Projet TKMD quiz personnalisez :

Cliquez sur la séance pour être redirigez

Tableau récapitulatif des tache accomplie au sien du projet : ↓↓↓↓↓↓

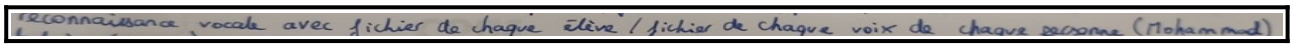
Séance 1	All	Origine du projet	Trouvez le projet	
Séance 2 à 3	Mohammad	WebAssembly/ Script	Trouvez un outil qui permet de lier le python aux navigateurs.	
Séance 3 à 5	Mohammad	Programme python index 2	Deuxième Programme Python	
Séance 4 à 5	Mohammad	Liaison entre script HTML et python	Lier les trois éléments et tester	
Séance 5	Mohammad	Script loader	Faire un loader qui permet de faire patientez le temps que les formulaires du quiz soit prêt.	
Mohammad Rezki	È environ 25%			

Le projet a durée environ sur une période de 5 séance. De ce fait le présentation sera découpé en 5 partie pour raison de clarté.
Concernant la répartition des taches je suis a environ 25 %

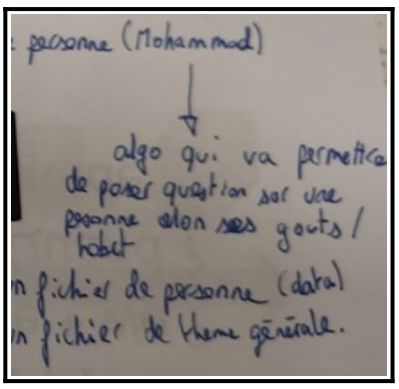
Séance 1

Lors de la première séance nous nous sommes concerté afin de trouver le projet qui nous convenez le mieux, de ce fait chacun à proposer des idées.
Les idées suivante sont celle que j'ai proposé :

- un quiz personnalisée à reconnaissance vocal. Ce quiz pose des question en fonction de la voix identifier auparavant. Des extrait de voix sont auparavant enregistrer auquel nous attribuons des question. Le problème rencontré face a cette idée fut la reconnaissance vocal. A vouloir faire trop grand on risquez de pas réussir. Cette idée fut donc délaissez.↓↓↓↓↓↓



- Et ensuite j'ai basculé sur un quiz personnalisée mais sans reconnaissance vocal



Séance 2 à 3

Dès le 2e séance après distribution des taches je me suis vu affecter comme tache de trouver une solution qui permette de faire le lien entre le HTML et le python ou autrement dit un moyen par lequel le navigateur pourrait lire le programme python. Pour trouvez la solution final je suis passé par deux phase :

- La première étant : flask un micro framework permettant de rendre compatible le programme python avec le navigateur web. C'est a dire a l'aide de tout un programme flask permet au navigateur de pouvoir lire et exécuter la programme tel une page HTML. Le problème que j'ai rencontré dès lors fut le suivant :
  pour pouvoir fonctionner flask nécessite l'installation d'un module en plus au sein de Winpython. Par ailleurs au lycée les droits administrateurs n'étant pas accessible, j'ai du cherché une autre solution.

- La deuxième fut : WebAssembly

Objectifs de WebAssembly

WebAssembly est en cours de création en tant que standard ouvert au sein du W3C WebAssembly Community Group avec les objectif suivants :

- Être rapide, efficace et portable — Le code WebAssembly peut être exécuté à une vitesse proche du natif sur plusieurs plateformes en profitant des capacités matérielles communes.
- Être lisible et débuggable — WebAssembly est un langage d'assemblage de bas niveau, mais son format de texte est lisible par l'homme (la spécification pour laquelle il est encore en cours de finalisation) et permet au code d'être écrit, lu et débuggé à la main.
- Conserver la sécurité — WebAssembly est conçu pour être exécuté dans un environnement sûr, en sandbox. Comme d'autres codes web, il imposera les règles de même origine du navigateur, ainsi que ses politiques d'autorisations.
- Ne pas casser le web — WebAssembly est conçu de manière à facilement s'associer aux autres technologies web et à maintenir une rétrocompatibilité.

