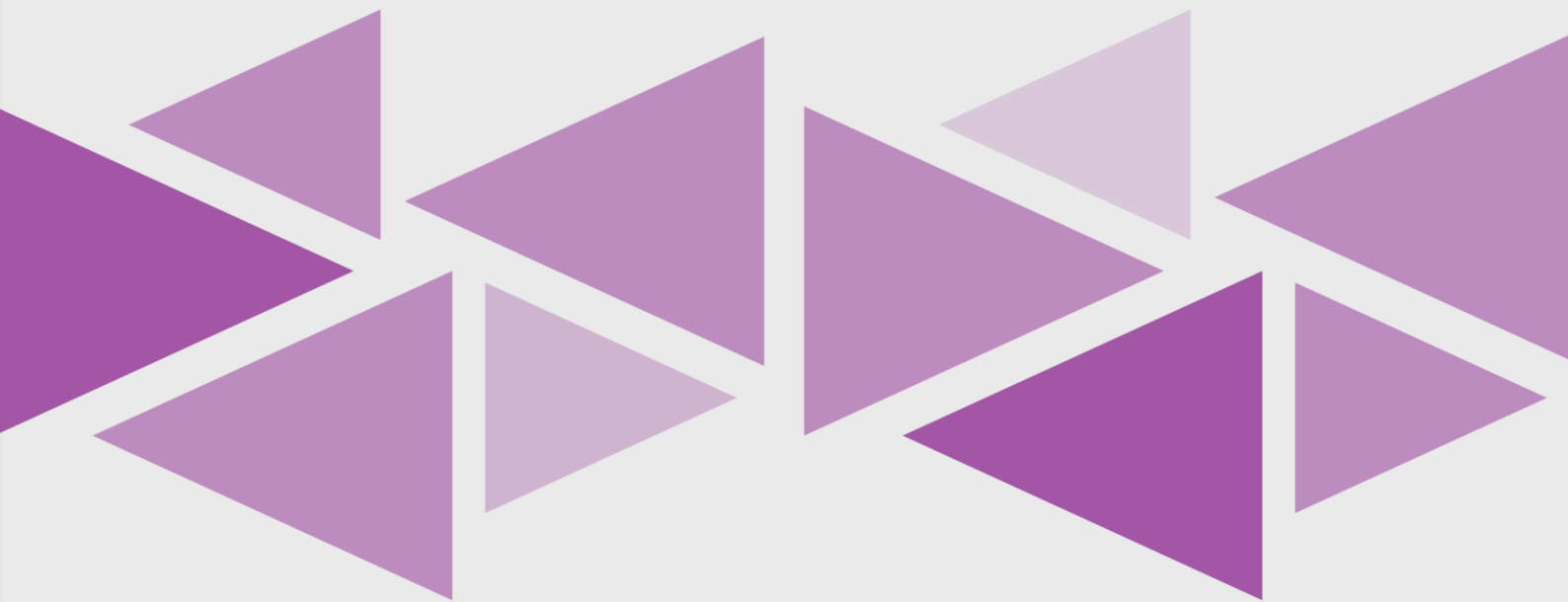


2022

Projet Programmation Web III



Cahier de travail

Part 0 : Fundamentals of Web apps :

- a) General info (22 Avril 13h14)
- b) Fundamentals of Web apps (22 Avril 13h25)

Part 1 : Introduction to React :

- a) Introduction to React (22 avril 14h50 & 6 mai 13h30)
- b) JavaScript (13 mai 14h00 & 20 mai 18h00)

FeedBack:

- a) 6/5/2022 : (6 mai 15h00)
- b) 13/5/2022 : (24 mai 4h00)

Part 0 : Fundamentals of Web apps

a) General info

Descriptif :

Cette partie est une présentation du cours disponible sur le site ainsi que les différents objectifs à atteindre tout en suivant une méthode de travail classique, qui comprend explications théoriques ainsi que des exercices.

b) Fundamentals of Web apps

Descriptif :

Présentation de l'outil console en détail, de ces fonctionnalités utiles telles que « console log » qui permet aux développeurs, un environnement de test dans la console ainsi que présentation d'AJAX combiné avec des protocoles HTTP tel que POST et GET qui permettent la création de pages unique à implémentation dynamique, la page est modifié par rapport aux besoins de l'utilisateur automatiquement sans création de nouvelle page mais directement sur la page même.

