TO DO LIST :

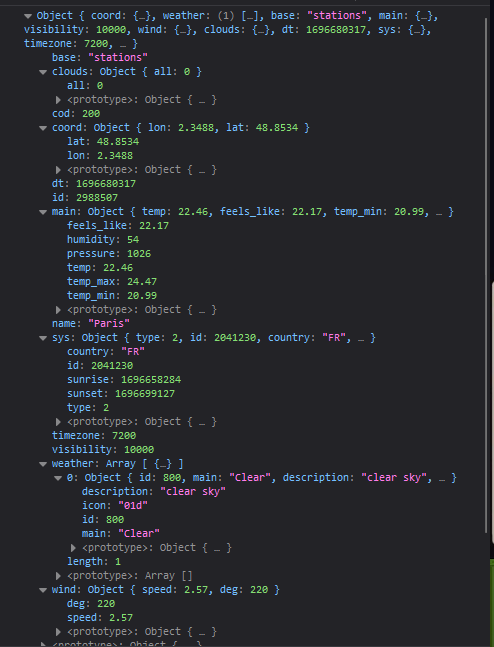
~~- Faire une API responsive (changer la ville => reload les datas)~~

~~- Faire pour la semaine entière~~

~~- Faire le design (Cartes film)~~

- Utiliser des cookies pour la valeur par défaut

- Faire un panneau comme omoi avec une suggestion ?



**Objet recu avec l’API**

# Fetch les datas

# Une image contenant texte, capture d’écran, affichage Description générée automatiquement

Code mauvais :

Une image contenant texte, capture d’écran, affichage

Description générée automatiquement

Correction : **(Il y avait un .then() en trop)**

Une image contenant texte, capture d’écran, Police, conception

Description générée automatiquement

Résultat du jour :

# Modif fetch datas

Une image contenant texte, capture d’écran

Description générée automatiquement

J’ai changé la [méthode](#_Fetch_les_datas), au lieu de push en boucle, je push un tableau qui contient toutes mes datas

Prendre tous les 8 (8x3 = 24) (data par paquet de 3h sur l’api gratuite)

Une image contenant texte, capture d’écran, affichage, Police

Description générée automatiquement

On recupere les données meteo des 5 prochains jours (7,15,23,31,39) :

Une image contenant texte, capture d’écran, conception

Description générée automatiquement

+ Quelques modifications faites au niveau du code :

Une image contenant texte, capture d’écran, Police

Description générée automatiquementMeilleure lisibilité + toutes les chaines commencent par une majuscule

Une image contenant texte, capture d’écran, affichage, Police

Description générée automatiquement

Permet de générer de la même manière les 4 cases de données stockées dans le tableau (ci-dessous)Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

Une image contenant texte, capture d’écran, conception

Description générée automatiquement

Résultat :

Une image contenant capture d’écran, texte, logiciel, ordinateur

Description générée automatiquement

Résultat :

Probleme avec les div dans « chaine », fatigué, flm de faire des erreurs

* Solution : Full unité relative = minimum nécessaire => min-height pour fix

Ordre :

1. Jour
2. Description

------------------

1. Ville h1

------------------

1. Image img

------------------

1. Temperature h1

------------------

1. Ressenti Data
2. Humidité Data
3. Vent Data

Choix :

~~Faire 5 tableaux et un for pour chaque tableau~~ ???? KEKW

Faire 5 for pour le même tableau

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

Générer dynamiquement :

Une image contenant texte, capture d’écran

Description générée automatiquement

Code :

Ordre :

1. Jour h1

------------------

1. Image img

------------------

1. Temperature h1

------------------

1. Ressenti Data
2. Humidité Data
3. Vent Data

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Code pour un jour +1 mini :

Quel est l’ordre pour faire ma chaine ?

D’abord je dois créer une cellule, puis refaire le procede (boucle)

# GROSSE GALERE POUR GENERER LES STATS DANS LES CARDS JOUR :

Faudra refacto tout ca, mais pour l’instant go sur la merde

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

Une image contenant texte, capture d’écran, logiciel, ordinateur

Description générée automatiquement

Je veux faire une loop 3 fois pour mettre NomCarac[i] + TableauSemaine[k

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

Code :

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

Code que je genere avec le script :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

J’ai modifié le code pour la generation des 3 stats :

Une image contenant texte, capture d’écran, Police, menu

Description générée automatiquement

Code que je genere avec le script :

Une image contenant texte, capture d’écran, Police, logo

Description générée automatiquement

Résultat :

Toutes mes donnees sont injectees dans .jour, je dois créer une div overlay

Une image contenant diagramme, Rectangle, ligne, croquis

Description générée automatiquement

Une image contenant texte, capture d’écran, logiciel, Système d’exploitation

Description générée automatiquement

Résultat final ? :

Reste a refactor et c’est clear

1. Utilisation de fonctions : Vous pouvez découper votre code en fonctions pour le rendre modulaire et plus facile à lire. Par exemple, vous pouvez créer une fonction pour charger les informations météorologiques d'aujourd'hui et une autre pour charger les informations des 5 prochains jours.
2. Éviter la duplication de code : Il y a de la duplication de code pour la création d'éléments HTML. Vous pourriez créer une fonction générique pour créer des éléments HTML, ce qui réduirait la répétition du code.
3. Utilisation de promesses : Au lieu d'utiliser plusieurs gestionnaires .then(), vous pouvez utiliser des promesses pour améliorer la lisibilité du code et éviter les niveaux excessifs d'indentation. Vous pouvez créer des fonctions asynchrones pour les appels API et les appeler avec await.
4. Séparer le HTML, le CSS et le JavaScript : Pour un code plus maintenable, il serait judicieux de séparer votre HTML, votre CSS et votre JavaScript en fichiers distincts.
5. Variables globales : Essayez de minimiser l'utilisation de variables globales pour améliorer la lisibilité et la prévisibilité de votre code.
6. Utilisation de l'ES6 : Vous pouvez tirer parti des fonctionnalités ES6, comme const et let au lieu de var, pour déclarer des variables.
7. Gestion des erreurs : Il est important d'ajouter une gestion des erreurs robuste pour les appels API. Assurez-vous de gérer les erreurs de manière appropriée et d'informer l'utilisateur en cas d'échec.
8. Séparation des préoccupations : Essayez de séparer la logique métier de l'interface utilisateur. Cela rendra le code plus facile à comprendre et à tester.

Voici un exemple de comment vous pourriez refactorer certaines parties de votre code en utilisant des fonctions et des promesses pour améliorer la lisibilité et la maintenabilité :

* CreateDiv
* addClassDiv
* addIdDiv
* CreateElement
* addClassElement
* addTextTableau
* appendChildElement
* appendChildDiv