DOCUMENT ELEVE / JAVASCRIPT

**Fiche mémo**

**1**[**.**](#_gjdgxs) **Sommaire**

[**2.** **Les commentaires**](#_gjdgxs)

[**3.** **Les variables**](#_30j0zll)

[**4.** **Les opérateurs**](#_1fob9te)

[**5.** **Les conditions**](#_3znysh7)

[**6.** **Les boucles**](#_2et92p0)

[**7.** **Les fonctions**](#_tyjcwt)

[**8.** **Les exceptions**](#_3dy6vkm)

[**9.** **Les affichages**](#_1t3h5sf)

[**10.** **Les évènements**](#_4d34og8)

[**11.** **L’objet window**](#_2s8eyo1)

[**12.** **L’objet String**](#_17dp8vu)

[**13.** **L’objet Date**](#_3rdcrjn)

[**14.** **L’objet Array**](#_26in1rg)

[**15.** **L’objet Math**](#_lnxbz9)

[**16.** **Le DOM (Document Object Model)**](#_35nkun2)

# Les commentaires

*// une ligne en commentaire*

*/\* plusieurs lignes*

*en commentaire \*/*

# Les variables

Les variables peuvent être déclarées de façon explicites (conseillé) : *var numero = 1*

ou implicites : *numero = 1*

Les variables ne sont pas typées. Elles prennent le type de la donnée qui leur est

affectée.

Variables globales : ce sont soit les variables déclarées tout au début du script, en dehors et avant toutes fonctions, soit les variables déclarées dans une fonction sans le mot clé var

# Les opérateurs

Opérateur d’affectation : = prend pour valeur

Opérateurs de calcul : + plus

- moins

\*multiplié

/ divisé

% modulo (reste)

Opérateurs logiques : && et

||ou

Opérateurs d’incrémentation : x++ équivalent à x=x+1

x-- équivalent à x=x-1

Opérateurs associatifs : += : x+=y équivalent à x=x+y

-=  : x-=y équivalent à x=x-y

\*= : x\*=y équivalent à x=x\*y

/= : x/=y équivalent à x=x/y

Opérateurs de comparaison : == égal

< inférieur

<= inférieur ou égal

> supérieur

>= supérieur ou égal

!= différent

=== strictement égal : la comparaison porte

en plus sur le type des variables (renvoie vrai si

les opérateurs sont égaux et de même type) (conseillé)

!== strictement différent : la comparaison

porte en plus sur le type des variables (renvoie

vrai si les opérateurs ne sont pas égaux ou pas

de même type) (conseillé)

# Les conditions

Syntaxe des conditions : *if (var1==var2)*

*{*

*...............;*

*}*

*else*

*{*

*...............;*

*}*

On peut utiliser les opérateurs logiques && (et) et || (ou)

On peut aussi utiliser : *(expression) ? instruction a : instruction b*

équivalent à *if (expression)*

*{ instruction a }*

*else*

*{ instruction b }*

switch – case : *switch (expression) {*

*case valeur1:*

*instructions1;*

*[break;]*

*case valeur2:*

*instructions 2;*

*[break;]*

*...*

*default:*

*instructions-par-défaut*

*[break;]*

*}*

# Les boucles

La boucle for : *for ( i=0;i<10; i++ )*

*{*

*document.write ("ligne : " + i);*

*}*

La boucle while : *while ( i<10 )*

*{*

*document.write ("ligne : " + i);*

*i++;*

*}*

La boucle do while : *do*

*{*

*document.write ("ligne : " + i);*

*i++;*

*} while ( i<10 ) ;*

# Les fonctions

Déclaration d’une fonction : *function nomfonction()*

*{ .. code des instructions ... }*

Appel de la fonction : *nomfonction() ;*

Les fonctions peuvent admettre des arguments (des paramètres) :

*function afficher(texte)*

*{*

*alert(texte);*

*}*

L’appel se fera alors : *afficher("Bonjour à tous") ;*

Les fonctions peuvent avoir une valeur de retour, avec return :