Le modèle DOM a été introduit, il définit une certaine structure des éléments de la page et permet par la suite une modification plus simple, la partie Css qui définit le désigne suit le schéma classique de toute application web.

Exercice :

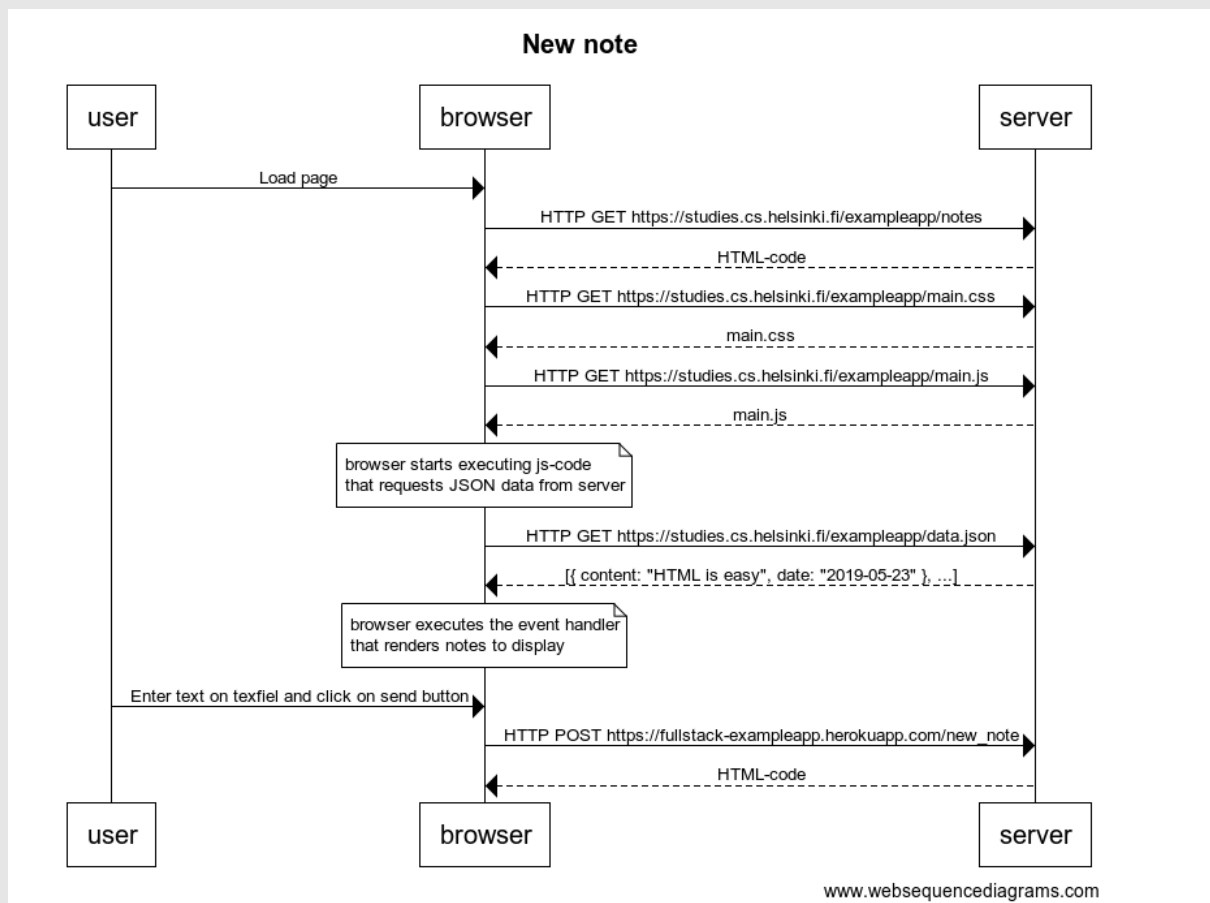
Note : Tous les exercices sont disponibles sur [mon dépôt git](#).

