arithmétiques

Nom	Symbole	Exemple
Addition	+	somme = nombre1 + nombre2
Soustraction	-	différence = nombre1 - nombre2
Multiplication	*	produit = nombre1 * nombre2
Division	/	quotient = nombre1 / nombre2
Modulo (reste de la division entière de deux nombres)	%	reste = nombre1 % nombre2

Opérateurs logiques

Certains de ces opérateurs peuvent être notés différemment dans d'autres langages de programmation. Nous nous en tenons, ici, à la notation javascript.

Nom	Symbole	Exemple
AND (et)	&& (shift + 6)	vrai && vrai = vrai faux && faux = faux faux && vrai = faux vrai && faux = faux
OR (ou)	 (alt gr + 7)	vrai vrai = vrai faux faux = faux vrai faux = vrai faux vrai = vrai
NOT (non, négation)	! (point d'exclamation)	!vrai = faux !faux = vrai
Égalité faible	==	2 == 2 (vrai) 2 == 1 (faux) caractère == chaîne (vrai) undefined == null (vrai)
Égalité stricte	===	caractère === chaîne (faux) undefined === null (faux)

Strictement plus grand que	>	2 > 1 (vrai) 2 > 8 (faux) 2 > 2 (faux)
Strictement plus petit que	<	2 < 8 (vrai) 2 < 1 (faux) 2 < 2 (faux)
Plus grand ou égal	>=	2 >= 2 (vrai) 2 >= 1 (vrai) 2 >= 3 (faux)
Plus petit ou égal	<=	2 <= 2 (vrai) 2 <= 3 (vrai) 2 <= 1 (faux)
N'est pas égal	!=	2 != 2 (faux) 2 != 6 (vrai)

Variables

Une variable se définit avec le mot-clé “var”. Par exemple :

```
var message;
```

Crée une variable vide nommée “message”. Les variables peuvent être de différents types, mais en javascript il n'est pas nécessaire de le préciser avant de leur affecter une valeur. Dans d'autres langages de programmation, il peut être nécessaire de spécifier le type de variable lors de la création.

Types de variables

Nom	Symbole	Exemple
Nombre entier	int (integer)	nombre = 2; ("nombre" est un nombre entier)
Nombre décimal (en virgule flottante)	float (floating point)	nombre = 1.2; nombre = 2/3; ("nombre" est un nombre décimal)
Caractère	char	caractere = "a"; caractere = "23"; (si on ne met pas de guillemets a est une variable et 23 est un entier)
Chaîne de caractères	string	chaine = "bonjour, j'ai 27

		ans" (la taille n'est pas limitée. Tout écrire entre guillemets)
Booléen	bool	logique = true; logique = false;
Tableau	array	tableau = ["premier element", 24, true]; tableau[0] = "premier element" tableau[1] = 24 tableau[2] = true (le tableau commence à 0, pas à 1)

Opérations spéciales avec les variables

Nom	Symbole	Exemple
Concaténation (colle deux éléments l'un contre l'autre en une chaîne de caractères)	+	"a" + 23 = a23 (si on utilise deux nombres, on obtiendra le résultat d'une addition. Un nombre et un caractère transforme le tout en chaîne) "Salut, " + "ça va ?" = "Salut, ça va ?"
Égalité	==	"a" == "a" (vrai) "a" == "A" (faux)
Ajouter à la fin du tableau	push()	tableau.push("element1", "element2"); Pas de limite d'éléments.
Retirer le dernier élément du tableau	pop()	tableau.pop() retire le dernier élément du tableau. Si le tableau est vide, retourne la valeur "undefined"
Ajouter au début du tableau	unshift()	tableau.unshift("element1", "element2"). Si plusieurs éléments sont spécifiés, ils sont ajoutés

		dans l'ordre dans lequel vous les avez écrits.
Retirer le premier élément du tableau	shift()	tableau.shift() retire le premier élément du tableau. Si le tableau est vide, retourne la valeur "undefined"
Convertir un nombre en chaîne de caractères	nombre.toString()	il est possible de convertir le nombre dans la base voulue (10 par défaut) toString(2) toString(8) toString(16)
Convertir une chaîne en nombre	parseInt(chaine) parseFloat(chaine)	On ne peut convertir que des chiffres en nombres. parseInt("bonjour") retourne "NaN" (not a number).

