

# Exercices

Pratique

# JavaScript

## Exercices FizzBuzz

### Version 1

Ecrire une fonction `fizzBuzz` qui prend 2 entiers en paramètre 'start' et 'end' et qui pour chaque entier de 'start' à 'end' inclus effectue l'opération suivante

- Si le nombre est divisible par 3, on affiche 'Fizz'.
- Si le nombre est divisible par 5, on affiche Buzz
- Si le nombre est divisible par 3 et par 5, on affiche Fizzbuzz
- Sinon, on affiche le nombre passé en paramètre.

L'affichage dans la console, doit être sous cette forme  
<valeur de n>: <valeur de la fonction `fizzBuzz` pour n>

Pour faire un test, vous pouvez appeler la fonction avec les paramètres suivants  
`fizzBuzz(1, 30);`

# JavaScript

## Version 2

Écrire un moteur de règle pour notre précédente fonction fizzBuzz.

La fonction décrite précédemment doit prendre un tableau de règles et appliquer ces règles pour des nombres allant de 1 à 30.

Voici un exemple d'appel de cette fonction

```
const rules = [  
  { predicate: (n) => n % 3 === 0, tag: "Fizz" },  
  { predicate: (n) => n % 5 === 0, tag: "Buzz" }  
];  
  
fizzBuzz(rules, 1,30);
```

# JavaScript

## Version 3

Utiliser des expressions régulières pour définir le domaine d'exécution de la fonction fizzBuzz.

Au lieu de passer en paramètre les nombres de début et de fin, utilisez une chaîne de caractères avec une notation mathématique.

Pour afficher les nombres fizzbuzz de 1 à 30.

```
fizzBuzz(rules, "[1,30]");
```

Pour afficher les nombres fizzbuzz de 2 à 29.

```
fizzBuzz(rules, "]1,30[");
```

# JavaScript

## Bubbling est capturing d'événements

Intercepter les événements bubbling et capturing dans une page web.  
Loguer les résultats dans la console.

Une interface simplifiée peut être utilisée à cet effet.



- ☒ Bubbling
- ☐ Capturing

# JavaScript

**Le palindrome (Exo4)**

# JavaScript

**Les nombres Armstrong**

**Les taux d'intérêts (Exo9)**

**Calcul impôts (Exo17)**

**VariablesChaines (Exo1)**

**Accroissement de la population (Exo28)**

# JavaScript

**Code Hexadécimal d'une chaîne de caractère**

**Calcul de CheckSum**

Pour un fichier donné calculer le checksum

**Sauvegarder et restituer le contenu JSON dans un fichier**

Base de donnée légère.



# JavaScript

## **Exercice appel d'une API**

XmlHttpRequest ou Fetch

Exercice

# HTML 5

**Exercice Geolocalisation**

**Exercice Drag & Drop**

**Exercice PushState**

Utilisation de tabulation.

**Exercice Demo Three.js**

# TypeScript

## **Additionner les valeurs d'un arbre**

Ecrire une fonction TypeScript qui additionne les valeurs d'un arbre

L'objet aura 2 propriétés gauche et droite de type number ou null ou un objet.

## **Relation Eleves Professeur**

Créer une sorte de base de données élève, classe, professeur et coder quelques fonctions.

## **Union Type**

créer une fonction qui va vous retourner la

valeur d'un nombre si le paramètre dans la fonction est un nombre unique ou la somme

d'un tableau de nombre si le paramètre passé dans la fonction est un tableau de nombre