

Prenez en main les outils de débogage

5–7 minutes



Dans le chapitre précédent, nous avons vu ensemble les différentes sources de bug possibles sur une interface front-end.

Vous allez maintenant voir comment tirer parti des nombreux outils de votre navigateur pour suivre et traquer les bugs !

Utilisez un environnement interactif grâce à la console

Première étape de notre tour du monde des outils : **la console** !

C'est certainement **l'un des outils les plus intéressants** que vous ayez dans le monde du développement front-end.

Maintenant qu'on sait y accéder, à quoi ça sert et comment s'en servir ?

Première chose à savoir : **la console est un environnement de programmation interactif** . Grâce à cet environnement, vous allez pouvoir :

- initialiser et stocker des variables ;
- analyser les erreurs JavaScript ;
- afficher (on dit aussi logger) des informations (fonctions, variables, objets, etc.)

- et même écrire des fonctions pour les exécuter directement.

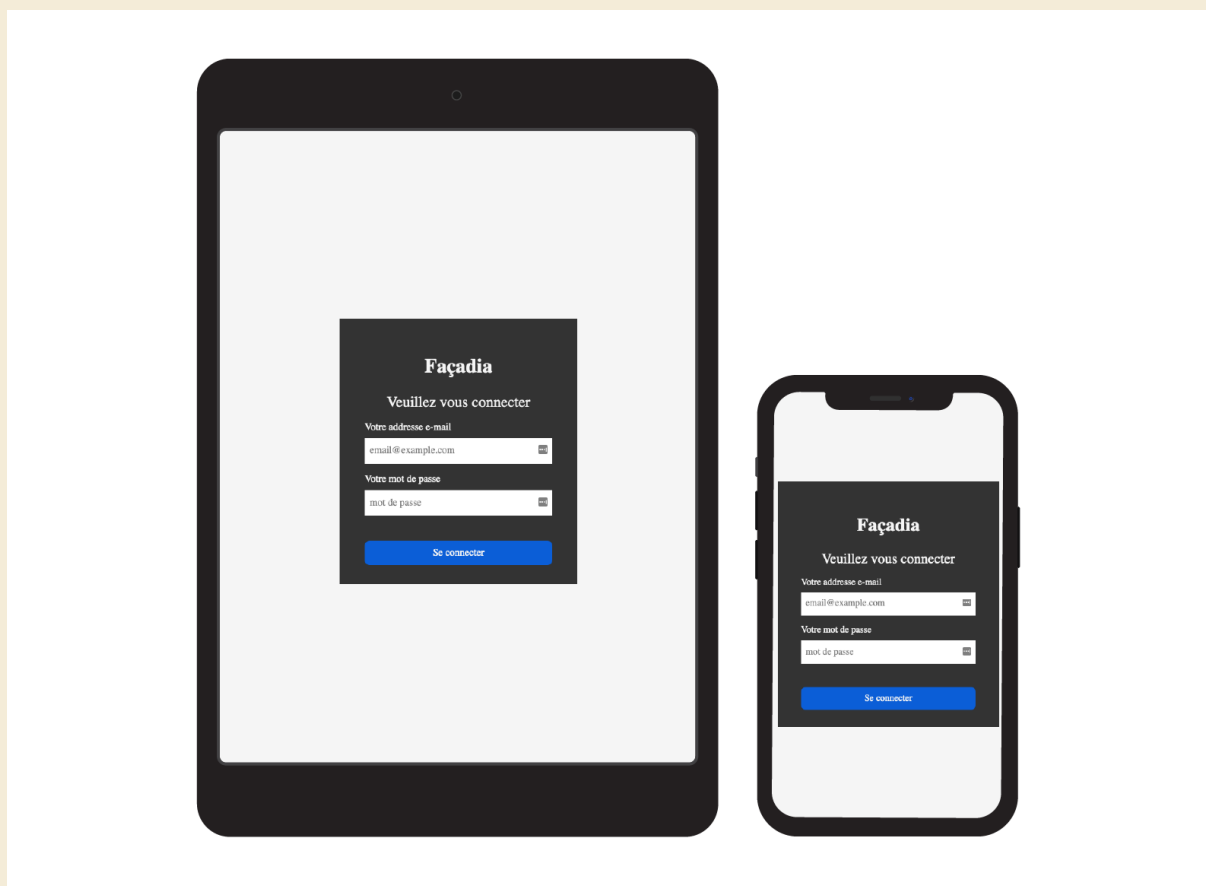
Cette dernière va nous être particulièrement utile lors de vos débogages : elle se prend très facilement en main, est très puissante et vous permet de directement coder du JavaScript dans votre navigateur.

Une bonne compréhension de la console est essentielle pour bien déboguer vos applications front-end ; ça tombe bien, vous allez pouvoir pratiquer car nous allons l'utiliser énormément dans ce cours !

