



# Baptême Helper

Correction par Morvan DOUAUD

## Table des matières

1	Correction pour chaque étudiant.....	4
1.1	Etudiant 1 .....	4
1.1.1	Exercice 1.....	4
1.1.2	Exercice 2.....	4
1.1.3	Exercice 3.....	4
1.1.4	Exercice 4.....	4
1.1.5	Exercice 5.....	4
1.2	Etudiant 2 .....	5
1.2.1	Exercice 1.....	5
1.2.2	Exercice 2.....	5
1.2.3	Exercice 3.....	5
1.2.4	Exercice 4.....	5
1.2.5	Exercice 5.....	5
1.3	Etudiant 3 .....	6
1.3.1	Exercice 1.....	6
1.3.2	Exercice 2.....	6
1.3.3	Exercice 3.....	6
1.3.4	Exercice 4.....	6
1.3.5	Exercice 5.....	6
1.4	Etudiant 4 .....	7
1.4.1	Exercice 1.....	7
1.4.2	Exercice 2.....	7
1.4.3	Exercice 3.....	7
1.4.4	Exercice 4.....	7
1.4.5	Exercice 5.....	7
2	Commentaires sur chaque étudiant à l'équipe pédagogique .....	8
2.1	Etudiant 1 .....	8
2.2	Etudiant 2 .....	8
2.3	Etudiant 3 .....	8
2.4	Etudiant 4 .....	8
3	Commentaires sur le challenge à O'clock.....	9
3.1	Exercice 1.....	9
3.2	Exercice 2.....	9
3.3	Exercice 3.....	9
3.4	Exercice 4.....	9

3.5	Exercice 5.....	9
-----	-----------------	---

# 1 Correction pour chaque étudiant

## 1.1 Etudiant 1

### 1.1.1 Exercice 1

Il n'est pas la peine de créer une variable pour un algorithme si simple, sauf pour le debug ou si une charte de développement associée au programme veut que l'on en retourne une et non un calcul.

### 1.1.2 Exercice 2

Même chose que l'exercice 1 : ce code est très bien détaillé mais un peu verbeux. Faire un return du calcul fonctionne directement.

### 1.1.3 Exercice 3

Que vient faire le console.log (à part pour du debug) ?

De plus il vaut mieux tester l'absence du point au début de la fonction afin de traiter ce « cas d'erreur » et partir dans le vrai algorithme ensuite. Mais ce code fonctionne bien.

### 1.1.4 Exercice 4

Même chose que pour l'exercice 3 : tester la taille du paramètre au début de la fonction afin de traiter le cas.

Et le type des éléments en paramètre n'est pas testé, un length sur des nombres est donc réalisé (pour quel résultat ?) alors que le principe est de retourner une chaîne.

### 1.1.5 Exercice 5

Encore une fois, des console.log sont présents : s'ils ont été utiles afin de développer, vérifier qu'ils sont autorisés par celui recevant la fonction (au cas où il souhaite la lancer sans laisser de trace).

De plus, un ELSE aurait suffi au lieu des 2 IF et le type de chaque élément du paramètre n'est pas testé donc s'il s'agit d'un tableau, on ne le parcourt pas.

En résumé, ces développements pourront devenir plus efficaces mais cela ne viendra qu'avec l'expérience. Ils sont bien réalisés et ne demandant qu'à se concentrer sur bien passer en revue chaque cas possible afin d'être ce que l'on attend. Efforts à poursuivre mais bon départ !

## 1.2 Etudiant 2

### 1.2.1 Exercice 1

Bien.

### 1.2.2 Exercice 2

Les parenthèses sont inutiles.

### 1.2.3 Exercice 3

Le séparateur pourrait être une constante.

Attention également à ne pas nommer une variable du même nom qu'une fonction en Javascript.

Enfin faire la vérification de la taille de getExtension avant de travailler dessus avec un slice.

### 1.2.4 Exercice 4

Fallait-il prendre en compte les nombres alors que la description de la fonction dit de retourner une chaîne ?

