

JavaScript : initiation et programmation orientée objet



Chapitre 1: Le langage du web

Le langage du web

JavaScript : initiation et programmation orientée objet

Page 3



JavaScript : initiation et programmation orientée objet

Page 4

Une naissance contrariée

1995

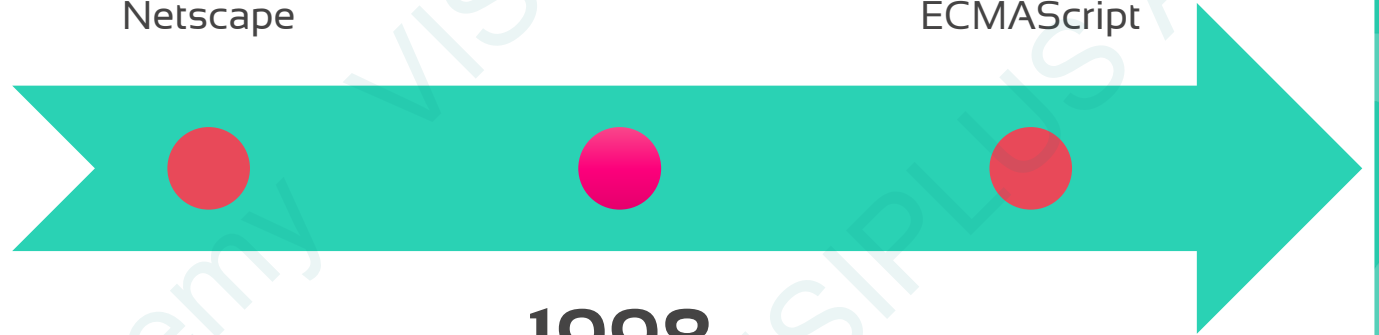
Sun
Netscape

1999

ECMA
ECMAScript

1998

Microsoft
JScript
Internet
Explorer



JavaScript : initiation et programmation orientée objet

Page 5

La fin des dissensions

2004

Apple, Mozilla,
Opera
WHATWG

2017

Microsoft
rejoint le
WHATWG

2012

Abandon de
JScript

JavaScript : initiation et programmation orientée objet

Page 6

Souplesse et efficacité

Langage
interprété.
Pas d'étape de
compilation

100% objet.
Moderne dès sa
conception

100%
programmation
web.
Pas un ancien
langage
arrangé...

JavaScript : initiation et programmation orientée objet

Page 7

Sommaire

Environnement
de travail
VS Code

Les bases de
JavaScript
Les 6 outils du
développeur

Les bases de la
POO
Classes,
propriétés,
méthodes,
héritage...

JavaScript : initiation et programmation orientée objet

Page 8

Ce qu'il faut **retenir**

- ▼ Le langage du web
- ▼ 100% interprété
- ▼ 100% objet
- ▼ 100% programmation web



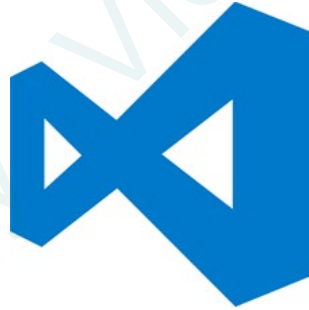


Chapitre 2 : Environnement de travail

JavaScript : initiation et programmation orientée objet

Page 10

VS Code



Open-source

Gratuit

Multi-
plateformes

Nombreuses
extensions...

Très répandu
en entreprise

JavaScript : initiation et programmation orientée objet

Page 11

Ce qu'il faut **retenir**



- ▼ **VS Code**
 - ▶ Travailler efficacement
 - ▶ Déboguer rapidement
- ▼ **JavaScript** fonctionne en association avec les vues HTML.
- ▼ **Séparer JavaScript et HTML**
 - ▶ Lisibilité
 - ▶ Maintenance

Chapitre 3 :

Les variables

Les variables

Une variable
permet de
stocker une
valeur

Il est ensuite
possible de lire la
valeur stockée

Cette valeur peut
varier au fil de
l'exécution

JavaScript : initiation et programmation orientée objet

Page 14

Ce qu'il faut **retenir**



- ▼ **Une variable est un identifiant associé à une valeur.**
On la nomme en camelCase.
- ▼ **Déclaration systématique avec let.**
- ▼ **Différents types de variables**
number, string, boolean
- ▼ **Des types spéciaux**
undefined

Chapitre 4 : Les opérateurs

Les opérateurs

- ▼ Un **opérateur** réalise une opération (un traitement simple) sur une variable ou entre deux variables (les opérandes).

JavaScript : initiation et programmation orientée objet

Page 17

Ce qu'il faut **retenir**



- ▼ Différents types d'opérateurs
 - ▶ Opérateur d'affectation
 - ▶ Opérateurs arithmétiques
 - ▶ Opérateur de concaténation
 - ▶ Opérateurs de comparaison
 - ▶ Opérateurs logiques

- ▼ Conversions entre types
Implicites ou explicites



Chapitre 5 :

Les structures conditionnelles



Une structure conditionnelle...