Visualisez et éditez vos règles CSS avec l'inspecteur

Deuxième étape dans notre tour du monde des outils du débogage : **l'inspecteur** ! Vous allez principalement vous servir de cet outil pour **déboguer** votre CSS et votre HTML ainsi que pour **tester** rapidement une solution (en ajoutant quelques règles CSS directement depuis l'outil).

L'un des principaux avantages de l'inspecteur est de pouvoir observer comment se comporte l'interface avec différentes résolutions et/ou différents types d'écran.



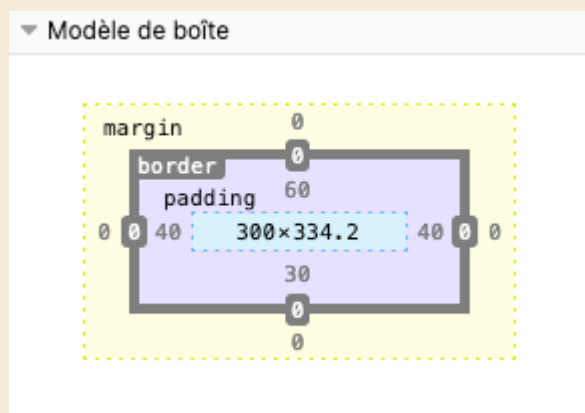
La page d'accueil de Façadia sur deux terminaux de taille différente

De plus, l'inspecteur vous donne parfois la **ligne correspondant à la propriété CSS**. Vous pourrez aussi voir si les propriétés sont surchargées à certains endroits.

```
.sign-in-main-wrapper {  
  background-color: var(--color-secondary);  
  color: var(--color-primary);  
  padding: 60px 40px 30px;  
}  
  
Hérité de body  
  
body {  
  color: var(--color-secondary);  
}
```

Ici on voit que la propriété color du body est surchargée dans la classe `sign-in-main-wrapper`

Enfin, l'inspecteur vous permet de visualiser les **règles d'espacement** qui sont appliquées au modèle de boîte (ou box model en anglais). Cet outil de visualisation bien pratique va vous permettre de **traquer pourquoi un contenu dépasse de sa "boîte" et entraîne un bug visuel**.



Exemple d'un modèle de boîte avec l'affichage des règles d'espacement

Suivez étape par étape l'évolution de votre code avec le débogueur

Troisième étape de notre tour du monde des outils de débogage : le célèbre **débogueur** ! Cet outil bien pratique vous permet de **suivre l'état de votre code étape par étape**.

En ajoutant des **points de rupture** (ou **breakpoint**), vous allez savoir quelles fonctions vont être appelées, quel sera l'état des

variables, etc. À la différence du console.log, le débogueur vous permet de comprendre plus finement et plus précisément comment se comporte une application.

Cela dit, avant de vous présenter le dernier outil dédié au débogage, je vous ai réservé une petite surprise ! Dans ce cours, nous allons utiliser **VsCode** comme éditeur de texte, car ce dernier dispose d'une extension particulièrement pratique dédiée au débogage (et au débogueur).

Vous allez pouvoir utiliser cette extension avec Google Chrome. S'il n'est pas installé sur votre ordinateur, je vous invite dès à présent à le faire.

Dans la vidéo ci-dessous, je vous montre comment installer et paramétrer cette extension VsCode sur Chrome.

Analysez vos appels réseaux avec l'onglet Network de votre navigateur

Dernière étape de notre tour du monde des outils de débogage : **l'onglet Network** ! C'est l'un des outils dont on parle le moins souvent et qui demeure pourtant essentiel pour déboguer.

Cet outil va vous permettre de **traquer toutes les requêtes et réponses reçues par votre navigateur**. Dans ce cours, vous utiliserez principalement cet outil pour déboguer nos requêtes HTTP.

Dans la vidéo ci-dessous, je vais vous présenter où se trouve cet outil et comment vous pouvez vous en servir.

En plus de la partie débogage des API, **l'onglet Réseaux** vous permet de connaître **le poids des fichiers reçus**. Cela peut être particulièrement intéressant si vous souhaitez optimiser le chargement de votre site (ou au contraire, comprendre pourquoi ce dernier met du temps à charger 😊).

En résumé

- La console est un environnement interactif où vous allez pouvoir créer des variables et des fonctions, exécuter du code ou afficher des informations sur l'état de votre application.
- L'inspecteur vous permet de consulter votre HTML et votre CSS. Vous pourrez modifier votre code HTML (en ajoutant des classes par exemple) mais aussi activer et désactiver des règles CSS.
- Le débogueur vous permet de suivre le déroulement de votre code étape par étape grâce à des breakpoints (ou points de rupture). À chaque breakpoint, votre code se stoppe et vous permet de consulter l'état de votre application.
- L'onglet Network (ou réseau en français) est utilisé tant pour analyser des requêtes HTTP que pour optimiser le chargement de votre site.

Vous êtes maintenant prêt à vous lancer dans le débogage de vos projets ! Dans le prochain chapitre, nous allons parler des étapes à suivre pour déboguer vos applications.

- <#>