

JavaScript

Mémo

Voici un **petit mémo** contenant l'essentiel de ce qu'il faut retenir en javascript. Tout n'y est pas bien évidemment, le but est de retrouver les termes et concepts les plus **communément utilisées**.

Javascript

Les objets JS primitifs

window

propriétés

- **window.name** : nom de la fenêtre
- **window.statut** : texte de la barre de statut

méthodes

- **window.open()** : nouvelle fenêtre (onglet)
- **window.alert()** : message dans une pop-up
- **window.confirm()** : nouvelle fenêtre de choix (true/false)
- **window.prompt()** : nouvelle fenêtre de saisie
- **window.setTimeout()** : déclencher une minuterie
- **window.clearTimeout()** : suspendre une minuterie
- **window.setInterval()** : lancer un traitement à intervalle régulier
- **window.clearInterval()** : suspendre un traitement à intervalle régulier

sous-objets

- **window.document** : document actif
- **window.location** : informations sur la barre d'adresse
- **window.history** : naviguer dans l'historique

Javascript

Les objets JS primitifs

document

propriétés

- *document.title* : titre de la page
- *document.body* : corps du document
- *document.cookie* : gestion des cookies

méthodes

- *document.getElementById()* : récupérer un élément HTML par son ID
- *document.getElementsByTagName()* : idem mais par son tag (nom de balise)
- *document.getElementsByClassName()* : idem mais par sa classe CSS
- *document.write()* : écrire dans le document

sous-objets

- *document.forms* : tableau des formulaires de la page
- *document.images* : tableau des images de la page

Javascript

Les objets JS primitifs

location

propriétés

- *location.href* : URL actuelle complète
- *location.host* : nom de domaine de l'URL actuelle
- *location.pathname* : chemin de l'URL actuelle sans le nom de domaine

méthodes

- *location.reload()* : recharger la page

history

propriétés

- *history.length* : nombre d'URLs stockées dans l'historique

méthodes

- *history.back()* : revenir à la page précédente
- *history.forward()* : aller à la page suivante

Javascript

Les objets JS primitifs

Date

méthodes

- *Date.getDate()* : jour dans le mois (1 à 31)
- *Date.getDay()* : jour dans la semaine (0 dimanche à 6 samedi)
- *Date.getMonth()* : mois dans l'année (0 janvier à 11 décembre)
- *Date.getFullYear()* : année sur 4 caractères
- *Date.getYear()* : année sur 2 caractères
- *Date.getHours()* : heure dans la journée (0 à 23)
- *Date.getMinutes()* : minutes dans l'heure (0 à 59)
- *Date.getSeconds()* : secondes dans la minute (0 à 59)
- *Date.getTime()* : nombre de millisecondes écoulées depuis le 1er janvier 1970 (aussi appelé le "timestamp")

Javascript

Les objets JS primitifs

Math

propriétés

- **Math.E** : constante exponentielle
- **Math.PI** : constante PI
- **Math.SQRT2** : constante racine carré de 2

Méthodes

- **Math.abs()** : retourne la valeur absolue d'un nombre réel
- **Math.round()** : retourne l'arrondi (entier) d'un nombre réel
- **Math.floor()** : retourne l'entier le plus proche inférieur d'un nombre réel
- **Math.ceil()** : retourne l'entier le plus proche supérieur d'un nombre réel
- **Math.pow()** : retourne un nombre à une certaine puissance
- **Math.random()** : génère un nombre aléatoire entre 0 et 1
- **Math.max()** : compare 2 nombres et renvoi le plus grand
- **Math.min()** : compare 2 nombres et renvoi le plus petit

Javascript

Les objets JS primitifs

String

propriétés

- *String.length* : retourne la longueur d'une chaine

Méthodes

- *String.charAt()* : retourne le caractère à la position précisée
- *String.indexOf()* : retourne la position d'une sous-chaine
- *String.slice()* : extrait une sous-chaine
- *String.substr()* : variante de slice()
- *String.split()* : fragmente une chaine en éléments d'un tableau
- *String.toLowerCase()* : retourne la chaine en minuscule
- *String.toUpperCase()* : retourne la chaine en majuscule
- *String italics()* : retourne la chaine en italique
- *String.bold()* : retourne la chaine en gras