0.1: HTML : Lecture du tutorial.

0.2: CSS : Lecture du tutorial.

0.3: HTML forms : Lecture du tutorial.

0.4-0-5: New note:



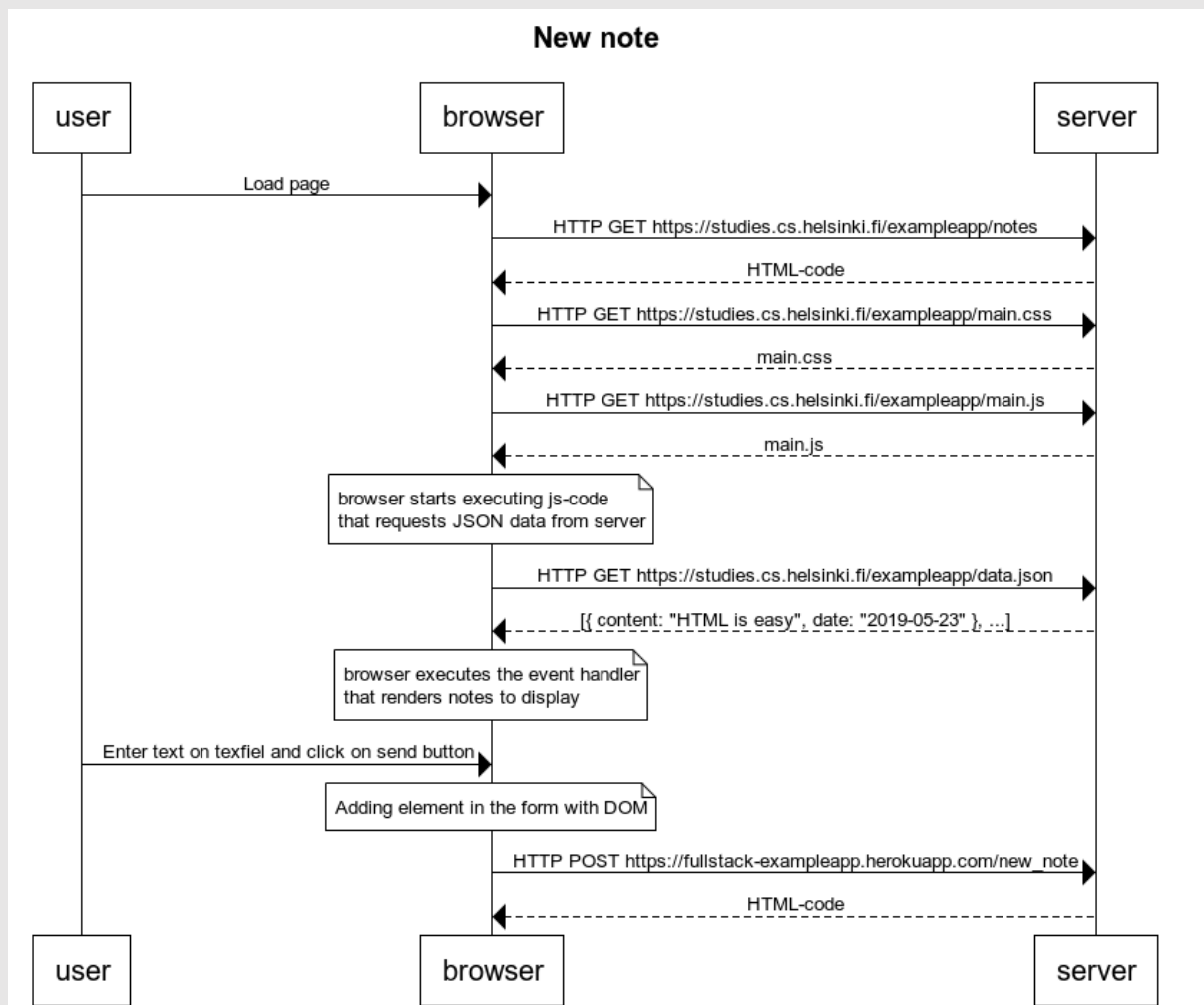
Explication :