### 1.2.5 Exercice 5

C'est bien d'avoir su créer une seconde fonction mais cela correspond-il à la règle de l'exercice ? Il est réalisable en une seule.

En résumé, ces développements sont bons mais peut-être trop poussés. Attention à respecter les consignes afin de fournir le travail attendu.

### 1.3 Etudiant 3

#### 1.3.1 Exercice 1

Attention à bien indenter le code.

#### 1.3.2 Exercice 2

Attention à bien indenter le code.

Faire un return du calcul aurait été plus rapide.

#### 1.3.3 Exercice 3

Attention à bien indenter le code.

Retirer le console.log.

#### 1.3.4 Exercice 4

Attention à bien indenter le code.

#### 1.3.5 Exercice 5

Attention à bien indenter le code.

Ne pas créer de IF sans code à l'intérieur : utiliser un NOT afin d'aller directement à l'autre portion de code ou insérer une instruction sans conséquence.

Et éviter autant que possible les multiples boucles FOR imbriquées.

En résumé, ces développements sont bons mais il faut penser à toutes les autres personnes qui reprendront la charge de gérer l'application créée : les futurs développeurs sur le projet, les gens qui résoudront les incidents en Production... Une priorité est de produire un code clair et facilement maintenable.

## 1.4 Etudiant 4

### 1.4.1 Exercice 1

Attention à ne pas livrer les développements qui ont servi à produire le résultat : console.log, commentaires...

La variable est également inutile, un return du calcul suffisait.

### 1.4.2 Exercice 2

Même chose que l'exercice 1.

### 1.4.3 Exercice 3

Ce code ne produit pas de résultat côté HTML puisque le Javascript est mal écrit (point-virgules).

### 1.4.4 Exercice 4

Pas de proposition de ta part. Quel a été le problème ? N'hésite pas à me contacter lorsque tu rencontres une difficulté.

### 1.4.5 Exercice 5

Même chose que l'exercice 4.

En résumé, je vais te contacter afin que nous fassions le point sur ton avancée. Tu pourras m'expliquer les problèmes rencontrés. N'hésite pas à le faire de toi-même ou à demander de l'aide à tes camarades si je ne suis pas disponible (la promo est grande).

## 2 Commentaires sur chaque étudiant à l'équipe pédagogique

### 2.1 Etudiant 1

Débutant mais appliqué. Doit juste se concentrer.

### 2.2 Etudiant 2

Très doué mais veut en faire trop. Doit suivre les consignes.

### 2.3 Etudiant 3

A résolu le challenge mais pas assez soigneux. Doit penser que la présentation de son développement sera très importante dans le monde du travail.

### 2.4 Etudiant 4

Trop en difficulté. Je vais le contacter.



### 3 Commentaires sur le challenge à O'clock

Ce challenge est bien pensé, un seul inconvénient viendrait si les fonctions créées étaient réellement utilisées dans une application de Production : il faudrait alors peut-être ajouter une gestion d'erreur sur chaque paramètre en entrée des fonctions de test (string attendu mais réellement fourni?).

#### 3.1 Exercice 1

Pourquoi tester le zéro ?

#### 3.2 Exercice 2

Bien.

#### 3.3 Exercice 3

Un fichier jquery.min.js est certes un fichier JS mais surtout un .min.js. Quelle extension retourner alors ?

.es1intrc est un fichier cache. Est-ce considéré comme un fichier sans nom et juste une extension ?

Enfin, une fonction retournant un booléen ou nombre doit-elle tout d'un coup retourner false lorsque le paramètre est mauvais ? Il vaudrait mieux lui préférer null ou réaliser une gestion d'erreur.

#### 3.4 Exercice 4

Même chose pour le false.

De plus, dans la correction, tester la taille du paramètre au début de la fonction.

#### 3.5 Exercice 5

Attention dans la fonction de test fournie aux étudiants, si la somme d'un tableau vide est considérée égale à zéro, comment différencier ce cas d'une réelle suite de nombres égale à zéro ?