

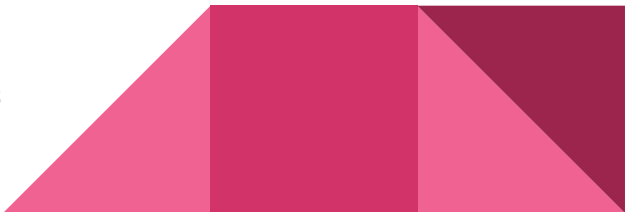
# Partiel 1 - HTML/CSS et JS

JUNIA - AP4 2024/2025  
Antoine DE CONTO



# Modalités du partiel

- La durée du partiel est de 3h
  - Les élèves ont droit :
    - à l'usage de leur ordinateur
    - à l'usage d'internet
    - à l'usage de leurs cours
  - Les élèves n'ont pas le droit :
    - De discuter entre eux
    - D'utiliser l'application Discord ou autre messagerie (hors urgence pour me contacter)
    - D'utiliser leur téléphone portable
    - D'utiliser abusivement de ChatGPT ou autre outil d'IA.

Chaque ligne doit être maîtrisée sinon attention aux sanctions
- 

# Mission

Le but est de reproduire la maquette présente sur cette [maquette figma](#) en HTML/CSS et de rendre le code dynamique grâce au JavaScript et à l'appel de l'API [HP-API](#).

Pour cela, vous allez récupérer le code déjà existant qui a été fait par votre collègue, il a déjà fait la structure HTML et il a déjà fait une grande partie du CSS.

Mais certains bouts du code HTML ont disparu (comme par magie), à vous de les compléter et prendre le relais !

A chaque sous partie, pensez à faire un commit. Le versionning sera pris en compte dans la notation. Donc exemple à l'étape 1.1 puis 1.2 etc ... [\[2 points\]](#)



# Tips et astuces

- Faire des commits réguliers
- Bien lire le document jusqu'à la fin avant de débiter
- Il ne faut pas obligatoirement avoir fini l'étape 1 parfaitement pour faire la partie 2
- Ce que j'attends de vous c'est de me démontrer un peu vos connaissances en HTML/CSS mais surtout l'utilisation du JavaScript avec des API.
- Tous les points sont bons à prendre, je pourrais donner des points pour chaque initiative !



# Partie 1 : Intégration en HTML/CSS - 8 points

1. Créer un repo sur Github pour versionner votre projet. M'envoyer le lien de votre repo à l'adresse suivante : [antoine.deconto@gmail.com](mailto:antoine.deconto@gmail.com)
2. Intégrer la page d'accueil de manière statique [3 points]
  - a. Ajouter les 4 maisons
  - b. Ajouter les 6 personnages (pour l'instant les cercles et noms sont en jaune/or)
3. Ajouter les effets de hover sur chaque maison. Aussi sur les personnages. Mais cette fois-ci, colorier les bords des personnages selon la couleur de leurs maisons (rouge pour Griffondor, vert pour Serpentard etc) [1 point]
4. Intégrer la page de détail d'un personnage de manière statique. [1 point]

## Partie 2 : Utiliser le JS et les API - 8 points

1. Intégrer dynamiquement les personnages grâce à de l'API HP-API. Le contenu dynamique remplace les données statiques de l'étape 1. Limitez-vous aux 12 premières personnes ! Pensez à utiliser fetch, les template literals etc [2 points]
2. En cliquant sur une maison, ça doit trier parmi les personnages présents ! Par exemple, si on clique sur Griffondor, seul Harry, Ron, Hermione et Minerva reste. [2 points]
3. Faire en sorte que dès qu'on clique sur un personnage, on atterrit sur une page dédiée au personnage ! Attention, à ne pas créer 10 000 fichiers details.html, à la place, utilisez la bibliothèque **URLSearchParams** ! [2 points]
4. Remplir les données dynamiquement sur le personnage qu'on vient d'ouvrir [2 points]

## Partie bonus : 2 points

- Faire le responsive. Le menu burger n'est pas obligé d'être fonctionnel. Et pour la version tablette, vous avez quartier libre, tant que le rendu est propre.
- Ajouter des boutons pour trier par ordre alphabétique, par âge etc



# Contraintes

- Doit être valide au W3C HTML et CSS
- Pas d'erreurs ni warning dans la console





# Ressources utiles

- Voici le repo avec la correction des TP vu en cours :  
<https://github.com/antoine-dc/pokeAPI>
- Sur le **URLSearchParams** :
  - <https://builtin.com/articles/urlsearchparams>
  - <https://delicious-insights.com/fr/articles-et-tutos/url-search-params/>



Bon courage :)