Comment fonctionne WebAssembly :

Comme on le voit sur cette capture WebAssembly est natif, portable ce qui exclu donc les problèmes de droit d'administrateur.

- Et donc WebAssembly pour pouvoir lire le python et le retranscrire dans la page web il va utilisez la librairie pyodide : que l'on peut retrouver avec se lien : ↓↓↓↓↓↓

To include Pyodide in your project you can use the following CDN URL:

```
https://cdn.jsdelivr.net/pyodide/v0.19.0/full/pyodide.js
```

En effet il s'agit tout simplement d'un script que nous allons ajouter dans notre page web permettant l'exécution du programme python a l'aide de la librairie Pyodide.

Capture ci dessous : ↓↓↓↓↓↓

```
pyodide.runPython(`
  import sys
  sys.version
`);
```

Ici Pyodide va lancer le programme Python qui lui est donné.

Ce script appelle une librairie (Pyodide) qui est composée de modules permettant l'exécution du programme python car Pyodide est capable d'interpréter le python : dès lors il suffit d'ajouter le programme python. Comme on peut le voir ci-dessous.↓↓↓↓↓↓

```
<!DOCTYPE html>
<html>
  <head>
      <script src="https://cdn.jsdelivr.net/pyodide/v0.19.0/full/pyodide.
  </head>
  <body>
    Pyodide test page <br>
    Open your browser console to see Pyodide output
    <script type="text/javascript">
      async function main(){
        let pyodide = await loadPyodide({
          indexURL : "https://cdn.jsdelivr.net/pyodide/v0.19.0/full/"
        });
        console.log(pyodide.runPython(`
            import sys
            sys.version
        `));
        console.log(pyodide.runPython("print(1 + 2)"));
      }
      main();
    </script>
  </body>
</html>
```

Nous avons ici une certaine subtilité. Car le script est exécuté seulement dans la console, or nous voulons avoir une interaction avec la navigateur et pas seulement avec la console.
C'est pour cela que le code final contient certaines modification contrairement à celui-ci. Ce que nous allons voir plus bas.

Pour résumer durant le deuxième séance j'ai donc chercher à utiliser Flask qui n'a pas fonctionné pour ensuite faire des recherches, tester d'autres alternatives notamment Django, mais qui contient un autre souci pour finir par trouvez WebAssembly. Seulement à ce moment j'ai réellement commencé car maintenant il reste à savoir comment utiliser le programme, le faire fonctionner sachant qu'il n y a pas énormément de tutoriel. J'ai donc du faire le test, manipuler, mais tout ce travaille n'était pas sans résultat.

Séance 3 à 5

En parallèle j'avais comme tâche de faire le programme initial permettant d'exécuter le quiz.
Ce programme à comme fonction d'afficher un input dans laquelle l'utilisateur doit renseigner son nom. Pour ensuite lui proposer un quiz adaptée sur ces centres d'intérêts. Ce quiz est donc personnalisé en fonction de la classe. Cependant nous avons ajoutés une option permettant de renvoyer l'utilisateur s'il est inconnu vers une page avec tous les sujets ou il sera libre de choisir.

