openclassrooms.com

Tirez parti de l’inspecteur pour résoudre vos bugs

4–5 minutes

C’est maintenant l’heure de gloire de l’inspecteur ! 😎

Inspectez votre HTML

Maintenant que vous savez comment ouvrir l’inspecteur sous Chrome et Firefox et visualiser votre HTML, nous allons pouvoir nous intéresser à comment déboguer avec ce dernier !

En effet, grâce à l’inspecteur, vous avez la possibilité d’éditer directement votre code HTML depuis l’outil. Vous allez pouvoir enlever ou ajouter une classe, modifier le contenu d’un `div` , etc.

Cette méthode va vous permettre de tester rapidement une hypothèse pour un débogage. Par exemple, si vous ajoutez une classe à un élément, comment ce dernier se comporte-t-il ?

Cela va, par exemple, vous permettre de résoudre des bugs de sélection CSS.

Cela dit, en pratique, on n’utilise cet outil que pour modifier **quelques** lignes ou **quelques** éléments dans votre page. Si vous devez, par exemple, ajouter une vingtaine de lignes pour tester votre hypothèse, il peut être plus malin de le faire sur votre code

source plutôt que directement sur votre page web.

Inspectez votre CSS

Dans la vidéo de Will, vous voyez comment inspecter les règles CSS qui sont appliquées et aussi comment les modifier en direct. En plus de ce panel d'outils, l'inspecteur peut vous donner la ligne précise où est définie une règle CSS. Cela va vous être particulièrement utile pour déboguer ce dernier.

Un extrait de la page d'accueil de Façadia, le code HTML correspondant et le CSS

Ci-dessus, voici un exemple de la page d'accueil du projet. Vous voyez que la classe `main-title` est stylisée dans notre fichier `main.css` à la ligne 108. C'est quelque chose de particulièrement pratique pour déboguer 🙂.

Dans le chapitre précédent, je vous ai aussi parlé du **modèle de boite**. Grâce à l'inspecteur CSS, vous allez pouvoir observer comment ce dernier se comporte. Cela peut s'avérer particulièrement utile quand vous avez des bugs d'espacement.

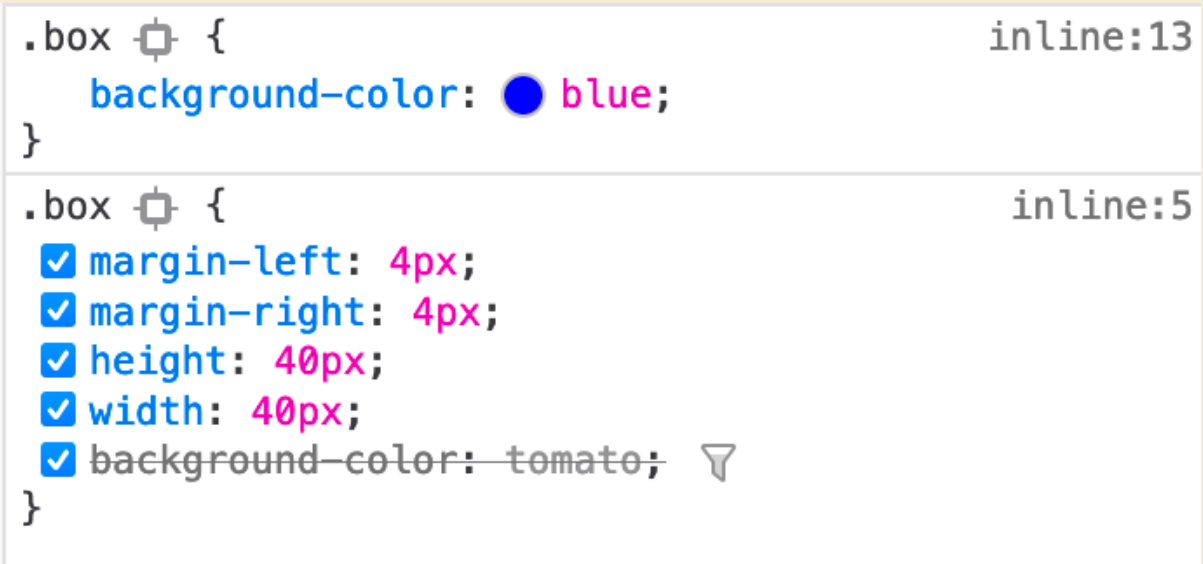
Le formulaire de connexion avec un bug au niveau du bouton "se connecter" et le modèle de boîte correspondante

Ci-dessus, vous voyez un bug côté CSS : le bouton “Se Connecter” est décalé sur la gauche. En utilisant le modèle de boite de l’inspecteur CSS, je constate qu’un margin de -100px est appliqué. Pratique comme outil, n’est-ce pas ?

Avant de passer à la pratique, dernière chose !

Dans le chapitre précédent, vous avez vu que le CSS pouvait générer des bugs d’intégration dus à son comportement en cascade. Là encore, l’inspecteur peut nous aider à traquer cela.

Un exemple de code HTML (à gauche), avec sa feuille de style associée (au centre) et le résultat sur le site web (à droite)



Et le résultat de ce qu’affiche l’inspecteur par rapport à notre CSS

Dans les deux screenshots ci-dessus, vous pouvez voir une règle surchargée. Les trois `div` ayant la classe `box` sont bleues au lieu d’être couleur tomate. L’inspecteur nous montre que la propriété

tomate a été surchargée.

À vous de jouer !

Maintenant que vous avez vu comment utiliser l'inspecteur pour résoudre des bugs d'affichage, passons à la pratique !

Nous allons repartir sur le bug de notre projet Façadia que vous avez vu dans le chapitre précédent. Voici le nom de la branche : partie-2/chapitre-1/section-3/exercice

La solution se trouve sur la branche : partie-2/chapitre-1/section-3/exercice-solution

En résumé

- Les bugs d'intégration peuvent être liés à des erreurs côté HTML, CSS et navigateurs. Le plus souvent, ces erreurs seront dites "silencieuses". Elles ne feront pas planter votre application.
- Les "Vendor Prefix" correspondent aux préfixes dédiés à chaque navigateur. Ils vous permettent d'implémenter des propriétés CSS non standardisées.
- Pour analyser une erreur, servez-vous des validateurs HTML et CSS et du site CanIUse. Cela peut parfois être une solution miracle.
- L'inspecteur va vous aider à mieux comprendre comment se comporte votre CSS, par exemple au niveau du modèle de boite (Box Model)

Félicitations, vous avez appris à résoudre des bugs d'intégration et vous êtes maintenant fin prêt à passer à l'étape d'après : les bugs

concernant le JavaScript et le DOM !