

📦 TP JavaScript – Mini Food Truck : Commande de repas en ligne 📦

📦 Objectifs pédagogiques

- Manipuler dynamiquement le **DOM** (création d'éléments, événements, interactions).
- Utiliser les **promesses** avec `fetch` et `async/await`.
- Structurer une application JS simple et interactive.
- Gérer un **workflow complet utilisateur** : menu → panier → récapitulatif → commande → suivi.

📦 Fonctionnalités à implémenter

1. Chargement du menu (via API simulée)

- Charger un menu depuis un fichier `menu.json` ou une [API](#).
- Afficher dynamiquement les plats (image, nom, description, prix).

2. Ajout au panier

- Chaque plat dispose d'un bouton **"Ajouter"**.
- Le panier :
 - Affiche les plats ajoutés avec quantité modifiable.
 - Affiche le total mis à jour automatiquement.
 - Contient un bouton **"Commander"**.

3. Récapitulatif avant commande

- Lors du clic sur **"Commander"** :
 - Afficher une **fenêtre modale** ou une zone dédiée avec un **récapitulatif** :
 - Liste des articles
 - Quantités
 - Prix HT, TVA simulée, Total TTC
 - Boutons : **"Valider"** ou **"Annuler"**.

4. Simulation d'envoi + Suivi de commande

- Après validation :
 - Simuler un envoi avec une fonction asynchrone (`await fakePostCommande()`).
 - Toaster qui notifie la validation de la commande
 - Afficher un **état de progression** :
 1. 📦 Préparation
 2. 📦 En livraison
 3. 📦 Livré !
 - Utiliser `setTimeout` + `async/await` pour simuler les délais.

5. Gestion des commandes

- Une commande en préparation peut être annulée
- Il ne peut pas y avoir plus de 5 commandes en cours
- Toaster en cas d'erreur sur la commande
- Les commandes en cours et le panier doivent être persistants

6. Bonus

- Implémenter des notions non vues en cours
- Appel d'une api externe (autre que pour gérer le menu)

📦 Exemple de menu.json

```
[
  { "id": 1, "name": "Pizza", "price": 10, "image": "pizza.jpg" },
  { "id": 2, "name": "Tacos", "price": 8, "image": "tacos.jpg" },
  { "id": 3, "name": "Salade", "price": 7.5, "image": "salade.jpg" }
]
```

📅 Organisation suggérée (4h30)

Temps	Étape
0h30	Structure HTML/CSS de base
0h45	Chargement dynamique du menu avec <code>fetch</code>
1h00	Ajout au panier + calcul total
0h30	Mise en place du récapitulatif de commande
1h00	Simulation d'envoi + suivi de commande
0h45	Tests, messages d'erreur, finitions UX/UI

📋 Compétences validées

- Sélecteurs DOM, `createElement`, `appendChild`, `addEventListener`
- `fetch`, `async/await`, `try/catch`
- Gestion d'un état client (panier)
- Création d'un mini workflow utilisateur
- Bonus : modale, UX fluide, simulation de backend