Une fois de plus le travaille se déroule en deux parties :

- je fais le programme en question, tout fonctionne mais quand je le termine je me rends compte qu'il faut que je l'adapte aux WebAssembly. Car mon programme de base ne contenait pas de fonction. Ce qui était problématique pour pyodide mais il fonctionnait avec Winpython.↓↓↓↓↓↓

```
import numpy as np

l=['dylan','theo','momo','kev']
### liste des eleves de la classe.


l1=['dylan','theo']
### liste des eleves de la classe qui aime les jeu video

l2=['momo','kev']
### liste des eleves de la classe qui aime les jeu voiture


s=input ("veuillez entrez votre prenom en minuscule et sans majuscule ?")
### phrase qui lance le quiz afin de connaitre l'eleve et de lui proposé le bon


c=("Common Elements", set(l) & set(l1))
### liste des eleves de la classe qui aime les jeu video


for i in l:
    if(i==s):
        print("bienvennue dans le quiz tkmd :  ", s)
        if s==c :
            print ( "ont va parlez de jeu")
        else :
            print ( "ont va parlez de voiture")
```

Explication du programme plus bas.
Mais il fonctionnait avec winpython. ↓↓↓↓↓

```
Python 3.9.8 (tags/v3.9.8:bb3fdcf, Nov  5 2021, 20:48:33) [MSC v.1929 64 bit (AM
D64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: E:\programme quiz\q.py ==========
veuillez entrez votre prenom en minuscule et sans majuscule ?kev
bienvennue dans le quiz tkmd :   kev
ont va parlez de voiture
>>>
```

- J’ai donc du refaire le programme en faisant cette fois des fonctions afin qu’elles puissent être appelées par le module pyodide. La voici :

le programme : ↓↓↓↓↓↓

```
import time
import urllib
import webbrowser
themes=["Dylan","Theo","Mohammad","Kevin","Matthieu","Noah","Reese","Maksim","Charles","Myriam","Michael","Djibril","Matthias","Enola","Tom","Ali","html"]    ### liste des eleves de la classe.

themes1=["Tom"]                                                                                                                                          ### liste des eleves de la classe.
themes2=["Maksim","Charles","Ali","Myriam","Noah"]                                                                                                       ### liste des eleves de la classe.
themes3=["Kevin","Theo"]                                                                                                                                 ### liste des eleves de la classe.
themes4=["Dylan","Michael","Mohammad","Reese","Djibril","Matthieu"]                                                                                      ### liste des eleves de la classe.
themes5=["Matthias"]                                                                                                                                     ### liste des eleves de la classe.
themes6=["Enola"]                                                                                                                                        ### liste des eleves de la classe.
ee1=["html"]                                                                                                                                             ### liste des eleves de la classe.

def convert_time_in_seconds(time_string) :

    for i in themes1:                                                                                                                                    ###comparaison entre l'entré de l'utilisateur et les listes.
        if time_string ==i:                                                                                                                              ### condition si entrée egal au nom de la liste.
            webbrowser.open('themes/theme1_sport/theme/review/index.html')                                                                               ###ouverture de la page en question.
            return f'{time_string }'                                                                                                                     ### recuperation de la valeurs si condition valide.

    for i in themes2:                                                                                                                                    ###comparaison entre l'entré de l'utilisateur et les listes.
        if time_string ==i:                                                                                                                              ### condition si entrée egal au nom de la liste.
            webbrowser.open('themes/theme2_cinema/theme/review/index.html')                                                                              ###ouverture de la page en question.
            return f'{time_string }'                                                                                                                     ### recuperation de la valeurs si condition valide.

    for i in themes3:                                                                                                                                    ###comparaison entre l'entré de l'utilisateur et les listes.
        if time_string ==i:                                                                                                                              ### condition si entrée egal au nom de la liste.
            webbrowser.open('themes/theme3_jeuxvideos/theme/review/index.html')                                                                          ###ouverture de la page en question.
            return f'{time_string }'                                                                                                                     ### recuperation de la valeurs si condition valide.

    for i in themes4:                                                                                                                                    ###comparaison entre l'entré de l'utilisateur et les listes.
        if time_string ==i:                                                                                                                              ### condition si entrée egal au nom de la liste.
            webbrowser.open('themes/theme4_nourritures/theme/review/index.html')                                                                         ###ouverture de la page en question.
            return f'{time_string }'                                                                                                                     ### recuperation de la valeurs si condition valide.

    for i in themes5:                                                                                                                                    ###comparaison entre l'entré de l'utilisateur et les listes.
        if time_string ==i:                                                                                                                              ### condition si entrée egal au nom de la liste.
            webbrowser.open('themes/theme5_automobiles/theme/review/index.html')                                                                         ###ouverture de la page en question.
            return f'{time_string }'                                                                                                                     ### recuperation de la valeurs si condition valide.

    for i in themes6:                                                                                                                                    ###comparaison entre l'entré de l'utilisateur et les listes.
        if time_string ==i:                                                                                                                              ### condition si entrée egal au nom de la liste.
            webbrowser.open('themes/theme6_musique/theme/review/index.html')                                                                             ###ouverture de la page en question.
            return f'{time_string }'                                                                                                                     ### recuperation de la valeurs si condition valide.

    for i in ee1:                                                                                                                                        ###comparaison entre l'entré de l'utilisateur et les listes.
        if time_string ==i:                                                                                                                              ### condition si entrée egal au nom de la liste.
            webbrowser.open('themes/ee/ee1.html')                                                                                                        ###ouverture de la page en question.
            return f'{time_string }'                                                                                                                     ### recuperation de la valeurs si condition valide.

    for i in themes6:                                                                                                                                    ###comparaison entre l'entré de l'utilisateur et les listes.
        if time_string !=i:                                                                                                                              ### condition si entrée egal au nom de la liste.
            webbrowser.open('play.html')                                                                                                                 ###ouverture de la page en question.
            return f'{time_string }'                                                                                                                     ### recuperation de la valeurs si condition valide.
```

