

Durée du CC : 1h20.

La Fiche-Triche JavaScript est le seul document autorisé.

Les deux parties sont indépendantes ; on pourra les traiter dans l'ordre de son choix. Pour le rendu, deux fichiers sont attendus : `partie1.js` pour la première partie, et `partie2.js` pour la seconde.

## 1 Comptes bancaires

Le code de cette section sera à rendre dans le fichier `partie1.js`.

**Attention : dans cette partie, le mot-clef `class` est interdit !**

**Question 1.** Écrivez une fonction `Compte` dont le fonctionnement est le suivant : l'instruction

```
1 let compteToto = new Compte("Toto", 1200);
```

crée un `compteToto` contenant les deux attributs suivants :

**nom** : le nom du propriétaire du compte ("`Toto`" dans l'exemple)

**solde** : le montant d'argent déposé sur le compte (1200 dans l'exemple)

**Question 2.** En utilisant les prototypes, faites en sorte que les objets renvoyés par un appel à `new Compte()` soient dotés des trois méthodes suivantes :

**afficherSolde()** : sans argument, affiche le **solde** du compte sur la console

**ajouter(montant)** : ajoute **montant** au **solde** du compte

**virer(montant, compte)** : fait un virement de la somme **montant** sur le compte **compte**. On devra faire appel à la méthode `ajouter()`. Si jamais le **solde** du compte courant est inférieur à **montant**, on se contentera d'afficher un message d'erreur.

**Question 3.** Implémentez une fonction `totalBanqueFor()` qui prend en argument un tableau de comptes bancaires, et renvoie l'argent total contenu sur ces comptes. Vous devrez utiliser une boucle `for...of`.

**Question 4.** Implémentez une nouvelle fonction `totalBanqueReduce()` fonctionnant comme la fonction `totalBanqueFor()`, mais en utilisant `reduce` à la place de la boucle.

**Question 5.** Implémentez une fonction `LivreA` qui se comporte comme la fonction `Compte`, mais qui crée des objets ayant un **solde** maximal de 22950. Tout ce qui devrait faire passer le **solde** au-delà de cette valeur est perdu. On fera attention à avoir un code qui minimise les redondances entre `Compte` et `LivreA`, et qui utilise les prototypes pour simuler l'héritage.

## 2 King's Landing

Téléchargez le fichier `King's_Landing.html` sur Eprel.

Le code de cette section sera à rendre dans le fichier `partie2.js`. Le fichier `King's_Landing.html` n'est pas à rendre.

**Question 6.** Ajoutez une balise au fichier `King's_Landing.html` permettant d'exécuter le code contenu dans le fichier `partie2.js`.

Pour les questions suivantes, on ne touchera plus au fichier html : les balises devront être créées ou modifiées dynamiquement depuis le script.

**Question 7.** Créez une balise `<h1>` contenant le texte `King's Landing`, et placez-la au tout début du `<body>`.

**Question 8.** Faites en sorte que la couleur d'arrière-plan (qui est gouvernée par l'attribut CSS `"background-color"`) de chaque `<h2>` soit le bleu (`"blue"`).

**Question 9.** Faites en sorte que lorsque la souris passe au-dessus d'un `<h2>`, la couleur de ce `<h2>` (qui est gouvernée par l'attribut CSS `"color"`) devienne rouge (`"red"`) pendant une seconde, avant de retourner à la normale.

**Question 10.** Faites en sorte que lorsque l'on clique sur un paragraphe `<p>`, le paragraphe suivant prenne une couleur d'arrière-plan rouge (`"red"`). S'il s'agit du dernier paragraphe, alors c'est le premier paragraphe qui deviendra rouge.