Fonctions

Les fonctions sont définies avec le mot-clé "function" suivie de parenthèses contenant aucun ou plusieurs paramètres puis un code qui compose la fonction entre accolades.

```
function nom_de_fonction (paramètres) {
```

```
    /*votre code ici*/
```

```
}
```

Fonctions d'interaction avec le DOM

Les fonctions suivantes sont des fonctions de base de javascript qui servent à interagir avec une page HTML. Les majuscules sont importantes.

Nom	Utilisation	Explication
document	document.fonction	"document" est le premier mot clé que l'on utilise dans une commande. Il désigne la page web (le document) où est exécuté la fonction.
getElementById	document.getElementById("id_d-une_balise")	Permet de sélectionner la balise avec l'id spécifié.

querySelector	<pre>document.querySelector("#id") document.querySelector(".class")</pre>	<p>Permet de sélectionner une balise par son id ou sa classe. utiliser le “#” pour un id et le “.” pour la classe, comme en CSS.</p> <p>S'utilise comme getElementById.</p>
innerHTML	<pre>document.getElementById("id").innerHTML = "code html";</pre>	<p>Ajoute du contenu à la balise sélectionnée. Cela peut-être du texte, des nombres, un tableau ou des balises html. à part les nombres, tout doit être écrit entre guillemets avec le symbole de concaténation si besoin.</p>
style	<pre>document.getElementById("id").style</pre>	<p>Permet de changer le style de la balise sélectionnée. Les commande CSS s'ajoutent à la commande comme ci-dessous.</p>
style.display	<pre>document.getElementById("id").style.display = "none" ou "block" ou "inline" ou "inline-block"</pre>	<p>Change l'affichage de la balise. “none” la rend invisible, “block” l'affiche avec un retour à la ligne, “inline” l'affiche sans retour à la ligne et “inline-block” est un comportement particulier d'un block sans retour à la ligne.</p>
style.color	<pre>document.getElementById("id").style.color = "blue" ou "#2548aF" ou "rgb(124,255,14)"</pre>	<p>Change la couleur du texte contenu dans la balise sélectionnée. Les couleurs sont définies par un mot-clé (voir sur internet ou aide-memoire_CSS), un code hexadécimal ou des valeurs RGB.</p>
style.background-color	<pre>document.getElementById("id").style.background-color = "blue"</pre>	<p>Change la couleur de fond de la balise. mêmes règles que la couleur du texte ci-dessus</p>
<p>Pour plus de possibilités de changement de style, veuillez consulter le document aide-memoire_CSS. La règle est la même à chaque fois : style.”nom du paramètre” = “valeur”</p>		
addEventListener	<pre>document.getElementById("id").addEventListener("click", fonction())</pre>	<p>Attend qu'une interaction de type “click” soit effectuée sur la balise sélectionnée et exécute la fonction associée. Il est possible de définir une fonction anonyme dans cette commande:</p> <pre>addEventListener("click", fonction(){ /*votre code*/ });</pre> <p>Mais faites attention aux parenthèses et à la lisibilité du code. préférez créer une fonction ailleurs et l'appeler ici par</p>

		son nom.
Math.random()	nombre = Math.random()	Génère un nombre aléatoire non entier entre 0 et 1. 1 n'est pas inclus. Résultat minimal = 0 Résultat maximal = 0.9999999
Math.floor()	nombre = Math.floor(nombre)	Retourne le plus grand entier inférieur ou égal à nombre.
Math.ceil()	nombre = Math.ceil(nombre)	Retourne le plus petit entier supérieur ou égal à nombre.
Math.PI	nombre = Math.PI	Le nombre pi (~3.1415926535). Le ratio entre le périmètre d'un cercle et son diamètre
Date	date = new Date(); date.getDate()	Plusieurs exemples d'utilisation ici