

TP fetch()

Technologies web

Présentation du projet

Une page web d'articles

Objectif



- Développer une **page web** affichant des données récupérées depuis une API (**Interface de Programmation d'Application**) ouverte et gratuite.
- Ce projet vise à appréhender les bases du **langage JavaScript**, en particulier la **manipulation d'éléments** du DOM (**Document Object Model**) et l'utilisation de la **fonction fetch()** pour interagir avec des services web externes.

Attendus



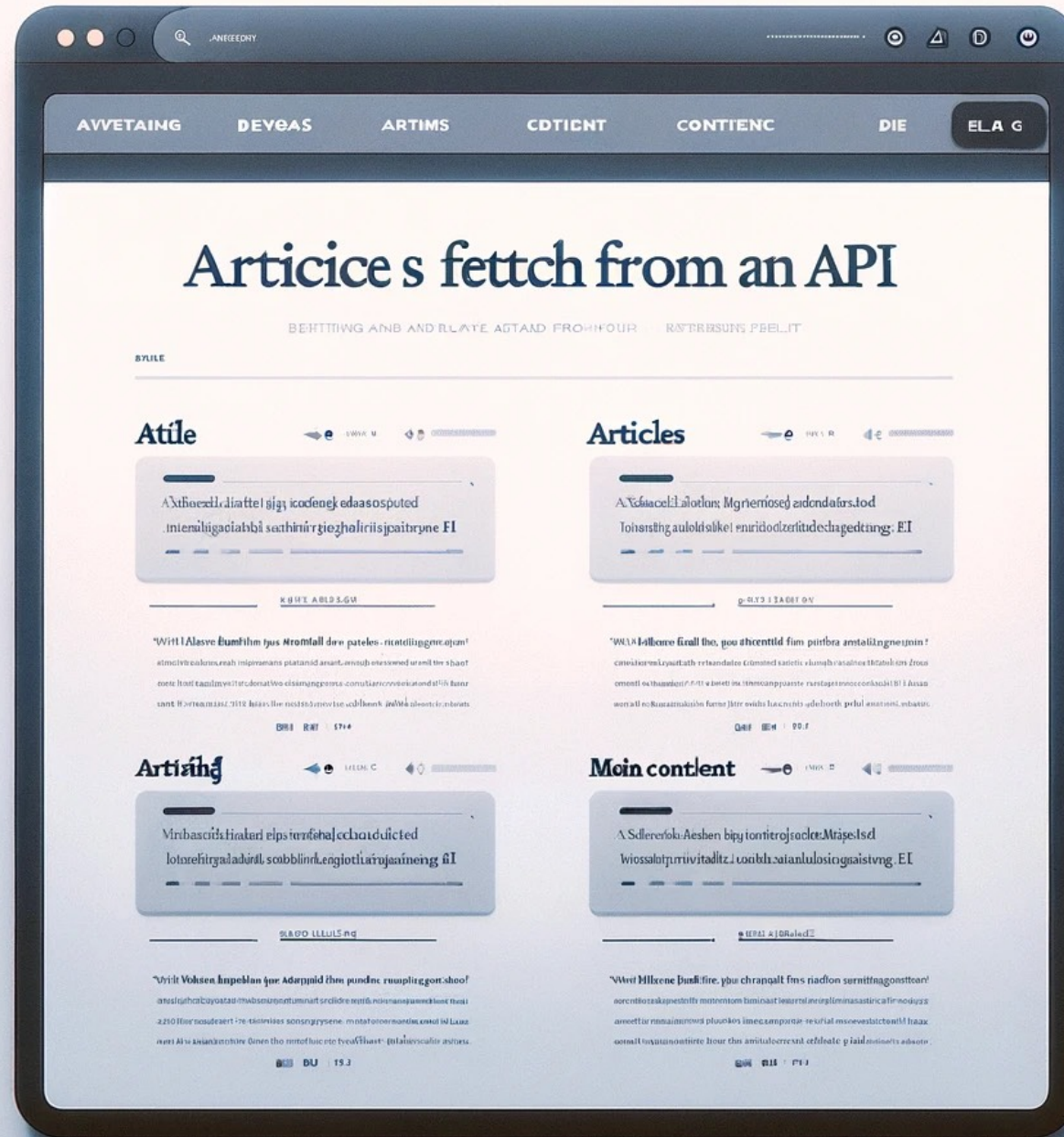
- Une **page web** utilisant les langages **HTML** pour sa **structure** et **CSS** pour son **style**.
- Un **script JavaScript** récupérant des données depuis une **API** fournit et les **affichant** sur la **page web**.
- 💡 Au total, un minimum de **trois fichiers** est attendu : `index.html`, `style.css` et `script.js`.

