GÉNIE INFORMATIQUE

JAVASCRIPT

FICHE EXPRESS

Pour **Awa Diop**

SOMMAIRE

1. Introduction à JavaScript
2. Syntaxe, Variables et Opérateurs
3. Structures, Fonctions et Tableaux
4. Les Objets en JavaScript
5. Les Événements en JavaScript

© 2026 Momo Lab — un produit Momo Creator Studio · Partage et reproduction interdits · N° EXEMPLE

1. Intégrer JS dans une page HTML

À retenir

JavaScript est le langage de programmation Web Client qui gère le comportement dynamique des pages.
Les 3 langages de base du Web : HTML (structure), CSS (mise en forme), JavaScript (comportement).
Créé par Brendan Eich en 1995. Nom officiel : ECMA-262 (standard ECMAScript).

1. ① Dans <body> : code exécuté directement au chargement de la page.
2. ② Dans <head> : code inséré dans des fonctions, exécuté lors d'un événement.
3. ③ Fichier externe .js : relié via l'attribut src dans la balise <script src="script.js"></script>.

Astuce

Avantages du fichier externe .js :

- Sépare le code JS du code HTML.
- Rend le code plus lisible et facile à maintenir.
- Accélère le chargement grâce à la mise en cache.

Attention : un fichier .js externe ne contient JAMAIS de balises <script>.

2. Méthodes d'affichage et de saisie

Méthode / Propriété	Usage
window.alert(msg)	Affiche une boîte de dialogue d'alerte avec un message.
document.write(txt)	Écrit directement dans le flux HTML (utilisé pour les tests).
element.innerHTML	Modifie ou lit le contenu HTML d'un élément.
console.log(val)	Écrit dans la console du navigateur (utilisé pour le débogage).

Méthode de saisie	Usage	Valeur retournée
window.confirm(msg)	Demande une confirmation (Oui/Non)	true (OK) ou false (Annuler)
window.prompt(msg)	Ouvre une boîte de saisie de texte	Chaîne saisie ou null

Attention

`document.write()` supprime tout le contenu HTML existant s'il est exécuté après le chargement complet de la page. À utiliser uniquement pour des tests rapides.

1. Syntaxe de base et Identificateurs

1. Chaque instruction se termine obligatoirement par un point-virgule (;).
2. Sensibilité à la casse : `lastName` et `lastname` sont deux variables complètement différentes.
3. Identificateurs : doivent commencer par une lettre, un trait de soulignement (_) ou un signe ($).
4. Interdictions : les chiffres au début, les caractères accentués/spéciaux et les mots réservés (ex: JavaScript).
5. Style de nommage préconisé : camelCase (ex: `nomDeFamille`). Le trait d'union (-) est interdit (réservé à la soustraction).

Attention

Mode Strict ('use strict') :

Déclaré au début d'un script ou d'une fonction, il force des règles d'écriture strictes et interdit l'utilisation de variables non déclarées.

À retenir

Commentaires en JS :

- Sur une ligne : // commentaire
- Sur plusieurs lignes : /* commentaire */

2. Déclaration et Portée des Variables

Mot-clé	Portée	Modifiable ?	Règles & Spécificités
var	Globale ou Fonction	OUI	Sujet au Hoisting (les déclarations var sont hissées en haut du bloc).
let	Bloc { }	OUI	Connue uniquement dans le bloc où elle est déclarée.
const	Bloc { }	NON	Valeur constante fixe, ne peut pas être réassignée.

Astuce

- Une variable déclarée sans valeur a automatiquement la valeur et le type `undefined` .
- JavaScript est dynamiquement typé : une même variable peut changer de type au cours de l'exécution (ex: `var x = 5; x = 'Jean';`).

3. Types de données et Opérateurs

Type	Exemple de déclaration	typeof retourne
String	var s = "Bonjour"; var t = 'Salut';	"string"
Number	var x = 42; var y = 3.14; var z = 123e5;	"number"
Boolean	var b = true; var f = false;	"boolean"
Array	var tab = ["Saab", "BMW"];	"object"
Object	var p = {nom:"Faye", age:20};	"object"
undefined	var x;	"undefined"
null	var x = null;	"object"
Function	function f() {}	"function"

Attention

Comparaison entre null et undefined :

- `null == undefined` → true (valeurs équivalentes sans vérification du type)
- `null === undefined` → false (types différents : object vs undefined)

4. Opérateurs et Opérations sur les Chaînes

1. Égalité simple (==) : compare la valeur après conversion automatique de type.
2. Égalité stricte (===) : compare la VALEUR et le TYPE (recommandé pour éviter les erreurs).
3. Addition + chaîne : additionner un nombre et une chaîne convertit le nombre en chaîne (ex: `5 + '3'` → `'53'`).
4. Longueur d'une chaîne : `ch.length`.
5. Caractère à l'indice n : `ch.charAt(n)`.
6. Extraction de sous-chaîne : `ch.substring(debut, fin)` (l'indice fin est exclu).