Ci dessus, nous avons 4 cadres, c’est a dire un programme composé en 4 parties. Sans allez trop loin dans les détails.
Le cadre jaune le plus grand, c’est le programme complet que va distribuer les thèmes en fonction du nom entré par l’utilisateur.

Entrons maintenant dans le vif du programme :
Le cadre vert contient les listes d’élèves, à chaque liste est associé un thème. Et ces listes vont donc nous permettre de comparer le nom entré par l’utilisateur avec celles-ci afin de lui renvoyez le bon quiz.
Le cadre rouge contient des boucles (un exemple de boucle avec sa condition, cadre bleu) qui elle contiennent des conditions :
Ces boucles nous permettrons de parcourir les liste et vérifier a l’aide de la condition si le nom de l’utilisateur est égal a l’un des nom de cette liste.

Je suppose que vous avez deviné, il y a autant de boucle que de liste et donc que de thème.
Et enfin nous avons mis une fonction qui nous permet de récupérer la valeur de la condition si elle est vérifiée et de la garder et donc de prendre le lien lier aux thèmes.

Séance 4 à 5

Dés lors, il me suffisait de lier le script WebAssembly avec la page HTML et avec le python :

- Premièrement, la liaison entre HTML et script WebAssembly :↓↓↓↓↓↓

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <title>Quiz-Login</title>
    <link rel="stylesheet" href="home.css">
    <link rel="stylesheet" href="https://unpkg.com/tailwindcss@^2/dist/tailwind.min.css">
    <script src="https://cdn.jsdelivr.net/pyodide/v0.18.0/full/pyodide.js"></script>
</head>

<body>
    <div class="loader" id="js-loader">…
    </div>

    <form id="js-conversion-form" class="form hidden">…
    </form>

    <script>
        async function main() {
            const pyodide = await loadPyodide({
              indexURL : "https://cdn.jsdelivr.net/pyodide/v0.18.0/full/"
            });

            pyodide.runPython(` …

            const conversionForm = document.getElementById('js-conversion-form');       //recuperation du formulaire ayant comme id ().