Javascript

Les objets JS primitifs

Array

propriétés

- ***Array.length*** : retourne le nombre d'éléments d'un tableau

Méthodes

- ***Array.join()*** : converti un tableau en chaîne
- ***Array.pop()*** : supprime le dernier élément d'un tableau
- ***Array.push()*** : ajoute un élément en fin de tableau
- ***Array.shift()*** : supprime le premier élément d'un tableau
- ***Array.unshift()*** : ajoute un élément en début de tableau
- ***Array.reverse()*** : inverse l'ordre des éléments d'un tableau
- ***Array.sort()*** : tri les éléments d'un tableau par ordre croissant

Javascript

Les objets JS primitifs

Images

propriétés

- *images.length* : retourne le nombre d'image chargées dans la page
- *images.src* : retourne l'adresse et le nom de l'image
- *images.alt* : infobulle de la balise IMG
- *images.fileSize* : poids de l'image en octets
- *images.width* : largeur de l'image
- *images.height* : hauteur de l'image
- *images.complete* : indicateur de fin de chargement (true/false)

Javascript

Les fonctions spéciales

fonctions spéciales

- ***eval()*** : évalue et exécute le code contenu dans une chaîne
- ***isFinite()*** : teste si le paramètre est un nombre fini (true/false)
- ***isNaN()*** : teste si le paramètre n'est pas un nombre (true/false)
- ***parseFloat()*** : convertit une chaîne en décimal
- ***parseInt()*** : convertit une chaîne en entier
- ***escape()*** : encode les caractères spéciaux
- ***unescape()*** : décode les caractères spéciaux

NOTE : Lorsque l'on veut convertir une valeur numérique en chaîne il n'y a pas de fonction particulière aussi en général on ajoute une chaîne de caractère vide à notre valeur numérique ce qui revient à une concaténation et renvoi donc en résultat une chaîne de caractères.

Javascript

Opérateurs & Structures

opérateurs

- **function** : déclarer une fonction
- **new** : permet de construire un objet (Array, Date...)
- **this** : pointe l'objet en cours de manipulation
- **var** : déclarer une variable

structures

- **for** : déclarer une boucle conditionnelle "pour"
- **while** : déclarer une boucle conditionnelle "tant que"
- **in** : permet de parcourir les propriétés d'un objet
- **break** : force la sortie d'une boucle conditionnelle
- **if else** : déclarer un test conditionnel
- **switch case** : déclarer un test "multiple"
- **return** : retourne le résultat d'une fonction
- **try catch** : gestion des erreurs

JavaScript

Opérateurs & Structures

opérateurs de calcul

- **+** : addition
- **-** : soustraction
- ***** : multiplication
- **/** : division
- **%** : modulo (reste d'une division)
- **=** : affectation

opérateurs d'affectation avancés

- **+=** : ajoute la partie à droite du signe à l'opérande de gauche
- **-=** : soustrait la partie à droite du signe à l'opérande de gauche
- ***=** : multiplie l'opérande de gauche par la valeur à droite
- **/=** : divise l'opérande de gauche par la valeur à droite
- **%=** : calcule le reste de la division de l'opérande de gauche par la valeur à droite

NOTE : Pour ces opérateurs avancés, le résultat est stocké dans l'opérande de gauche (qui est en quelque sorte mis à jour)

JavaScript

Opérateurs & Structures

opérateurs d'incrémentation

- **++** : incrémentation de 1 de la variable
- **--** : décrémentation de 1 de la variable

opérateurs de comparaison

- **==** : égalité (ATTENTION, ne pas confondre avec le symbole d'affectation)
- **!=** : différence
- **===** : identité ou égalité stricte (teste la valeur mais aussi le type)
- **!==** : non identité ou inégalité stricte
- **<=** : infériorité
- **<** : infériorité stricte
- **>=** : supériorité
- **>** : supériorité stricte

opérateurs logiques

- **&&** : ET logique
- **||** : OU logique
- **!** : NON logique