API utilisée

- Pour simplifier le projet, nous utiliserons l'**API** ouverte et gratuite JSONPlaceholder pour **récupérer** des données.
- Plus précisément, nous utiliserons le endpoint `/posts` pour obtenir une **liste d'articles** fictifs 
- Voici le **endpoint** en question : <https://jsonplaceholder.typicode.com/posts> 

Inspiration graphique

- Voici une capture d'écran illustrative d'une **page web** répondant à l'énoncé du projet.
- Cette image montre comment les **articles** pourraient être présentés sur la **page web**, avec un design minimaliste et moderne.
- Elle sert d'exemple pour que vous puissiez visualiser l'intégration des **articles** récupérés via l'**API**.



Étapes

Une page web d'articles

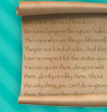
1 - Structure HTML

- Créez un **fichier HTML** nommé `index.html`.
- Dans le corps (`<body>`), ajoutez un **élément** `<div>` avec un identifiant (`id`) spécifique : `id='posts-container'`.
- 💡 Cet *id* nous servira à afficher les données récupérées en **JavaScript** par la suite.

2 - Stylisation CSS

- Créez un **fichier CSS** nommé `style.css` et liez-le à votre fichier `index.html` (`<link>`).
- Ajoutez des **styles de base** pour prévoir une présentation agréable des **articles**, par exemple, un peu de `padding` autour de votre `<div>` et un `style de police` lisible.

3 - Introduction à JavaScript



- Intégrez un **fichier JavaScript** `script.js` en bas du corps (`<body>`) de votre fichier `index.html` (`<script>`).
- Commencez simplement par afficher un message de test dans la console pour vérifier que tout est correctement configuré : `console.log('Script chargé!');`.

4 - Utiliser `fetch()` pour récupérer des données 🌟

- Afficher une **liste d'articles** sur votre **page web** en appelant la **fonction JavaScript** `fetch()` pour les récupérer à l'aide de ce endpoint.

5-1 - Fonctionnalités ✨

- En plus de l'affichage **responsif** de vos **articles**, prévoyez une **barre de navigation responsive** permettant de naviguer entre différentes pages fictives. 🍔 Implémentez la logique d'un menu burger au format mobile (< 768px).
- Ajoutez une **barre de recherche** à votre **page web** permettant de **filtrer** les **articles** et ainsi n'afficher que ceux dont le **titre** et/ou le **contenu** correspond à la **recherche textuelle saisie** par l'utilisateur.
- Ajouter une **page de contact** avec un **formulaire HTML** dont les entrées seront vérifiées à l'aide d'un script JavaScript. Un message devra expliciter les **saisies manquantes ou erronées**.

5-2 - Fonctionnalités ✨

- Par soucis de confort, implémentez une **fonctionnalité de thème clair/sombre** pour votre **site web**.
- ★ **Bonus** : À l'aide d'un **outil gratuit** comme [GitHub Pages](#), [Netlify](#) ou encore [000webhost](#), mettez **en ligne** votre **site web** et **envoyez l'URL** à votre formateur.

6 - Test et observation

- Ouvrez votre page dans un **navigateur** et observez les **articles** fictifs s'afficher.
- Inspectez les **éléments** sur votre **navigateur web** pour voir comment le **JavaScript** a modifié la **page web**/le **DOM** en ajoutant dynamiquement le contenu récupéré depuis l'**API**.

Conseils

- N'hésitez pas à expérimenter avec le **CSS** pour rendre votre page plus attrayante à l'aide de **styles travaillés** ou d'animations originales !
- Vous pouvez ajuster la quantité de données affichées ou encore leur manière de s'afficher en modifiant la logique de votre **script JavaScript**.
- Des **cours** sur le **langage JavaScript** et les points abordés dans ce projet vous sont partagés donc n'hésitez pas à les consulter pour en apprendre plus !
- N'hésitez pas, non plus à **échanger** avec votre formateur ou vos camarades pour mieux comprendre la logique du langage !