Ici le but de l'exercice est de démontrer la situation lorsqu'un utilisateur ajoute une nouvelle note sous forme d'un diagramme.

L'utilisateur qui correspond à l'utilisateur charge la page, ensuite une série de requêtes POST est exécuté vers le serveur afin de récupérer tout le code de la page, toutes les pages sont renvoyées vers les navigateurs afin d'être affichées pour l'utilisateur.

L'utilisateur entre les informations dans le textfield et appuie sur le bouton ce qui va enclencher une requête POST vers le serveur qui va stocker les informations dans les serveurs et va renvoyer de la page html avec la nouvelle donnée.

0.6: New note:



Explication :

La démarche est similaire à l'exercice précédent à la seule différence que les données sont ajoutées sur la page à l'aide du modèle DOM qui va ajouter dans le formulaire de lien la nouvelle donnée automatiquement.

Apprentissage et difficulté :

Au cours de cet exercice une compréhension du fonctionnement des différents outils utilisés avec Node Js, certains de ces outils étaient déjà connus mais un rappel aux bases ne fait pas de mal (HTML, Css, etc.).

Part 1 : Introduction to React

a) Introduction to React

Descriptif :

Comme son nom l'indique cette partie nous introduit aux applications react, dans un premier temps une installation des outils nécessaire est exigée tel que node.js et npm, pour ensuite nous familiariser avec react en créant une application react tout en expliquant les bases du langage.

Installation et outils :

Les outils nécessaires sont NodeJs ainsi que npm pour nodejs, l'installation se fait de façon classique en téléchargeant sur le site officiel et pour l'outil npm l'installation est faite avec nodeJs, afin de vérifier la présence de ces outils sur la machine, les commandes suivantes sont nécessaires, si une version chiffrée apparait l'opération est réussie :

```
C:\Users\Marwa\Desktop\ProjetHELB\WEB4\WEB3\web3\Exercice>npm -v
8.9.0

C:\Users\Marwa\Desktop\ProjetHELB\WEB4\WEB3\web3\Exercice>node -v
v16.15.0
```

Note :

Il arrive que parfois, l'installation se soit mal déroulé et donc une utilisation de npm ou nodejs est non fonctionnelle (voir capture d'écran à la suite), une réinstallation via l'outil d'installation sur le site officiel cité dans le paragraphe ci-dessus, comprend une option réparation ce qui répare les outils endommagés.

```
Creating a new React app in C:\Users\Marwa\Desktop\ProjetHELB\WEB4\WEB3\web3\Exercice\introReact\part1.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

Terminer le programme de commandes (O/N) ? [ ] / reify:lru-cache: timing reifyNo
npm ERR! process terminated
npm ERR! signal SIGINT

npm ERR! A complete log of this run can be found in:
npm ERR! C:\Users\Marwa\AppData\Local\npm-cache\_logs\2022-05-06T11_39_35_607Z-debug-0.log
```

Environnement de développement :

Ici l'environnement de développement utilisé est vscode, documentation disponible [ici](#), vscode est un clone de Visual Studio code, il dispose donc des mêmes fonctions à la seule différence qu'il est Open Source et garantit une confidentialité des données contrairement à Visual Studio code.

