



1 - Structure HTML et intégration Bootstrap

💡 Concepts à comprendre

Avant de commencer à coder, prenons le temps de comprendre la structure de votre projet :

Architecture des fichiers

Votre projet contient déjà :

- **index.html** : la page principale qui affiche la liste des voitures
- **car.html** : la page de détail pour une voiture spécifique
- **js/script.js** : votre fichier JavaScript principal
- **js/mock-data.js** : données d'exemple (pour référence uniquement)

Bootstrap : pourquoi et comment ?

Bootstrap est un framework CSS qui vous offre :

- Des composants prêts à l'emploi (cards, modals, navbar, etc.)
- Un système de grille responsive (12 colonnes)

- Des utilitaires CSS pour espacements, couleurs, etc.
- Du JavaScript pour les interactions (modals, collapse, etc.)

Question de réflexion : Regardez le code HTML fourni. Quels composants Bootstrap identifiez-vous ? (navbar, cards, modal, etc.)

✓ Tâches à réaliser

1.1 - Analyser la structure HTML existante

Objectif : Comprendre l'organisation du code HTML avant de le modifier.

Points à observer :

- Dans `index.html`, localisez la section `<section class="card-cont">`
- Vous voyez une `<article class="card">` avec des données en dur (Ferrari 250 GTO)
- Cette carte doit être **supprimée** et **remplacée** par du contenu généré dynamiquement
- Observez également le modal en bas de page (bouton "Add car")

Questions à vous poser :

- Quelle est la structure HTML d'une carte de voiture ?

- Quels éléments doivent être remplis dynamiquement ? (image, titre, description, lien)
- Comment le lien vers `car.html` fonctionne-t-il ? (regardez le paramètre `?id=1`)

1.2 - Comprendre la navigation entre pages

Concept : Query Parameters

L'URL `car.html?id=1` utilise un **query parameter** :

- `car.html` : la page de destination
- `?` : début des paramètres
- `id=1` : paramètre nommé "id" avec la valeur "1"

En JavaScript, vous pourrez récupérer cet ID avec :

JavaScript | ...

```
const urlParams = new URLSearchParams(window.location.search);
const carId = urlParams.get('id');
```

Réflexion : Pourquoi utilise-t-on l'ID dans l'URL plutôt que de stocker les données dans localStorage ou sessionStorage ?

1.3 - Préparer le modal d'ajout

Observez le modal dans `index.html` :

- Il a un bouton "Add car" en position fixe (en bas à droite)
- Le modal lui-même est caché par défaut (classe `modal fade`)
- Bootstrap gère l'ouverture/fermeture via `data-bs-toggle="modal"`

À faire plus tard : Vous devrez remplir le contenu du modal avec un formulaire pour ajouter une nouvelle voiture.

Points de vigilance

- **Ne supprimez pas** les attributs `data-bs-`* de Bootstrap : ils sont essentiels pour le fonctionnement des composants
- **Respectez la hiérarchie** des classes Bootstrap : l'ordre peut avoir de l'importance
- **Gardez le HTML sémantique** : utilisez `<article>`, `<section>`, `<header>`, etc.

