Aide-Mémoire JS



Lucas Simões Pólvora

MID2

2023-2024

Table des matières

[1. Introduction 3](#_Toc155687410)

[2. Fondamentaux 3](#_Toc155687411)

[2.1 Mettre un script JS dans un fichier HTML 3](#_Toc155687412)

[2.2 Commentaires 3](#_Toc155687413)

[2.3 Variables et Constantes 3](#_Toc155687414)

[2.4 Interactions : Alert, prompt, confirm 3](#_Toc155687415)

[2.5 Conversion de types 3](#_Toc155687416)

[2.5.1 Conversion en string 3](#_Toc155687417)

[2.5.2 Conversion numérique 4](#_Toc155687418)

[2.5.3 Conversion Booléenne 4](#_Toc155687419)

[2.6 Opérateurs de bases de mathématique 4](#_Toc155687420)

[2.7 Les conditions (if, else if, else, opérateur ternaire) 4](#_Toc155687421)

[2.8 Opérateurs logiques 5](#_Toc155687422)

[2.9 Opérateur de coalescence des nuls (??) 5](#_Toc155687423)

[2.10 Boucles (while, for, do…while) 5](#_Toc155687424)

[2.11 Fonctions 5](#_Toc155687425)

[2.12 Expressions de fonctions 6](#_Toc155687426)

[2.12.1 Fonction est une valeur 6](#_Toc155687427)

[2.12.2 Fonctions Callback 6](#_Toc155687428)

[2.13 Différences entre les fonctions et fonctions expressions 6](#_Toc155687429)

[2.14 Les fonctions fléchées 7](#_Toc155687430)

[3. Les Objets 7](#_Toc155687431)

[3.1 Définition 7](#_Toc155687432)

# Introduction

# Fondamentaux

## Mettre un script JS dans un fichier HTML

Nous avons 2 choix :

* Utiliser la balise script (Exemple : <Script> le code </script>)
* Utiliser un lien (Exemple : <Script src= « /chemin/vers/script.js »></script>

## Commentaires

Les commentaires sont écrits comme en C#. Ce qui veut dire qu’il faut introduire par // et ensuite écrire le commentaire.

## Variables et Constantes

Le JavaScript est un langage typé dynamiquement, ce qui veut dire qu’il existe des types de données, mais que les variables ne sont liées à aucun d’entre eux, donc on utilise « let » pour déclarer une variable. Le nom d’une variable doit être précis et doit montrer à quoi elle sert. Une variable peut être utilisée comme en C#.

Une constante est exactement comme en C#, elle ne peut pas changer de valeur, sa valeur est déclarée avec la constante. On utilise une écriture en majuscules seulement pour montrer que c’est une constante.

Exemples :

Variable : let jeSuisUneVariable ;

Constante : const JE\_SUIS\_UNE\_CONSTANTE ;

## Interactions : Alert, prompt, confirm

Alert(Message) : Affiche une fenêtre modale avec un message et attend que l’utilisateur appuie sur « OK »

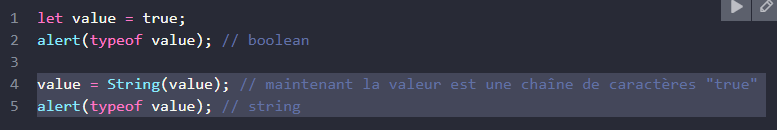
Prompt (Titre, valeur initiale du champ d’entrée(facultatif)) : Affiche une fenêtre avec un titre, un champ de saisie qui peut ensuite être dans une variable (let variable = prompt(titre)).

Confirm(question) : Affiche une fenêtre modale avec une question et deux boutons : « OK » et « annuler ». Le résultat est « true » si on appuie sur « OK » et « false » si on clique sur annuler.

## Conversion de types

### Conversion en string

On peut convertir un bool en string. Cependant elle sera égale soit à « true », soit à « false »

Exemple : 

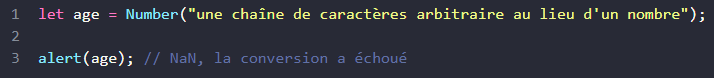
### Conversion numérique

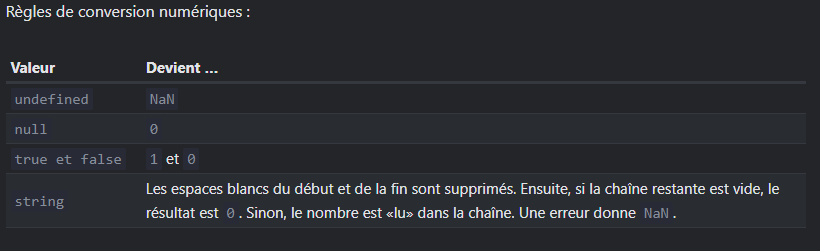
La conversion numérique dans les fonctions et expressions mathématiques s’effectue automatiquement.