[Documentation ainsi qu'installation ici.](#)

Création d'un projet node Js :

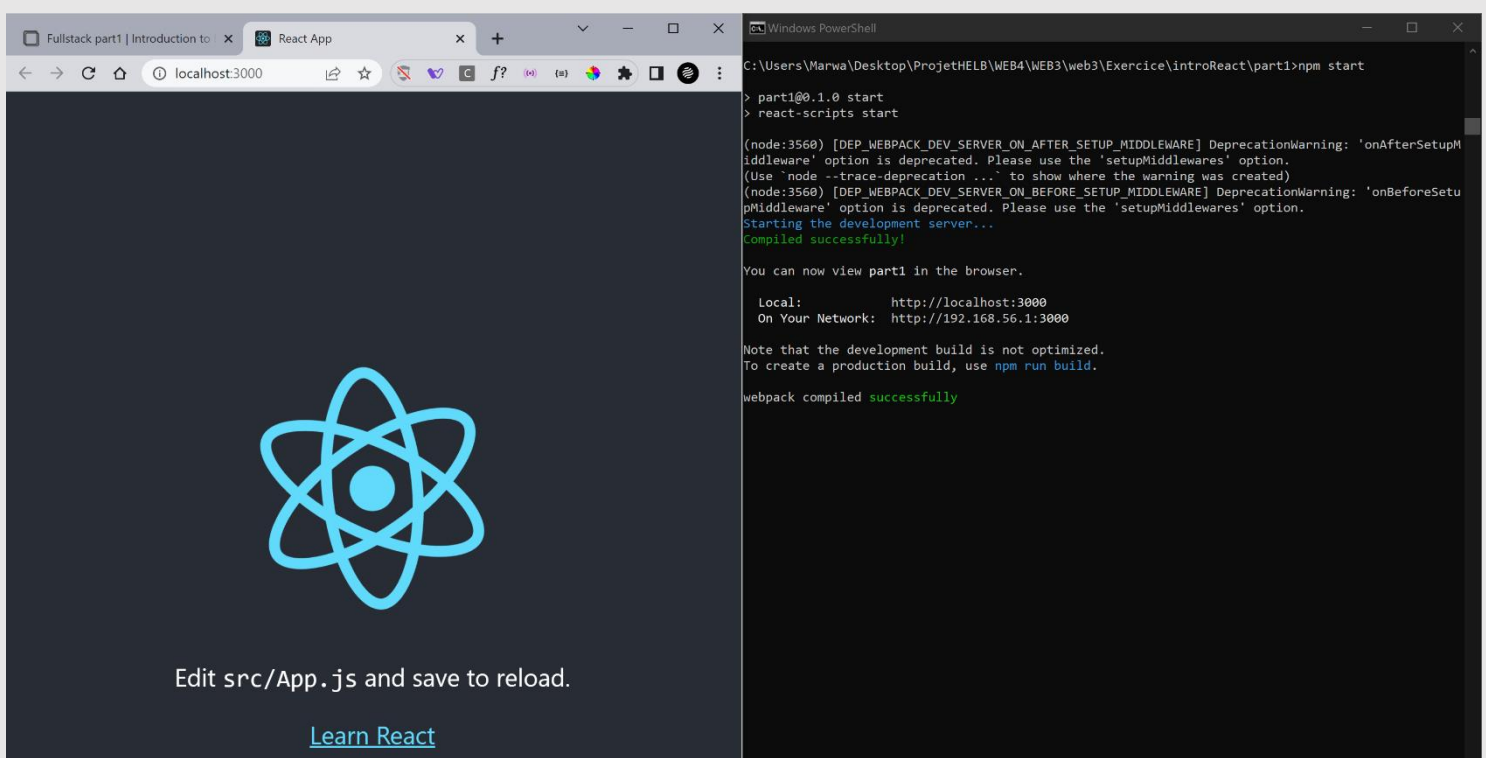
Le projet se crée dans l'invite de commande à l'aide de la commande `npx create-react-app` suivie du nom du projet

Exemple: `npx create-react-app monprojet`

Lancement d'un projet node Js :

Pour le lancement il faut se placer dans la racine du dossier depuis l'invite de commande (cédérom Mon Projet) et exécuté la commande `npx start`, si celle-ci réussie, l'application se lance sur le navigateur sur le port 3000.