1. Structures de contrôle et de répétition

Structure	Syntaxe
If … Else	if (a >= 0) { alert('positif'); } else { alert('négatif'); }
Switch case	switch(n) { case 1: alert('Un'); break; default: alert('Autre'); }

Boucle	Syntaxe
for	for (var i = 0; i < n; i++) { … }
while	while (condition) { … }
do … while	do { … } while (condition);

2. Fonctions et Fonctions prédéfinies

Définition

Fonction : Bloc de code nommé et réutilisable, déclenché par un appel explicite ou un événement.

Syntaxe : function nomFonction(param1, param2) { … return valeur; }

Une fonction peut être créée avec ou sans paramètres, et avec ou sans valeur de retour.

Attention

L'appel d'une fonction sans parenthèses (ex: `maFonction`) renvoie la définition du code source de la fonction et ne l'exécute pas !

Fonction prédéfinie	Rôle	Exemple & Résultat
parseInt(str, base)	Convertit une chaîne en entier dans la base spécifiée (2, 8, 10, 16)	parseInt('30.75', 10) → 30
parseFloat(str)	Convertit une chaîne en nombre décimal	parseFloat('3.14') → 3.14
isNaN(val)	Teste si la valeur n'est PAS un nombre (Not a Number)	isNaN('abc') → true
isFinite(val)	Teste si la valeur est un nombre fini (non infini, non NaN)	isFinite(240) → true
eval(str)	Exécute du code JavaScript contenu sous forme de chaîne	eval('2 + 3 * 4') → 14

3. Tableaux (Arrays)

À retenir

- Un tableau est une variable contenant plusieurs valeurs de types similaires ou distincts.
- Les indices d'un tableau commencent toujours à zéro (0).
- `typeof tableau` renvoie "object". JavaScript ne supporte PAS les tableaux associatifs.

Méthode / Propriété	Rôle	Exemple de code	Résultat
length	Donne le nombre d'éléments	voitures.length	3
push(val)	Ajoute un élément à la fin du tableau	voitures.push('Kia')	['Saab', 'Volvo', 'BMW', 'Kia']
pop()	Supprime et renvoie le dernier élément	voitures.pop()	['Saab', 'Volvo']
sort()	Trie le tableau en ordre alphabétique	voitures.sort()	['BMW', 'Saab', 'Volvo']
indexOf(val)	Trouve la position d'un élément (-1 si absent)	voitures.indexOf('Volvo')	1
join(sep)	Fusionne tous les éléments en une chaîne	voitures.join(' - ')	"Saab - Volvo - BMW"
concat(tab2)	Fusionne deux tableaux distincts	voitures.concat(['Kia'])	['Saab', 'Volvo', 'BMW', 'Kia']
forEach(fn)	Applique une fonction à chaque élément	voitures.forEach(afficher)	Parcours complet du tableau

Document personnel de Awa Diop · N° EXEMPLE · Partage interdit

1. Notions Fondamentales et Création d'objets

Définition

Un Objet est une entité qui regroupe des propriétés (ce qu'il est / ses données) et des méthodes (ce qu'il fait / ses fonctions).

Syntaxe : var nomObjet = { propriete1: valeur1, methode1: function() { … } };

Accès aux propriétés : nomObjet.nomPropriete OU nomObjet['nomPropriete']

Accès aux méthodes : nomObjet.nomMethode()

```
// Exemple de déclaration et parcours d'un objet
var personne = { prenom: "Momo", nom: "SECK", age: 25 };
var text = "";
for (var x in personne) {
  text += personne[x] + " ";  // Parcourt toutes les propriétés
}
```

Attention

Évitez d'instancier des objets avec `new String()`, `new Number()` ou `new Boolean()`. Cela complique inutilement le code et ralentit l'exécution.