- ▼ **Modifie le fil d'exécution** en fonction de l'évaluation d'une expression à vrai ou faux.

JavaScript : initiation et programmation orientée objet

Page 20

Ce qu'il faut **retenir**



- ▼ Différentes structures
 - ▶ if/else
 - ▶ switch/case
 - ▶ Opérateur ternaire
- ▼ Evaluation booléenne
Zéro est faux !
- ▼ Short-circuit operators
Avec ET et OU

Chapitre 6 :

Les structures itératives

Les structures itératives

Elles permettent de
répéter un traitement
tant qu'une condition
est vraie

Souvent appelées
« boucles »

JavaScript : initiation et programmation orientée objet

Page 23

Ce qu'il faut **retenir**



- ▼ Différentes structures
 - ▶ while
 - ▶ do/while
 - ▶ For
- ▼ Ruptures possibles
 - ▶ break
 - ▶ Continue
- ▼ Attention aux boucles sans fin !

Chapitre 7 : Les tableaux

Les tableaux

Un tableau regroupe
des valeurs sous un
unique identifiant

C'est le type de
collection le plus
simple, présent dans
tous les langages

Ce qu'il faut **retenir**



- ▼ Un tableau est une collection.
Variable indicée à partir de zéro
- ▼ Un tableau possède une longueur :
Nombre d'éléments
- ▼ Un tableau se parcourt habituellement
avec une boucle for ou for/of.



Chapitre 8 : Les tableaux multi-dimensionnels

Les tableaux multi-dimensionnels

- ▼ Les éléments d'un tableau peuvent être eux-mêmes des tableaux.

JavaScript : initiation et programmation orientée objet

Page 29

Ce qu'il faut **retenir**



- ▼ Limiter les tableaux multi-dimensionnels à 2 dimensions.
- ▼ Les tableaux multi-dimensionnels se parcourent avec des boucles for ou for/of imbriquées
- ▼ Le destructuring assignment est utile pour échanger 2 variables.

Chapitre 9 : Les fonctions

Les fonctions

Une fonction est un bloc de code que l'on peut exécuter chaque fois que nécessaire

Anciennement appelée « routine », « procédure » ou « sous-programme »

JavaScript : initiation et programmation orientée objet

Page 32

Ce qu'il faut **retenir**



- ▼ Une fonction...
 - ▶ Améliore la lisibilité
 - ▶ Evite les redondances
 - ▶ Simplifie la maintenance
 - ▶ Peut recevoir des paramètres
 - ▶ Peut retourner un résultat

Chapitre 10 : Les paramètres

Les paramètres

- ▼ Nous n'avons fait que les survoler, les paramètres méritent que nous nous attardions sur leurs différents usages et sur quelques détails subtils...

JavaScript : initiation et programmation orientée objet

Page 35

Ce qu'il faut **retenir**



- ▼ Les fonctions gagnent en généricité avec :
 - ▶ Les paramètres par défaut
 - ▶ Le rest-parameter
 - ▶ Le spread operator
- ▼ Une fonction variadique admet un nombre quelconque de paramètres
- ▼ Les paramètres sont passés par valeur (primitifs) ou par référence (objets)

Chapitre 11 :

La récursivité

La **récur**sivité

Une fonction récursive
est une fonction qui
s'appelle elle-même

Quel intérêt ?

JavaScript : initiation et programmation orientée objet

Page 38

Ce qu'il faut **retenir**



- ▼ Pour programmer un algorithme récursif... Rien de plus simple qu'une fonction récursive...
- ▼ Mais attention au temps d'exécution et gare au Stack Overflow !
- ▼ Lorsque possible, un algorithme itératif est toujours plus performant que son homologue récursif

Chapitre 12 : La POO

JavaScript : initiation et programmation orientée objet

Page 40

En POO...

Les entités de la vie
réelle sont
représentées par des
objets

Un objet a des
variables (propriétés)
et des fonctions
(méthodes)

JavaScript : initiation et programmation orientée objet

Page 41

Ce qu'il faut **retenir**



- ▼ Une classe contient des membres : propriétés et méthodes.
- ▼ Le mot-clé **new** crée une instance d'une classe. C'est une allocation mémoire.
- ▼ Dans un constructeur, **this** représente l'instance en cours de création.
- ▼ Dans une méthode, **this** représente l'instance sur laquelle a été invoquée la méthode.
- ▼ Une propriété statique concerne la classe elle-même dans son ensemble.
- ▼ Pas de **this** dans une méthode statique !



