

### TP 3 JavaScript & JSON

#### Exercice 1. Récupération de données dans un fichier Json

- Télécharger les fichiers "TP3-Exo1-Json.html", "TP3-Exo1-Json.json", "TP3-Exo2-Json.html" et "TP3-Exo2-Json.json" disponibles sur Moodle dans le dossier TP3

**Sujet :** l'objectif est de comprendre comment se fait l'intégration de données provenant d'un fichier JSON en Javascript.

Le fichier TP3-Exo1-Json.Json ne contient qu'un seul objet (*un objet json est séparé par des accolades { }*) avec trois données nom, prenom et age (paires nom/valeur) :

```
{  
  "nom" : "Antoine",  
  "prenom" : "Dupont",  
  "age" : "26"  
}
```

Le fichier TP3-Exo1-Json.html contient un script qui permet de sélectionner le fichier Json, de récupérer l'objet et ses données, et de les afficher dans le document (DOM).

**A faire :** Ouvrir dans votre navigateur le fichier html afin de vérifier le fonctionnement du script JS, puis étudier le contenu de ce fichier (script javascript) pour en comprendre le fonctionnement

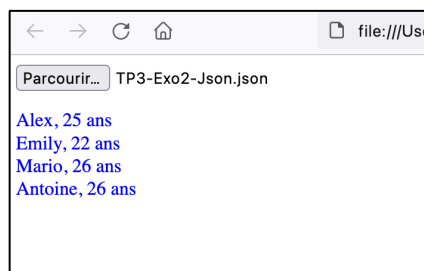
Un fois cette première phase terminée, ouvrir dans votre éditeur les deux autres fichiers : TP3-Exo2-Json.html et "TP3-Exo2-Json.json".

Le fichier Json contient un tableau « personnes » avec 4 objets, chacun avec 3 données : prenom, age et adresse.

**A faire :** Compléter le fichier html et écrire le script JS afin d'afficher les prénoms et âges des 4 personnes.

*Remarque : Il s'agit d'un objet tableau de personne, il faut donc faire une boucle pour traiter chaque donnée de l'objet.*

**Résultat attendu :**



## Exercice 2. Récupération de données Json à partir d'un site web

- ☐ Télécharger le fichier TP3-Exo3-Json.html

**Sujet :** Télécharger des données Json à partir d'un site web et créer dynamiquement une page web qui affiche et met en forme ces données.

Les données sont disponibles sur le site Github suivant :

<https://mdn.github.io/learning-area/javascript/oojs/json/superheroes.json>

### A faire :

1. Ouvrir le lien Github pour comprendre la structure du fichier Json. Quels sont les objets ? comment sont structurées les données (tableau...) ?

Puis, dans le fichier html compléter le script javascript pour :

2. **Étape 1 :** Créer la requête *XMLHttpRequest* afin de se connecter au site et lancer là (cf. cours Json – Diapo 11)
3. **Étape 2 :** Récupérer le résultat de la requête et afficher le dans une fenêtre `alert()`, (cf. cours Json – Diapo 11) . Pour cela on utilise une fonction qui s'activera à l'exécution de la requête (**.onload**)
4. **Étape 3 :** Créer une fonction **enTete** qui va définir dynamiquement :
  - ☐ Une balise `h1` : utilisez la méthode `createElement("h1")`
  - ☐ Ajouter le nom de la *squadName* comme texte à `h1` : utilisez la méthode `textContent()`
  - ☐ Intégrer le `h1` comme child du `<header>` dans le document : utilisez la méthode `appendChild()` ;
  - ☐ Changer la couleur du `h1` en bleu : utilisez la méthode `childNodes.style.color = "blue"` »
  - ☐ Suivre la même logique pour créer une balise `<h2>` dans laquelle vous afficherez les valeurs des données *Hometown* et *Formed in*
5. **Étape 4 :** Créer une fonction **affHeros** qui affichera le nom des différents membres de la *squadName* :
  - ☐ Une balise `h2` avec le nom du membre
  - ☐ 3 balises `p` avec les propriétés : « Secret Identity », « Age » et « Superpowers »
  - ☐ Une boucle avec un `Li` pour afficher les différents « *powers* » de chacun des membres
6. **Étape 5 :** appeler les fonctions **enTete()** et **affHeros()** dans la fonction

➔ cf. page suivante le résultat attendu

← → ↻ 🏠 file:///Users/youzrout/Documents/Yacine/IUT/YO-Cours/

## Super Hero Squad

**Hometown: Metro City Formed in: 2016**

### Molecule Man

Secret identity: Dan Jukes

Age: 29

Superpowers:

- Radiation resistance
- Turning tiny
- Radiation blast

### Madame Uppercut

Secret identity: Jane Wilson

Age: 39

Superpowers:

- Million tonne punch
- Damage resistance
- Superhuman reflexes

### Eternal Flame

Secret identity: Unknown

Age: 1000000

Superpowers:

- Immortality
- Heat Immunity
- Inferno
- Teleportation
- Interdimensional travel