CoursInfo

JavaScript — Programmation Web Dynamique

Apprenez à programmer en JavaScript et à créer des pages web interactives : DOM, événements, et requêtes asynchrones.

Niveau : Intermédiaire • Durée : ~18h • 9 chapitres • Certificat inclus

1. Introduction à JavaScript

JavaScript est le langage de programmation qui rend les pages web interactives : validation de formulaire en direct, menus déroulants, animations, requêtes vers un serveur sans recharger la page... C'est aujourd'hui le seul langage exécuté nativement dans tous les navigateurs.

Où placer le code JavaScript ?

```
<!-- 1. Fichier externe (recommandé) -->


<script

src
=
"script.js"

defer
></script>


<!-- 2. Bloc interne -->


<script>

console.log(
'Bonjour'
);

</script>
```

■ L'attribut defer : placez le <script> avec defer dans le <head> ; le fichier sera téléchargé en parallèle mais exécuté seulement après le chargement complet du HTML.

2. Variables et types de données

```
let
nom =
'Fatou'
;
// modifiable, portée de bloc


const
PI =
3.14
;
// constante, ne peut pas être réassignée


var
ancien =
'éviter'
;
// ancienne syntaxe, à éviter (portée de fonction)


// Types primitifs


let
texte =
"Bonjour"
;
// string


let
nombre =
42
;
// number


let
actif =
true
;
// boolean


let
rien =
null
;
// absence de valeur volontaire


let
indefini;
// undefined : pas encore assigné
```

Mot-clé	Réassignable	Portée

let	Oui	Bloc { }
const	Non	Bloc { }
var	Oui	Fonction (déconseillé)

■■ Bonne pratique : utilisez const par défaut, et let seulement si la valeur doit changer. Évitez var dans du code moderne.

3. Opérateurs et structures conditionnelles

```
const
age =
20
;


if
(age >=
18
) {
console.log(
"Majeur"
);
}
else if
(age >=
13
) {
console.log(
"Adolescent"
);
}
else
{
console.log(
"Enfant"
);
}


// Opérateur ternaire (raccourci)


const
statut = age >=
18
?
"Majeur"
:
"Mineur"
;


// Comparaison stricte === (recommandée) vs == (évitez)


5
===
"5"

// false (types différents)


5
==
"5"
```

```
// true (conversion implicite, source de bugs)
```

■ Toujours === : l'égalité stricte évite les conversions de type surprenantes qui causent des bugs difficiles à repérer.

4. Boucles

```
// Boucle for classique


for
(
let
i =
0
; i <
5
; i++) {
console.log(i);
}


// Boucle while


let
i =
0
;

while
(i <
3
) {
console.log(i);
i++;
}


// for...of : parcourt les valeurs d'un tableau


const
fruits = [
"pomme"
,
"banane"
,
"mangue"
];

for
(
const
fruit
of
fruits) {
console.log(fruit);
}
```

■■ Boucle infinie : vérifiez toujours que la condition de sortie (i < 5) finit par devenir fausse, sinon le navigateur se bloque.

5. Fonctions

```
// Déclaration classique


function
addition(a, b) {

return
a + b;
}


// Fonction fléchée (arrow function) — syntaxe moderne


const
addition2 = (a, b) => a + b;


// Paramètre par défaut


function
saluer(nom =
"invité"
) {
console.log(
`Bonjour ${nom}`
);
}

saluer();
// "Bonjour invité"

saluer(
"Fatou"
);
// "Bonjour Fatou"
```

■ Template literals : les guillemets obliques `...` permettent d'insérer des variables avec ${'{'}variable{'}'} sans concaténer des chaînes avec +.

6. Tableaux et objets

```
const
notes = [
12
,
15
,
9
,
18
];

notes.push(
20
);
// ajoute à la fin

notes.length;
// nombre d'éléments


const
moyennes = notes.map(n => n *
2
);
// transforme chaque élément


const
bonnes = notes.filter(n => n >=
10
);
// filtre selon une condition


const
total = notes.reduce((acc, n) => acc + n,
0
);
// accumule une valeur

const
etudiant = {
nom:
"Fatou"
,
filiere:
"Réseaux Informatiques"
,
niveau:
"Licence 3"

};

console.log(etudiant.nom);
// notation point

console.log(etudiant[
"filiere"
```

```
]);
// notation crochets

etudiant.diplome =
false
;
// ajout d'une propriété
```

■■ map / filter / reduce : ces méthodes de tableau sont essentielles en JavaScript moderne — elles remplacent la plupart des boucles for manuelles pour transformer des données.