Chapitre 13 : Les modificateurs d'accès

JavaScript : initiation et programmation orientée objet

Page 43

Les modificateurs d'accès

Ils définissent
les règles
d'accès aux
membres des
classes

Ils devraient
permettre de
travailler à
plusieurs
développeurs
avec une
meilleure
sécurité...

Mais en
JavaScript, ils
sont pour
l'instant limités

JavaScript : initiation et programmation orientée objet

Page 44

Ce qu'il faut **retenir**



- ▼ **JavaScript offre 2 types d'accès :**
 - ▶ **public** (par défaut)
Accès de partout.
 - ▶ **private**
Accès uniquement depuis les méthodes de la classe elle-même.

Chapitre 14 :

Les classes built-in

JavaScript : initiation et programmation orientée objet

Page 46

Les classes built-in

Ce sont les classes
déjà présentes dans
JavaScript

Les 3 plus
importantes :

- ▼ Math
- ▼ String
- ▼ Array

JavaScript : initiation et programmation orientée objet

Page 47

Ce qu'il faut **retenir**



- ▼ **La classe Math est 100% statique.**
Constantes et méthodes pour calculs mathématiques
- ▼ **La classe String enveloppe le type primitif string avec auto-boxing.**
Ses instances sont immutables et ses méthodes « pures »
- ▼ **La classe Array est la classe des tableaux.**



Chapitre 15 : Application Boulangerie Analyse

Application Boulangerie **Analyse**

Application très simplifiée de gestion des fabrications d'une boulangerie.

Documents fournis :

- ▼ Programme principal **main.js**
- ▼ Sortie attendue **log.txt**

Analysez le programme principal.

Lister les classes, propriétés et méthodes nécessaires


JavaScript : initiation et programmation orientée objet

Page 50

Ce qu'il faut **retenir**



- ▼ Le **diagramme des classes** est une étape majeure de la conception
- ▼ La normalisation UML est simple
 - ▶ Classes
 - ▶ Dépendances
 - ▶ Agrégations
 - ▶ Compositions
- ▼ Après, ce n'est que du codage...



Chapitre 16 : Application Boulangerie Codage

Application Boulangerie

Codage

JavaScript : initiation et programmation orientée objet

Page 52

Coder le squelette de
chaque classe d'après
le diagramme des
classes

Coder les méthodes
d'après la sortie
attendue

JavaScript : initiation et programmation orientée objet

Page 53

Ce qu'il faut **retenir**

- ▼ Coder d'abord le squelette des classes
- ▼ Coder ensuite les méthodes



Chapitre 17 : L'héritage

JavaScript : initiation et programmation orientée objet

Page 55

L'héritage

Une classe peut
hériter d'une autre

Elle bénéficie de ses
membres publics

Elle peut ajouter de
nouveaux membres

Elle peut **redéfinir**
les méthodes
parentes

JavaScript : initiation et programmation orientée objet

Page 56

Ce qu'il faut **retenir**



- ▼ En JavaScript, l'héritage est unique : une classe n'a qu'un seul parent
- ▼ L'héritage est une relation de spécialisation / généralisation
- ▼ Une classe enfant hérite des membres publics de sa classe parente
- ▼ Le mot-clé super permet d'accéder à la classe parente.



Chapitre 18 :

Application Boulangerie

Les pâtisseries

Application Boulangerie

Les pâtisseries

Les pâtisseries sont des
boulangers
spécialisés. Ils peuvent
fabriquer des produits
de boulangerie ainsi
que des pâtisseries

Documents fournis :

- ▼ programme principal
`main.js`
- ▼ Sortie attendue
`log.txt`

Ce qu'il faut **retenir**

- ▼ Le diagramme **UML** des classes doit intégrer les relations d'héritage





Chapitre 19 : Application Boulangerie Solution

Application Boulangerie

Solution

JavaScript :
initiation et
programmation
orientée objet

Page 61

Conserver le code
existant

Le faire évoluer via
l'héritage

Ce qu'il faut **retenir**

- ▼ Utiliser l'héritage dès qu'une relation de **généralisation / spécialisation** est en jeu.





Chapitre 20 :

Contexte et arrow functions

JavaScript : initiation et programmation orientée objet

Page 64

Le contexte

L'étude du **contexte**,
c'est la détermination
de la valeur de **this**.

Les **arrow functions** se
révèlent alors bien
utiles...

JavaScript : initiation et programmation orientée objet

Page 65

Ce qu'il faut **retenir**



- ▼ Ne pas confondre contexte et scope
- ▼ Utiliser les arrow functions chaque fois que possible pour simplifier l'écriture
- ▼ Une arrow function prend le contexte du bloc enveloppant.

Conclusion

JavaScript : initiation et programmation orientée objet

Page 67

Conclusion

Vous maîtrisez les
bases de JS et la POO.

Libérez toute sa
puissance avec :

- ▼ La **programmation événementielle** pour manipuler le DOM
- ▼ La **programmation asynchrone** pour des aller-retours client-serveur en tâche de fond