            document.getElementById('js-loader').classList.add('hidden');               // remplacez le formulaire temps qu'il n'est pas pret a l'aide du module hiden.
            conversionForm.classList.remove('hidden');


            conversionForm.addEventListener('submit', (e) => {                          //recuperation de la valeurs et passage dans le programme python et execution.
                e.preventDefault();

                const time = conversionForm.querySelector('input').value;


                pyodide.runPython(`convert_time_in_seconds('${time}')`);
            });
        }

        main();
    </script>
</body>
</html>
```

Comme ont peut le voir ici nous avons trois cadres, le orange et le rouge et le vert. Ici, concentrons nous sur les deux plus grands, le rouge et le orange. (ne pas prendre en compte les cadre vert ni ce qu'il contient.)
Je suppose que vous avez reconnu le rouge : oui en effet, il s'agit du script que je vous est présenté plus haut mais cette fois adapté a mon programme.
En effet on peut voir l' url contenant la librairie pyodide :

```
indexURL : "https://cdn.jsdelivr.net/pyodide/v0.18.0/full/"
```

Ainsi que le lancement du programme python :

```
pyodide.runPython(`convert_time_in_seconds('${time}')`);
```

Pour le cadre orange, il s'agit de la structure html avec un formulaire, une div et un body. Nous allons pas trop rentrer dans les détails. Il suffit de retenir que le script est à mettre a la fin du programme et dans le body de la page HTML.

Ici nous avons le contenu de base :
et le script cadre rouge

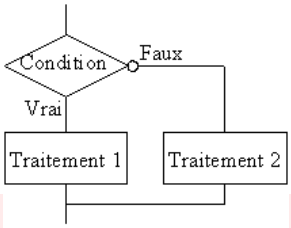
```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <title>Quiz-Login</title>
    <link rel="stylesheet" href="home.css">
    <link rel="stylesheet" href="https://unpkg.com/tailwindcss@^2/dist/tailwind.min.css">
    <script src="https://cdn.jsdelivr.net/pyodide/v0.18.0/full/pyodide.js"></script>
</head>
```

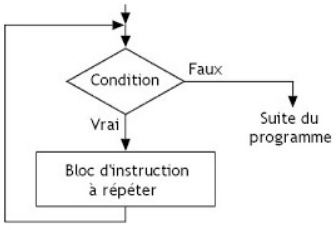
Vous me direz mais dans tous ça il n y a pas de python, en effet :
le voici : ↓↓↓↓↓↓

- deuxièmement ajout du python :
  il suffit de l'ajouter dans les accolades de runpython :

```
async function main() {
    const pyodide = await loadPyodide({
      indexURL : "https://cdn.jsdelivr.net/pyodide/v0.18.0/full/"
    });

    pyodide.runPython(`
        import time
        import urllib
        import webbrowser
        themes=["Dylan","Theo","Mohammad","Kevin","Matthieu","Noah","Reese","Maksim","Charles","Myriam","Michael","Djibril","Matthias","Enola","Tom","Ali","html"]   ### liste des eleves de la classe.

        themes1=["Tom"]                                                                                                                                        ### liste des eleves de la classe.
        themes2=["Maksim","Charles","Ali","Myriam","Noah"]                                                                                                     ### liste des eleves de la classe.
        themes3=["Kevin","Theo"]                                                                                                                               ### liste des eleves de la classe.
        themes4=["Dylan","Michael","Mohammad","Reese","Djibril","Matthieu"]                                                                                    ### liste des eleves de la classe.
        themes5=["Matthias"]                                                                                                                                   ### liste des eleves de la classe.
        themes6=["Enola"]                                                                                                                                      ### liste des eleves de la classe.
        ee1=["html"]                                                                                                                                           ### liste des eleves de la classe.

        def convert_time_in_seconds(time_string) :

            for i in themes1:                                                                                                                                  ###comparaison entre l'entré de l'utilisateur et les listes.
                if time_string ==i:                                                                                                                            ### condition si entrée egal au nom de la liste.
                    webbrowser.open('themes/theme1_sport/theme/review/index.html')                                                                             ###ouverture de la page en question.
                    return f'{time_string }'                                                                                                                   ### recuperation de la valeurs si condition valide.