*function carré(nb)*

*{*

*var carré = nb\*nb;*

*return nb;*

*}*

L’appel se fera alors : *result=carré(10) ;*

Les paramètres de type simple sont en entrée uniquement (passage par valeur)

Les objets sont passés en entrée/sortie (par référence)

# Les exceptions

Pour gérer les exceptions : try {

*instructions\_try;*

}

catch (*exception1*) {

*instructions\_catch\_1;*

}

catch (*exception2*) {

*instructions\_catch\_2;*

}

…

finally {

*instructions\_finally;*

}

# Les affichages

dans une popup : *alert("votre message");*

Dans la console du navigateur **:** *console.log("votre message");*

confirmation**:** *confirm("voulez-vous continuer");*Afficher du texte dans la page HTML : *document.write("votre texte");*

*document.writeln("votre texte");*

On peut associer du texte et des variables :

*document.write("Bonjour " + prenom);*

On peut utiliser des balises HTML :

*document.write("<B>Le résultat est </B>" +resultat);*

# Les évènements

Les évènements permettent l’interactivité des pages web

Ils sont déclenchés par : *onevenement="fonction() ;"*

*Par ex : onClick="alert(‘Bonjour') ;"*

Les principaux évènements :

• *onLoad* : se déclenche à la fin du chargement de la page

• *onUnload* : se déclenche lors de la fermeture de la page

• *onBeforeunload* : se déclenche juste avant la fermeture de la page

• *onClick* : se déclenche lors du clic de la souris

• *onDoubleclick* : se déclenche lors d’un double click

•*onmouseOver* : se déclenche lorsque la souris passe dessus un lien ou une image (sans cliquer)

• *onmouseOut* : se déclenche lorsque la souris quitte la zone sensible (lien ou image)

• *onmouseDown* : se déclenche lorsque l’on presse la souris

• *onmouseUp* : se déclenche lorsque l’on lâche la souris

• *onmouseMove* : se déclenche lorsque l’on bouge la souris

• *onFocus* : se déclenche lorsque le champ de saisie obtient le focus

• *onBlur* : se déclenche lorsque le champ de saisie perd le focus

• *onkeyDown* ou *onkeyPressed* se déclenche lorsque l’on presse une touche du clavier

• *onkeyUp* : se déclenche lorsque l’on lâche une touche du clavier

• *onChanged* : se déclenche lorsque le champ perd le focus et que son contenu a été changé

• *onSelect* : quand le contenu d’un champ est sélectionné

• *onSubmit* : se déclenche juste avant l’envoi du formulaire

• *onReset* : se déclenche lors de la réinitialisation du formulaire

# L’objet window

Les principaux attributs et méthodes :

• *alert()* : affiche un message.

• *confirm()* : affiche une demande de confirmation.

• *prompt ()* : affiche un prompt de saisie

• *setTimeout()* : permet de démarrer un timer

• *clearTimeout()* : stoppe le timer

• *status* : valeur du texte affiché dans la barre d'état de la fenêtre.

• *defaultStatus* : valeur par défaut qui s'affiche dans la barre d'état.

• *open()* : permet d’ouvrir une nouvelle fenêtre

• *close()* : permet de fermer la fenêtre en cours

• *location* : Objet l’URL de la fenêtre courante (*window.location.href* est la valeur de l’URL)

• *history* : Objet URL précédement visitées.

• *window.back()*: retourne sur l’URL précédente)

• *self* : représente la fenêtre en cours

# L’objet String

Les principaux attributs et méthodes :

• *length* : longueur de la chaîne de caractères.

• *charAt()* : méthode qui permet d'accéder à un caractère isolé d'une chaîne.

• *indexOf()* : méthode qui renvoie la position d'une chaîne partielle à partir d'une position donnée (en commençant au début de la chaîne principale soit en position 0).

• *LastIndexOf()* : méthode qui renvoie la position d'une chaîne partielle à partir d'une position déterminée (en commençant à la fin soit en position length moins 1).