2. Objets Natifs de JavaScript

Objet Math (Static)	Usage / Description
Math.PI / Math.LN2	Propriétés de constantes numériques (~3.14159 / ~0.693)
Math.round(x)	Arrondi à l'entier le plus proche
Math.random()	Renvoie un nombre aléatoire entre 0 et 1
Math.pow(base, exp)	Calcule la puissance d'un nombre
Math.sqrt(x) / Math.abs(x)	Calcule la racine carrée / valeur absolue

Objet Date (new Date())	Usage / Description
getDate() / getMonth()	Jour du mois (1-31) / Mois (0-11)
getFullYear()	Année complète sur 4 chiffres
getHours() / getMinutes()	Heures / Minutes actuelles
setFullYear(y) / setMonth(m)	Modifie l'année ou le mois
toLocaleString()	Convertit la date en chaîne selon la locale

3. Objets du Navigator (BOM & DOM)

1. **window** : Objet au sommet de la hiérarchie (fenêtre du navigateur). Méthodes : `open()`, `close()`, `alert()`, `confirm()`, `prompt()`, `setTimeout()`.
2. **document** : Représente la page HTML chargée dans la fenêtre. Contient les collections : `forms`, `images`, `links`, `cookie`, `title`.
3. **navigator** : Contient les informations relatives au navigateur utilisé (`appName`, `appVersion`, `platform`, `plugins`).

1. Nommage et Adressage des Objets HTML

Définition

Un Événement est une action de l'utilisateur (clic, survol, touche du clavier) qui déclenche l'exécution d'une fonction JavaScript.

Tout élément HTML d'un document est un objet manipulable.

Méthode d'identification	Syntaxe d'accès JS	Usage
Par ID (attribut id)	document.getElementById('monId')	Élément unique identifié par son ID
Par Nom (attribut name)	window.document.formulaire.zone.value	Accès par le chemin dans l'arborescence
Par Classe (attribut class)	document.getElementsByClassName('maClasse')	Renvoie un tableau d'éléments ayant cette classe
Par Balise (TagName)	document.getElementsByTagName('p')	Renvoie un tableau de toutes les balises spécifiées

2. Tableau des Événements Principaux

Événement	Se déclenche quand...	Exemple d'usage HTML
onLoad	La page web est entièrement chargée	<body onLoad="init()">
onUnLoad	L'utilisateur quitte la page	<body onUnLoad="nettoyer()">
onClick	Clic sur un élément (bouton, lien)	<button onClick="calculer()">
onDblClick	Double-clic rapide sur un élément	
onMouseOver	Le pointeur de la souris survole un élément	<img onMouseOver="src='photo2.jpg'">
onMouseOut	La souris sort de la zone de l'élément	<img onMouseOut="src='photo1.jpg'">
onFocus	Un champ de formulaire devient actif	<input onFocus="surligner()">
onBlur	Un champ de formulaire perd le focus	<input onBlur="valider()">
onChange	La valeur d'un champ est modifiée	<select onChange="mettreAJour()">
onSubmit	Clic sur le bouton d'envoi du formulaire	<form onSubmit="return validerForm()">

3. Modification des Propriétés et du Style

Ce qu'on modifie	Syntaxe JavaScript	Exemple concret
Contenu HTML	objet.innerHTML = valeur	document.getElementById('p1').innerHTML = 'Nouveau texte';
Attribut HTML	objet.attribut = valeur	document.getElementById('img1').src = 'image2.png';
Style CSS	objet.style.propriete = valeur	document.getElementById('box').style.color = 'blue';
Valeur de champ	objet.value = valeur	document.getElementById('input1').value = 'Résultat';

Attention

CSS en JavaScript : les propriétés CSS contenant un tiret (ex: `background-color`, `font-size`) s'écrivent obligatoirement en CamelCase (ex: `element.style.backgroundColor`, `element.style.fontSize`).

4. Exemple complet : Calcul du carré d'un nombre

```
  // Fonction JS de calcul du carré
function carre() {
  var nb = document.getElementById("nbre").value;  // Récupère la saisie
  nb = parseInt(nb, 10);  // Convertit en nombre entier
  var elt = document.getElementById("resultat");
  elt.innerHTML = nb + ' * ' + nb + ' = ' + (nb * nb);  // Modifie l'affichage
}
```