            for i in themes2:                                                                                                                                  ###comparaison entre l'entré de l'utilisateur et les listes.
                if time_string ==i:                                                                                                                            ### condition si entrée egal au nom de la liste.
                    webbrowser.open('themes/theme2_cinema/theme/review/index.html')                                                                            ###ouverture de la page en question.
                    return f'{time_string }'                                                                                                                   ### recuperation de la valeurs si condition valide.

            for i in themes3:                                                                                                                                  ###comparaison entre l'entré de l'utilisateur et les listes.
                if time_string ==i:                                                                                                                            ### condition si entrée egal au nom de la liste.
                    webbrowser.open('themes/theme3_jeuxvideos/theme/review/index.html')                                                                        ###ouverture de la page en question.
                    return f'{time_string }'                                                                                                                   ### recuperation de la valeurs si condition valide.

            for i in themes4:                                                                                                                                  ###comparaison entre l'entré de l'utilisateur et les listes.
                if time_string ==i:                                                                                                                            ### condition si entrée egal au nom de la liste.
                    webbrowser.open('themes/theme4_nourritures/theme/review/index.html')                                                                       ###ouverture de la page en question.
                    return f'{time_string }'                                                                                                                   ### recuperation de la valeurs si condition valide.

            for i in themes5:                                                                                                                                  ###comparaison entre l'entré de l'utilisateur et les listes.
                if time_string ==i:                                                                                                                            ### condition si entrée egal au nom de la liste.
                    webbrowser.open('themes/theme5_automobiles/theme/review/index.html')                                                                       ###ouverture de la page en question.
                    return f'{time_string }'                                                                                                                   ### recuperation de la valeurs si condition valide.

            for i in themes6:                                                                                                                                  ###comparaison entre l'entré de l'utilisateur et les listes.
                if time_string ==i:                                                                                                                            ### condition si entrée egal au nom de la liste.
                    webbrowser.open('themes/theme6_musique/theme/review/index.html')                                                                           ###ouverture de la page en question.
                    return f'{time_string }'                                                                                                                   ### recuperation de la valeurs si condition valide.

            for i in ee1:                                                                                                                                      ###comparaison entre l'entré de l'utilisateur et les listes.
                if time_string ==i:                                                                                                                            ### condition si entrée egal au nom de la liste.
                    webbrowser.open('themes/ee/ee1.html')                                                                                                      ###ouverture de la page en question.
                    return f'{time_string }'                                                                                                                   ### recuperation de la valeurs si condition valide.

            for i in themes6:                                                                                                                                  ###comparaison entre l'entré de l'utilisateur et les listes.
                if time_string !=i:                                                                                                                            ### condition si entrée egal au nom de la liste.
                    webbrowser.open('play.html')                                                                                                               ###ouverture de la page en question.
                    return f'{time_string }'                                                                                                                   ### recuperation de la valeurs si condition valide.

    `);

    const conversionForm = document.getElementById('js-conversion-form');       //recuperation du formulaire ayant comme id ().

    document.getElementById('js-loader').classList.add('hidden');               // remplacez le formulaire temps qu'il n'est pas pret a l'aide du module hiden.
    conversionForm.classList.remove('hidden');


    conversionForm.addEventListener('submit', (e) => {                          //recuperation de la valeurs et passage dans le programme python et execution.
        e.preventDefault();

        const time = conversionForm.querySelector('input').value;


