

TP Javascript (1) - 2025

Remarque : Ce TP a une visée pédagogique uniquement, donc le code réalisé ne sera ni optimisé ni sécurisé, l'objectif étant simplement de vous montrer les différents types de traitements permis par Javascript.

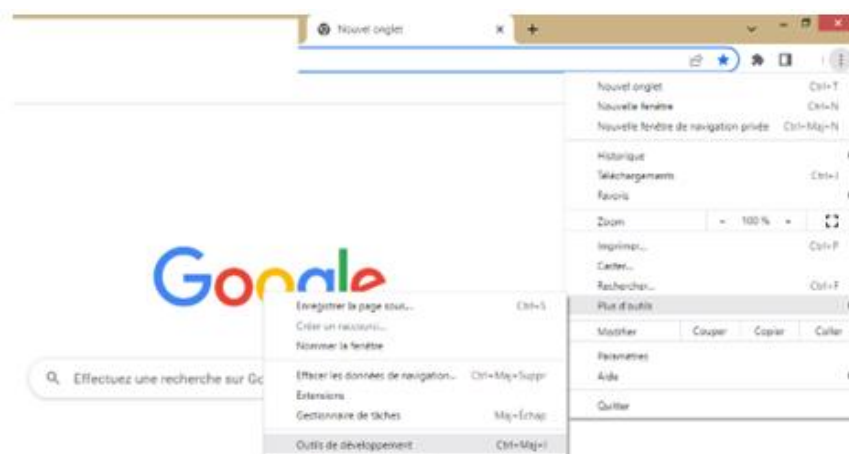
Ouvrir “**console**” dans votre navigateur :

➤ **Google Chrome :**

Soit : Ouvrir Google Chrome, presser Ctrl+Shift+j qui va ouvrir le JavaScript console là où vous pouvez taper vos codes et les tester.

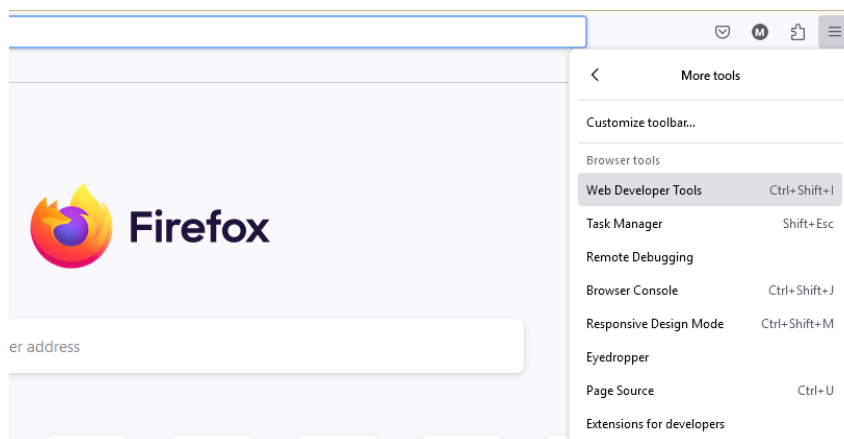
Soit :

Aller sur le menu (3 points verticaux ":" en haut droite de la page), ensuite sélectionner « Plus d'outils (en anglais : More Tools) », et ensuite choisir « Outils de développement (en anglais : Web Developer Tools) ». Enfin choisir Console tab.



➤ **Firefox:**

C'est quasi-pareil, sauf qu'au lieu de ":", vous avez "≡".




Pour exécuter vos codes Javascript :

Taper vos codes après web console prompt ">" (ou "»" dans Firefox), et presser Enter.

Pour les exercices suivants, n'hésitez pas à consulter <https://www.w3schools.com/js/> pour des commandes, exemples, ... du Javascript.

Quelques Remarques :

- Au lieu de copier-coller du texte de ce document, taper vous-même, sinon il y aura éventuellement des problèmes avec ~, ',
- Vous pouvez effacer les codes précédents par clear() ou cliquer sur  du console
- Si vous voulez utiliser apostrophe ', vous devez le précéder par un anti-slash \
- En Javascript, concaténation se fait par +
- **NaN** veut dire : Not a Number
- Au lieu de "alert()", vous pouvez utiliser "console.log()" pour avoir les sorties dans votre console du Chrome.
- N'hésitez pas à consulter <https://www.w3schools.com/js/>
- Lorsque vous monter une page web qui contient des éléments du Javascript, il est conseillé de mettre les codes Javascript dans le body du votre code HTML.

Exercice 1

- Deviner la sorties des exemples suivants.
- Ensuite copier-coller chaque exemple dans la console du Google Chrome pour savoir si vous avez deviné correctement.
 - (i) Evaluer les suivants :
5 + "39"
5 - "7"
12 % 6
6 % 12
"Java" + "Script"
" " + " "
" " + 0
true + true
true + false
false + true
false - true
7 - 8
"Harry" - "Billy"
 - (ii) Evaluer les comparaisons suivantes :
7 >= 5
0 === 1

```
3 <= 1
1 != 1
"D" > "E"
"E" < "F"
"a" > "A"
"b" < "A"
true === false
true != true
```

Exercice 2 : Calculatrice

Monter une petite calculatrice en utilisant `prompt()` et `alert()`. Pour cela, taper des codes qui font les suivants :

1. Il demande à l'utilisateur le premier nombre
2. En gardant le premier nombre, il demande le deuxième nombre, et il le sauvegarde
3. Il donne la somme des deux nombres et il l'affiche en utilisant `alert()`

Modifier votre programme de façon à pouvoir soustraire, multiplier et aussi diviser !