Exemple de lancement réussi :



Prise en main de nodejs :

Dans la partie 1 de fullstackopen, on est familiarisé avec nodeJs ,pour se faire l'ont créé une application NodeJs que l'on va manipuler afin de démontrer toutes ces particularités, le répertoire qui nous intéresse est le dossier Src qui contient les fichiers source de l'application ensuite une manipulation des fichiers javascripts est faite et le contenu de la page web est automatiquement mise à jour.

Remarque personnelle :

L'intention mise sur la manipulation de fichiers Javascript indique clairement la nature de NodeJs, la programmation est orienté Javascript, une redécouverte de Javascript a été notée, en effet , à l'aide d'outil tel que DOM ou JQuery, une manipulation, voire une génération dynamique, des pages web étaient possibles mais la dimension prise sous nodejs est tout autre, nodejs permet donc de créer une page complète avec Javascript sans avoir recours à un code HTML.

Exercice :

Note : Tous les exercices sont disponibles sur [mon dépôt git](#).

1.1.-1.2. : Exercices

Explication :

L'exercice met l'accent sur la manipulation de section HTML à l'aide du fichier Javascript, dans cet exercice il est demandé de créer des sections avec des données calculer voire remplacer dans des sections adéquates.

Apprentissage et difficulté :

Familiarisation avec nodejs ainsi que son incorporation au sein d'une page web et de l'affichage de son contenu, les difficultés relève de l'inconnu de cette technique (aucune expérience avec NodeJs)et donc des recherches supplémentaires ont permis d'achever les exercices.

Documentation utilisée : [Forum](#) et [StackOverflow](#).


b) JavaScript

Descriptif :

- Transpilation :

Cette section s'appuie sur le fonctionnement de Javascript en détail, on y apprend un nouveau type de compilation de code, la transpilation qui compile le code source en un code soit plus récent soit plus ancien en fonction de la version supporté par le navigateur à l'aide d'outils tels que babel, ce fonctionnement permet de régler le problème de compatibilité des versions de Javascript avec les différents navigateurs.

Exemple de transpilation :

```
JavaScript  Copy

// Babel Input: ES2015 arrow function
[1, 2, 3].map(n => n + 1);

// Babel Output: ES5 equivalent
[1, 2, 3].map(function(n) {
  return n + 1;
});
```

Note : Grâce à l'outil babel on voit ici une transpilation d'un code Javascript sous la version ES2015 (version de 2015) qui a été transformer en code exécutable sous la version ES5 (version de 2009).

Documentation [Babel](#) et renseignement sur transpilation [ici](#) .

- Manipulation de données :

Var,let,const :

La manipulation de données sous le langage Javascript est beaucoup moins rigoureuse qu'avec java ou en c, Javascript est un langage dynamique qui simplifie au maximum la manipulation de données, l'utilisation de let et var accompagné de constant sont les structures les plus utilisées, un tableau descriptif des différences entre leurs utilisations sera suivi à la suite de ce paragraphe.

	REASSIGNABLE	REDECLARABLE	BLOCK SCOPED	CREATES GLOBAL PROPERTY	TDZ
var	✓	✓	✗	✓	✗
let	✓	✗	✓	✗	✓
const	✗	✗	✓	✗	✓

Explication :

Var a = 5

Reassignable : a =7

Redeclarable : var a =7 ;

Block scoped : {let = a} (déclaré dans un block de code)

Global property : var a = 7 (peut etre réutilisé en dehors du block de code)

TDZ : Existence dans le block de code jusqu'à assignation.

Exemple :

```
{
  // This is the temporal dead zone for the age variable!
  // This is the temporal dead zone for the age variable!
  // This is the temporal dead zone for the age variable!
  // This is the temporal dead zone for the age variable!
  let age = 25; // Whew, we got there! No more TDZ
  console.log(age);
}
```

Documentation disponible [ici](#) sur TDZ.