Lorsque la division est appliquée à des non-numéros :



Cependant, si la chaîne de caractères est invalide, le résultat de la conversion sera NaN (Not a Number)

Exemple :



### Conversion Booléenne

Pour faire simple, si la chaîne de caractères est vide («») la valeur est « false », s’il y a qqch comme un espace ou n’importe quel caractère c’est « true »

## Opérateurs de bases de mathématique

Les opérateurs mathématiques de base sont les mêmes qu’en C#.

+ => Additionner

- => Soustraire

\* => Multiplier

/ => Diviser

% => Modulo

Il y a juste l’exponentiation qui se note « \*\* » c’est la même chose que les puissances en mathématiques.

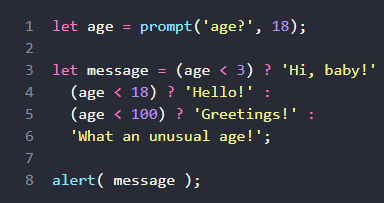
## Les conditions (if, else if, else, opérateur ternaire)

La règle est la même qu’en C#.

If(condition){Code} else if(condition){code} else{code}

Let result = condition ? valeur 1(si vrai) : valeur 2(si faux)

Nous pouvons enchaîner les opérateurs ternaire si besoin :  
Exemple :



## Opérateurs logiques

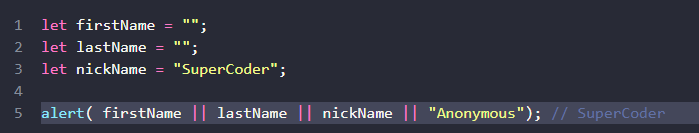
Même chose qu’en C#

|| = ou

&& = et

Elle peut aussi être utilisé pour obtenir la première valeur vraie

Exemple :



## Opérateur de coalescence des nuls (??)

Prend la première valeur « true »

Exemple : a ?? b (si a est défini alors a, si a n’est pas défini, alors b)

## Boucles (while, for, do…while)

La boucle while et do…while est comme en C#, il s’écrit de la manière suivante :

While (condition) {code (appelé Loop body)}

Do { code } while (condition)

For ((déclaration) ; (condition) ; (incrémentation)) { code }

## Fonctions

Pour déclarer et coder dans la fonction on écrit cela :

Function nomDeLaFonction((paramètres)) { code }

Pour utiliser la fonction :

nomDeLaFonction(paramètres)

Si on veut spécifier la valeur par défaut dans les paramètres on fait :

Function nomDeLaFonction(nomDuParamètre = (valeurDuParamètre)

Pour retourner une valeur on fait :

Function nomDeLaFonction((paramètres)) {

Code

Return (valeur que l’on veut retourner)

}

Si l’on met « Return ; » alors on va retourner aucune valeur.

## Expressions de fonctions

### Fonction est une valeur

Voici comment on écrit :

Let nomDeLaVariable = function () { code }

Ou

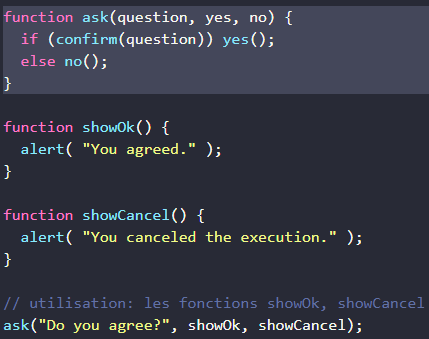
Function nomDeLaFonction((paramètres)) { code } ;

Let nomDeLaVariable = nomDeLaFonction ;

La variable prend la valeur de la fonction cela veut dire que nous pouvons soit appeler nomDeLaVariable() soit nomDeLaFonction().

### Fonctions Callback

Une fonction callback est une fonction qui va appeler d’autres fonctions par exemple :



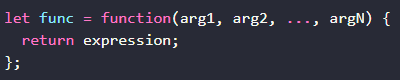
## Différences entre les fonctions et fonctions expressions

* La syntaxe
* Une fonction normale peut être appelée plus tôt que sa définition
* Quand une fonction normale est dans un bloc de code elle n’est que visible dans ce bloc. Exemple :

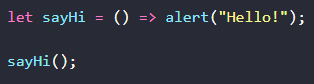


## Les fonctions fléchées

C’est une syntaxe plus concise et plus simple pour créer des fonctions que des fonctions expression, voici comment en créer une :



On n’a pas besoin de mettre des arguments mais il faut mettre les parenthèses :



Il faut cependant mettre des accolades si c’est une fonction multiligne.

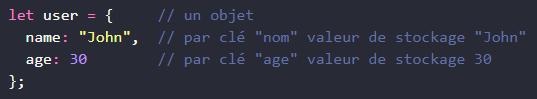
# Les Objets

## Définition

Un objet est un type de donnée qui pourra stocker des données variées et d’entités plus complexes

## Comment l’utiliser

Voici un exemple simple :



On donne un nom de variable à cet objet, ici on crée une personne elle aura donc une variable pour le nom (on va nommer la variable name) et age.

Pour reprendre ces informations on va faire user.name ou user.age.