Exercice 3 : école de conduire I

Ecrire des codes pour vérifier l'âge d'un utilisateur s'il a plus de 18 ans, et donner des réponses suivantes :

1. S'il a moins de 18 ans : désolé vous êtes trop jeune pour conduire une voiture !
2. S'il a juste 18 ans : Bienvenue au club !
3. S'il a plus de 18 ans : Bon Voyage !

Exercice 4 :

Ecrivez un petit programme Javascript :

Calculer la somme de deux nombres, vérifiez si la réponse est correcte, et si c'est le cas, le code affiche " Bravo ! T'es un champion ! ".

Faire de sorte que la sortie soit dans une fenêtre pop-up.

Ensuite changer votre code de façon que la sortie soit directement dans la console de votre navigateur.

- Au lieu de "`alert()`", vous pouvez utiliser "`console.log()`" pour avoir les sorties dans votre console du Google Chrome.

Exercice 5 : école de conduire II

Améliorer vos codes de l'exercice « d'école de conduire I » en utilisant une fonction `checkDriverAge()`. Chaque fois que vous appelez cette fonction (sans ou avec argument), elle va demander pour l'âge

- Utiliser l'approche « *Function Declaration* » pour créer cette fonction
- Utiliser l'approche « *Function Expression* » pour créer cette fonction

En tout cas, votre programme va donner la réaction (notice) correspondant à l'âge de l'utilisateur.

► Consulter la documentation pour voir la différence de deux approches :

https://www.w3schools.com/js/js_function_definition.asp

Exercice 6 : Manipuler les tableaux (array)

Créer un tableau (array) de fruits/légumes qui contient les éléments suivants :

Bananes, Pommes, Oranges, Fraises

Ensuite, faites les opérations suivantes :

1. Supprimer Bananes su tableau
2. Trier le tableau
3. Ajouter « Kiwi » au tableau
4. Trier le tableau dans le sens inverse (e.g., ['a', 'c', 'b'] devient ['b', 'c', 'a'])

N'hésitez pas à consulter <https://www.w3schools.com/js/>

Exercice 7 : Manipuler les tableaux (array)

Etant donné le tableau suivant :

```
var array2 = ["BMW", ["AUDI", ["Ferrari"], "Benz"]];
```

Comment accéder au " Ferrari " ?

Exercice 8 : Créer un nano-facebook I

Créer un objet et un array qu'on va utiliser pour notre nano-facebook :

1. Créer un objet qui a les propriétés "username" et "password". Donner-les des valeurs (du type String).
2. Créer un array, nommé "database", qui contient l'objet que vous venez de créer.
3. Créer un array nommé "newsfeed" qui contient 3 objets avec des propriétés "username" et "timeline".

Plus précisément, votre code doit être en mesure

- de vérifier l'éligibilité d'avoir accès à nano-facebook
- et, si la permission est donnée, d'afficher le timeline

Exercice 9 : Créer un nano-facebook II

Comment pourriez vous étendre le facebook que vous venez de créer de façon à accueillir plusieurs (voire millions ? enfin potentiellement) d'utilisateurs !

Tester votre code avec 3 utilisateurs différents.

Exercice 10 : Manipulation du DOM (Document Object Model)

Vérifier dans les documentions (ou <https://www.w3schools.com/js/>)

- `getElementsByTagName`

- getElementById
- createElement
- createTextNode
- KeyboardEvent *keyCode*
- addEventListener
- querySelector
- querySelectorAll
- getAttribute
- setAttribute
- parentElement
- appendChild

➤ Maintenant, vous allez utiliser **certains** (pas tous) de ces DOM selectors et méthodes pour la réalisation de l'exercice suivant.

Exercice 11 : Shopping liste

En utilisant HTML, CSS et Javascript, créer une shopping liste tel que :

1. Contient plusieurs item (produits, ...) à acheter
2. D'abord ajouter une balise « button » et vérifier quelques-uns des « events » que vous trouvez sur le site suivant :
<https://developer.mozilla.org/en-US/docs/Web/Events>
3. Quels sont les events qui peuvent vous servir dans cet exercice ?
4. Ensuite, ajouter une balise « input » qui vous permet d'ajouter davantage items/produits à la fin de votre liste d'achats. L'ajout doit être effectué soit par un button, soit en tapant sur « Enter ».
5. Faites attention qu'un input est valide s'il n'est pas vide.
6. Comment empêcher l'ajout d'un item/produit infiniment ?
7. Est-ce que et comment pouvez-vous améliorer votre code ?