        pyodide.runPython(`convert_time_in_seconds('${time}')`);
    });
}

main();
```

Comme on peut le voir ci dessus nous avons ce cadre rouge et violet.
Le rouge est le programme python et le violet le script.
En effet nous pouvons constater que le programme est incrémenté dans les accolades de pyodide.
La librairie Pyodide exécute et ou parcourt le programme. Qui elle même est exécuté par le script de WebAssembly (cadre violet). Ci-dessous vous avez le fonctionnement d'une condition et d'une boucle.↓

Une condition ⇒ 

Une boucle ⇒ 

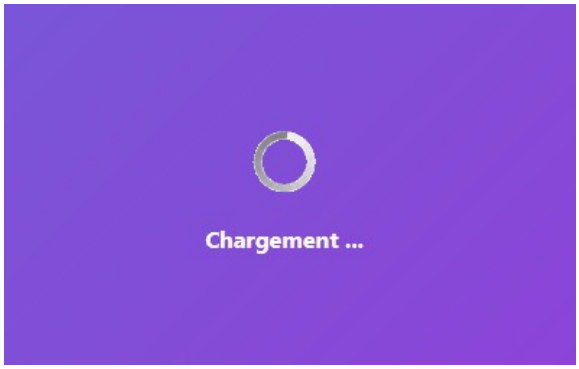
Haut de page ⇒ ↑↑↑↑↑

Séance 5

Pour finir lorsque l'on exécute le quiz nous avions un temps d'attente. En fait il s'agit d'un laps de temps pendant lequel le navigateur va récupérer la librairie pyodide. Des lors nous avions du trouver une solution, après avoir fait des recherches j'ai réussi à trouver une solution, il s'agit en fait d'aller mettre une page chargement le temps que le quiz soit prêt. Soit un loader.

Voici le notre :

Ce loader agit comme un voile devant le programme réel temps que les datas de librairie ne sont pas prêt et permettent ainsi d'éclaircir l'utilisateur . Or s'il n'y a pas de loader il peut y avoir un malentendu et donc pousser l'utilisateur à croire que c'est un beug !



Des lors il fallait donc faire un programme. ↓↓↓↓↓↓↓↓↓

```html
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <title>Quiz-Login</title>
    <link rel="stylesheet" href="home.css">
    <link rel="stylesheet" href="https://unpkg.com/tailwindcss@^2/dist/tailwind.min.css">
    <script src="https://cdn.jsdelivr.net/pyodide/v0.18.0/full/pyodide.js"></script>
</head>

<body>
    <div class="loader" id="js-loader">···
    </div>

    <form id="js-conversion-form" class="form hidden">···
    </form>

    <script>
        async function main() {
            const pyodide = await loadPyodide({
              indexURL : "https://cdn.jsdelivr.net/pyodide/v0.18.0/full/"
            });

            pyodide.runPython(`···

            const conversionForm = document.getElementById('js-conversion-form');       //recuperation du formulaire ayant comme id ().

            document.getElementById('js-loader').classList.add('hidden');               // remplacez le formulaire temps qu'il n'est pas pret a l'aide du module hiden.
            conversionForm.classList.remove('hidden');


            conversionForm.addEventListener('submit', (e) => {                          //recuperation de la valeurs et passage dans le programme python et execution.
                e.preventDefault();

                const time = conversionForm.querySelector('input').value;


                pyodide.runPython(`convert_time_in_seconds('${time}')`);
            });
        }