Les objets :

Les objets sont manipulés de façon simple, des champs de valeurs sont attribués par objet, afin d'à un des champs de l'objet l'utilisation de « . » ou encore « [] » permette d'y accéder.

Exemple

Création d'objet

```
let maVoiture = {  
  make: 'Ford',  
  model: 'Mustang',  
  year: 1969  
};
```

Ou encore

```
function Voiture(fabricant, modele, annee) {  
  this.fabricant = fabricant;  
  this.modele = modele;  
  this.annee = annee;  
}
```

Utilisation de ces valeurs

```
maVoiture.fabricant = Ford  
maVoiture.modele = Mustang  
maVoiture.annee = 1969
```

Les fonctions :

Les fonctions peuvent t'être utiliser de façon standard, de la même manière que son utilisation pour créer un objet dans la section précédente, mais elles peuvent également avoir une utilisation plus souple, plus orientée donnée.

Comme énoncé précédemment une vue en détail de cette manière d'utilisation dite "souple" ou encore orienté "donnée"

Exemple 1

```
const sum = (p1, p2) => {  
  console.log(p1)  
  console.log(p2)  
  return p1 + p2  
}
```

Explication :

Ici l'on voit que la fonction est utilisée comme une variable avec assignation directe, les paramètres sont donnés dans l'assignation et le reste de la fonction est déterminé dans le bloc du code entre crochets

Exercice :

Note : Tous les exercices sont disponibles sur [mon dépôt git](#).

1.3.-1.5. : Exercices

Explication :

L'exercice met l'accent sur la manipulation d'objet, dans un premier temps il est demandé de reprendre les exercices précédents et de reproduire une manipulation semblable à la seule différence, l'utilisation des propriétés d'un objet à restituer dans des sections HTML comme dans les exercices antérieurs.

Apprentissage et difficulté :

Utilisation de concept complexe sous node Js (tel que des objets, des tableaux), expérimentation personnelle afin de mieux comprendre leur structure, difficulté au niveau de l'utilisation de NodeJs car nouvelle technologie mais avec l'appui de documentation ainsi que de pratique, la difficulté d'utilisation s'est transformée en aisance (bien évidemment loin de la perfection).

Documentation utilisée : [Manipulation d'objet](#) , [Manipulation de tableau](#) et [utilisation de fonction](#).

FeedBack :

Descriptif :

Cette partie contient tous les feedbacks émis suite à la qualité du rapport, on y retrouvera les points énoncés ainsi que les appréhensions vis-à-vis du contenu, cette section est subdivisée par paragraphe, chaque résultante d'une date, indiquée par le biais du titre, suivie d'une partie solution, qui présentera les modifications suite aux feedbacks.

6/5/2022:

Une validation du format ainsi que de la qualité écrite à être soulignée ainsi que de la pertinence du contenu, une remarque quant aux explications de certaines parties serait la bienvenue ainsi que la présence de captures d'écran afin d'imager certains propos pour la suite.

Solution:

Ajout d'une partie feedback ainsi qu'un ajout de détails vis-à-vis de l'installation des différents outils le tout accompagné de capture d'écran.

13/5/2022:

Contenu du rapport en accord avec l'intention du cours, les changements ainsi que le nouveau formatage au sein du rapport dû aux remarques de la review précédente ont été validés et finalement le rythme du travail est en accord avec les attentes du cours.

Solution:

- Réorganisation de la partie exercice sur le dépôt git, chaque exercice est uploadé dans un dossier différent ce qui permet de voir tous les exercices séparément et non de façon continue dû à leurs liens qui transforment donc les exercices en un exercice unique et continu, sans oublier la présence de documentations supplémentaires.
- Incorporation de la partie "apprentissage et difficulté" à la fin de chaque partie afin d'avoir une review personnelle vis-à-vis de celle-ci, chaque review comprend à la fois le contenu de mon apprentissage ainsi que les difficultés rencontrées.