

## **TP1 : Découverte des APIs animales avec XMLHttpRequest**

**Thème :** Utilisation de XMLHttpRequest pour interroger une API d'animaux.

**API suggérée :** RandomDog – Fournit une image aléatoire de chien.

**Consignes :**

- Crée une page HTML avec un bouton "Afficher un chien".
- Au clic, utilise XMLHttpRequest pour récupérer une image de chien depuis l'API.
- Affiche l'image dans un conteneur <div> dédié.

**Structure HTML suggérée :**

```
<button id="btn-chien">Afficher un chien</button>
<div id="resultat-chien"></div>
```

---

## **TP2 : Affichage de faits sur les chats avec fetch()**

**Thème :** Utilisation de fetch() pour récupérer des données textuelles.

**API suggérée :** Cat Facts – Fournit un fait aléatoire sur les chats.

**Consignes :**

- Crée une page HTML avec un bouton "Fait sur les chats".
- Au clic, utilise fetch() pour obtenir un fait sur les chats.
- Affiche le fait dans un paragraphe sous le bouton.

**Structure HTML suggérée :**

```
<button id="btn-chat">Fait sur les chats</button>
<p id="fait-chat"></p>
```

---

## **TP3 : Recherche de livres avec l'API Open Library**

**Thème :** Intégration d'un champ de recherche pour interroger une API de livres.

**API suggérée :** Open Library Search API – Permet de rechercher des livres par titre.

**Consignes :**

- Crée une page HTML avec un champ de saisie et un bouton "Rechercher".
- Lors de la recherche, utilise fetch() pour interroger l'API avec le titre saisi.
- Affiche les résultats (titre, auteur, année) dans une liste.

### Structure HTML suggérée :

```
<input type="text" id="recherche-livre" placeholder="Titre du livre">
<button id="btn-recherche">Rechercher</button>
<ul id="resultats-livres"></ul>
```

---

## TP4 : Liste des animes en cours de diffusion avec `async/await`

**Thème :** Utilisation de `async/await` pour interroger une API d'anime.

**API suggérée :** [Jikan API](#) – Fournit des informations sur les animes.

### Consignes :

- Crée une page HTML avec un bouton "Animes en cours".
- Au clic, utilise `async/await` pour récupérer la liste des animes actuellement diffusés.
- Affiche les titres et les images dans une grille.

### Structure HTML suggérée :

```
<button id="btn-animes">Animes en cours</button>
<div id="liste-animes" class="grille-animes"></div>
```

---

## TP5 : Détails d'un livre avec l'API Google Books

**Thème :** Récupération et affichage détaillé d'un livre.

**API suggérée :** Google Books API – Permet de rechercher des livres et d'obtenir des détails.

### Consignes :

- Crée une page HTML avec un champ de recherche et un bouton.
- Lors de la recherche, utilise `fetch()` pour obtenir les détails du premier livre correspondant.
- Affiche le titre, l'auteur, la description et la couverture du livre.

### Structure HTML suggérée :

```
<input type="text" id="recherche-detail" placeholder="Titre du livre">
<button id="btn-detail">Afficher les détails</button>
<div id="details-livre"></div>
```

---

## TP6 : Comparaison entre `XMLHttpRequest` et `fetch()`

**Thème :** Comparer les deux méthodes pour effectuer des requêtes HTTP.

**Consignes :**

- Crée une page HTML avec deux boutons : "Utiliser XMLHttpRequest" et "Utiliser fetch()".
- Les deux boutons effectuent la même requête vers une API (par exemple, RandomFox) et affichent l'image reçue.
- Comparez le code des deux méthodes et discutez des avantages de `fetch()`.

**Structure HTML suggérée :**

```
<button id="btn-xhr">Utiliser XMLHttpRequest</button>
<button id="btn-fetch">Utiliser fetch()</button>
<div id="resultat-fox"></div>
```