        main();
    </script>
</body>
</html>
```

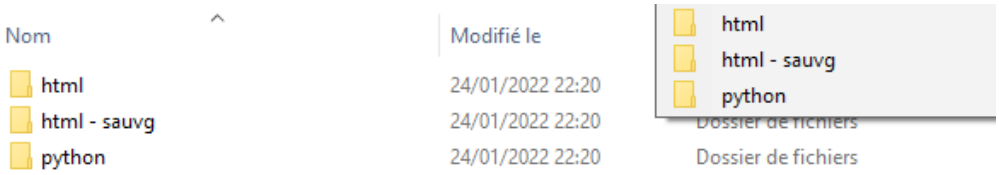
C'est là que entre en jeux le cadre vert. Ce cadre contient en fait du java script et comme l'explique les commentaires. Nous donnons un id au formulaire (cadre jaune). Cet id est en fait comme un numéro de série et il va nous permettre donc de lui appliquer un programme à lui seul (le formulaire).

Organisation :

Dans l'organisation nous pouvons parler de la gestion du temp, la coordination avec les autres personnes du groupe, le rangement des dossiers et fichiers ect…
Commençons par la coordination :
Afin d'être coordonnée avec mes collègues, j'utilisais deux éléments. Le premiers s'agissait du padlet fourni par le professeur : ou encore les tableaux utilisés à même escient :

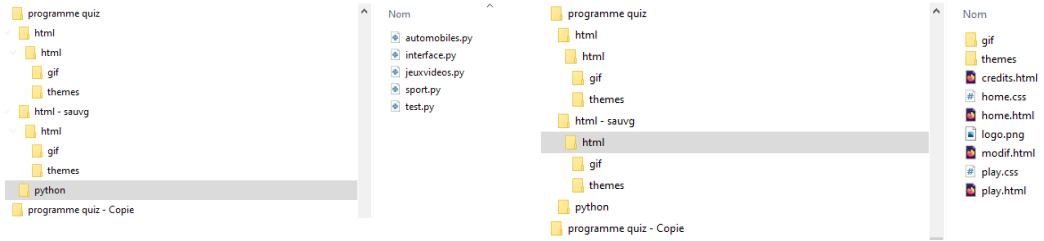
Et ensuite la communication. Car oui la communication est un facteur très important dans un projet en commun ou nous avons des taches partagées et reliées. Et comme outil, j'utilisais donc les appels ou encore en cours lorsqu'on se voyait et surtout discord qui fut l'outil principale notamment pour se partager des documents, se contacter par visio ect…

Quand a l'organisation personnel il s'agit de travailler en classe bien sur, mais étant donné que se soit une matière qui me plaise



énormément je n'ai pas vraiment eu besoin de faire de planning car des que j'avais un moment je me penchais sur le projet.

Et enfin, pour le rangement du travail, j'essaie de faire le plus dans le détail possible c'est a dire de faire le nombre de dossier nécessaire de fichiers ect… quand au code les commentaires ect..De ce fait je contactais mes collègues très régulièrement afin de savoir ou il en sont afin de réussir avoir une coordination assez parfaite. Ensuite les users stories aidés également à mettre au courant les élèves de mon avancée et à moi de savoir ou j'en suis.



Connaissances acquise a travers ce projet.

Enfin les connaissances acquise et mon ressentie.

Connaissances acquise a travers ce projet				
Séance 1	All	Origine du projet	Trouvez le projet	Apprendre a discuté a accepter que mon avis soit prit ou refuser. Cherchez la réussite du projet et non pas seulement la réussite personnel. Apprendre a expliquez a dialoguer ralentir ci nécessaire.
Séance 2 à 3	Mohammad	WebAssembly/ Script	Trouvez un outil qui permet de lier le python aux navigateurs.	Apprendre a faire de la recherche méthodique réfléchir fasse a un problème, savoir s'adapter, regardez le problème différemment apprentissage et un début de compréhension en JS
Séance 3 à 5	Mohammad	Programme python index 2	Deuxième Programme Python	Savoir chercher a retranscrire une idée humaine en langue de programme, utilisée l'outil le plus approprier (condition, boucle, fonction…) et enfin utilisation des notions apprise en NSI
Séance 4 à 5	Mohammad	Liaison entre script HTML et python	Lier les trois éléments et tester	Savoir rechercher, tester essayez de comprendre : un travail de réflexion. Ma permis d'avoir un rappel en HTML
Séance 5	Mohammad	Script loader	Faire un loader qui permet de faire patientez le temps que les formulaires du quiz soit prêt.	Commencez a faire la différence avec un langage de programmation web et un langage de programmation pour domotique et exe. Et un début en JS
Mohammad Rezki	È environ 25%			

A travers ce Projet j'ai appris de nombreuse chose que se soit le coté relationnel interaction ect… du point de vue intellectuel, ou encore de coté organisation. C'est un projet qui m'a plus énormément car il m'a permis d'avoir une image du monde du travail et de faire la différence avec les études.