7. Manipulation du DOM et événements

Le DOM (Document Object Model) est la représentation en mémoire de la page HTML que JavaScript peut lire et modifier en temps réel.

```
// Sélectionner un élément


const
bouton = document.querySelector(
'#monBouton'
);

const
titre = document.querySelector(
'h1'
);


// Modifier le contenu / le style

titre.textContent =
"Nouveau titre"
;
titre.style.color =
"#264de4"
;
titre.classList.add(
"actif"
);


// Écouter un événement (clic)

bouton.addEventListener(
'click'
, () => {

alert
(
'Bouton cliqué !'
);
});


// Empêcher un formulaire de recharger la page

document.querySelector(
'form'
).addEventListener(
'submit'
, (e) => {
e.preventDefault();
});
```

8. Requêtes asynchrones : fetch & JSON

fetch() permet à une page web de communiquer avec un serveur (ex. un script PHP) sans recharger la page — c'est la base des sites web modernes interactifs.

```
async function
chargerCours() {

try
{

const
reponse =
await
fetch(
'/api/cours.php'
);

const
donnees =
await
reponse.json();
console.log(donnees);
}
catch
(erreur) {
console.error(
"Erreur de chargement :"
, erreur);
}
}

chargerCours();
```

JSON (JavaScript Object Notation) est le format d'échange de données standard entre un navigateur et un serveur — un texte structuré ressemblant à un objet JavaScript.

```
{

"nom"
:
"HTML5"
,

"niveau"
:
"Débutant"
,

"chapitres"
:
8

}
```

■■ async/await : une fonction contenant await doit être déclarée async, et tout appel réseau doit être entouré d'un try/catch pour gérer les erreurs.

9. Projet final : mini application interactive

■ Projet final : Créez une liste de tâches (« to-do list ») en JavaScript pur : Un champ de texte + un bouton « Ajouter » Chaque tâche ajoutée s'affiche dans une <ul> avec un bouton « Supprimer » Utilisez addEventListener, createElement et classList.toggle pour cocher/décocher une tâche

■ Étape suivante : après avoir maîtrisé JavaScript côté navigateur, le cours PHP vous apprendra à traiter les données côté serveur et à les stocker dans une base de données.

Exercices & Quiz de vérification

Testez vos connaissances avant de passer au cours suivant. Chaque question a une correction détaillée à dérouler.

1. Quelle est la différence principale entre let et const ?

- A. Aucune différence
- B. const empêche la réassignation de la variable
- C. let est plus rapide

Correction : Réponse : B. Une variable déclarée avec const ne peut pas être réassignée après son initialisation.

2. Que renvoie 5 === "5" en JavaScript ?

- A. true
- B. false
- C. Une erreur

Correction : Réponse : B. L'opérateur === compare aussi le type ; un nombre et une chaîne ne sont jamais strictement égaux.

3. Quelle méthode permet d'écouter un clic sur un bouton ?

- A. bouton.onEvent('click')
- B. bouton.addEventListener('click', fonction)
- C. bouton.click.listen()

Correction : Réponse : B. addEventListener('click', callback) est la méthode standard pour réagir à un événement.

4. Pourquoi utiliser try/catch avec fetch ?

- A. Pour accélérer la requête
- B. Pour gérer proprement les erreurs réseau ou serveur
- C. Ce n'est jamais utile

Correction : Réponse : B. Une requête réseau peut échouer (serveur indisponible, pas de connexion) ; try/catch évite que l'erreur ne casse le script.

Ressources supplémentaires

- MDN Web Docs — Référence JavaScript
- MDN — Apprendre les bases de JavaScript
- MDN — Le DOM (Document Object Model)
- JavaScript.info — Tutoriel moderne détaillé (EN)