Pré-requis : pip install intertools + connexion internet (pyscript)

Protocole de Pamplemousse

Ce script Python utilise le module 'itertools' et le module 'window' de la bibliothèque 'pyscript' pour générer et afficher des informations sur les matchs d'un tournoi de manière interactive dans une page web HTML.

- 1. Création des groupes d'équipes (`creer_groupes`): Cette fonction répartit les équipes en un nombre donné de groupes. Chaque groupe est représenté par une liste d'équipes.
 - 2. Affrontement de toutes les autres équipes d'un groupe (`affronter_tous_les_autres`):

Cette fonction affiche les matchs entre toutes les équipes d'un groupe spécifique. Elle utilise la méthode 'document.write()' pour afficher les informations dans la page HTML.

3. Organisation des matchs dans un groupe ('organiser_matchs_dans_groupe'):

Cette fonction génère et affiche tous les matchs pour un groupe donné. Elle utilise des combinaisons de toutes les équipes du groupe pour garantir que chaque équipe affronte toutes les autres équipes du groupe.

- 4. Démarrage du tournoi ('start_tournament'):
 Cette fonction initialise le tournoi en demandant à l'utilisateur le nombre total d'équipes et, éventuellement, le nombre de groupes souhaité. En fonction de ces informations, elle crée les groupes et organise les matchs.
 Elle utilise également des interactions avec la fenêtre ('window.prompt') pour obtenir les informations de l'utilisateur.
- 5. Ajout d'un événement au bouton (`button.addEventListener`): Cette partie du script sélectionne le bouton sur la page HTML et ajoute un écouteur d'événements pour déclencher la fonction `start_tournament` lorsque le bouton est cliqué.

En résumé, ce script Python permet de générer dynamiquement des informations sur les matchs d'un tournoi dans une page web en utilisant des interactions utilisateur et en affichant les données résultantes directement dans la page HTML grâce à la méthode `document.write()`.

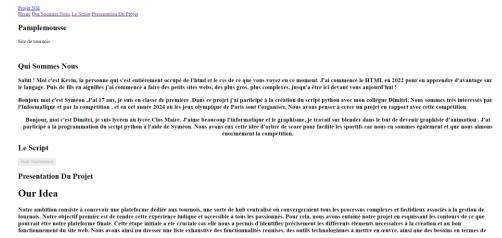
ORGANISATION DU TRAVAIL + REPARTION DES TACHES :

- Html + Css = Kevin Roussel

Script Python = Baudoin Dimitri et Lalarme Syméon

PRESENTATION DES ETAPES DU PROJET:

- 1 On fais le cahier des charges, on recherche des façons de connecter le html et le python
- 2 On trouve pyscript, on regarde la documentation et on l'utilise
- 3 Début du code, on fais un html sans css :



4 J'ajoute le CSS :



- 5 – On commence le python, on importe pyscript :

```
import itertools
from pyscript import document, window

### Le document.write() provient du python pour faire interagir le html directement dans la page web
### Cela permet d'avoir un rendu html avec notre script python

# Fonction pour créer les groupes d'équipes
def creer_groupes(equipes, nb_groupes);

# Initialisation d'une liste de listes, chaque sous-liste représente un groupe
groupes = [[] for _ in range(nb_groupes)]

# Boucle à travers les équipes et les répartit dans les groupes
for i, equipe in enumerate(equipes):
groupes[i % nb_groupes].append(equipe)

return groupes

# Fonction pour affronter toutes les autres équipes d'un groupe
def affronter_tous_les_autres(equipe1, equipe2, groupe_index, match_index):
# Affichage HTML du match entre deux équipes
document.write(f'div id-'match-{groupe_index}-{match_index}'>")
document.write(f'div id-'match-{groupe_index}-{match_index}'>")
document.write(f'divi id-'match-{groupe_index}-{match_index}'>")
document.write(f'divi id-'match-{groupe_index}-{match_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_index}-{groupe_inde
```

 5 bis : On prend du temps à regarder les docs, mais au moins on comprend

OUVERTURE:

On pourra continuer le script en ajoutant la partie pour faire les quarts, demi et les finales afin de vraiment automatiser le tout. Et on pourra même dans le futur, créer un système "graphique" ou on aura vraiment un arbre ou un tableau complet avec le tout.