• *substring(x,y)* : méthode qui renvoie un string partiel situé entre la position x et la position y-1.

• *toLowerCase()* : Transforme toutes les lettres en minuscules.

• *toUpperCase()* : Transforme toutes les lettres en majuscules.

# L’objet Date

Les principaux attributs et méthodes :

• *New Date();* : donne la date et l’heure du poste client.

• *getYear()* : retourne l’année (sur 2 car).

• *getMonth()* : retourne le mois

• *getDate()* : retourne le jour

• *getDay()* : retourne le jour de la semaine (0 pour dimanche)

• *getHours()* : retourne l’heure

• *getMinutes()* : retourne les minutes

• *getSeconds()* : retourne les secondes

# L’objet Array

Les principaux attributs et méthodes :

• *tab = new Array (x);* : crée le tableau

• *tab[i] = "élément";* : remplit la ième ligne du tableau

• *length* : longueur du tableau

• *sort() :* trie le tableau

• *reverse()* : inverse les éléments

• *join()* : renvoie tous les éléments dans une chaine, séparés par des virgules (ou par un séparateur passé en argument)

# L’objet Math

• *Math.abs (x);* : valeur absolue

• *Math.round (x);* : arrondit x à l’entier le plus proche

• *Math.max (x,y);* : maximum de x et y

• *Math.min (x,y);* : minimum de x et y

• *Math.random()* : valeur aléatoire

• *Math.sqrt(x)* : racine carrée

• *Math.pow(x,y)* : x puissance y

• *Math.parseInt(x);* : convertit en entier

• *Math.parseFloat(x);* : convertit en float (attention pour avoir uniquement 2 chiffres après la virgule : x = Math.round(x\*100)/100

• *Math.sin(x), Math.cos(x),…* : Fonctions trigonométriques

• *Math.log(x), Math.exp(x),…* : Fonctions logarithmiques

# Le DOM (Document Object Model)

Accès aux éléments du document

par id : *getElementById() ,*

par nom :  *getElementsByName()*

par nom de balise : *getElementsByTagName()*

récupèrer ou définir le contenu d'un élément donné :

*var contenu = element.innerHTML;*

ou *element.innerHTML = contenu;*

Les propriétés de chaque nœud du document (objet Node) :

• *attributes* : tableau des attributs disponibles du noeud

• *childNodes* ; liste des noeuds enfants

• *data* : données des noeuds texte

• *firstChild* : premier noeud enfant d'un noeud

• *lastChild* : dernier noeud enfant d'un nœud

• *nextSibling*: prochain noeud de même niveau

• *previousSibling*: noeud précédent de même niveau   
• *nodeName*: nom du noeud

• *nodeType*: type du noeud

*1 Noeud élément*

*2 Noeud attribut*

*3 Noeud texte*

*4 Noeud pour passage CDATA*

*5 Noeud pour référence d'entité*

*6 Noeud pour entité*

*7 Noeud pour instruction de traitement*

*8 Noeud pour commentaire*

*9 Noeud document*

*10 Noeud type de document*

*11 Noeud de fragment de document*

*12 Noeud pour notation*

• *nodeValue*: valeur/contenu du noeud

- *Pour les noeuds texte, c'est le texte,*

*- pour les noeuds attribut la valeur affectéeà l'attribut,*

*- pour les noeuds élément, valeur null*

• *parentNode*: noeud parent

Les principales méthodes de chaque nœud du document (objet Node) :

• *getAttribute()*: rechercher la valeur d'un nœud attribut

• *hasChildNodes()*: vérifier l'existence de nœuds enfants

• *appendChild()*: ajouter un noeud enfant

• *insertBefore()*: insérer un noeud avant

• *appendData()*: ajouter des données en caractères

• *insertData()*: insérer des données en caractères

Les principales méthodes du document (objet Document) :

• *createAttribute()*: créer un noeud d'attributs

• *createElement()* : créer un noeud d'éléments

• *createTextNode()*: créer un noeud de texte