

Détectez des bugs HTML et CSS

8–11 minutes

Maintenant que vous avez vu les différents outils à votre disposition ainsi qu'une méthodologie de débogage, vous pouvez désormais passer à l'action et résoudre votre premier type de bug : **les bugs d'intégration** !

Dans ce chapitre, nous allons nous concentrer sur **les causes** des bugs d'intégration et nous allons les analyser. Nous verrons dans le prochain chapitre comment nous servir de l'inspecteur pour les résoudre.

Identifiez les bugs d'intégration

Les bugs d'intégration correspondent aux bugs que nous pouvons rencontrer côté HTML et CSS. Ces derniers s'avèrent souvent assez compliqués à résoudre. En effet, à la différence du JavaScript, les erreurs d'intégration ne soulèvent pas d'erreurs dans la console ou ne font pas "planter" l'intégralité de votre application.

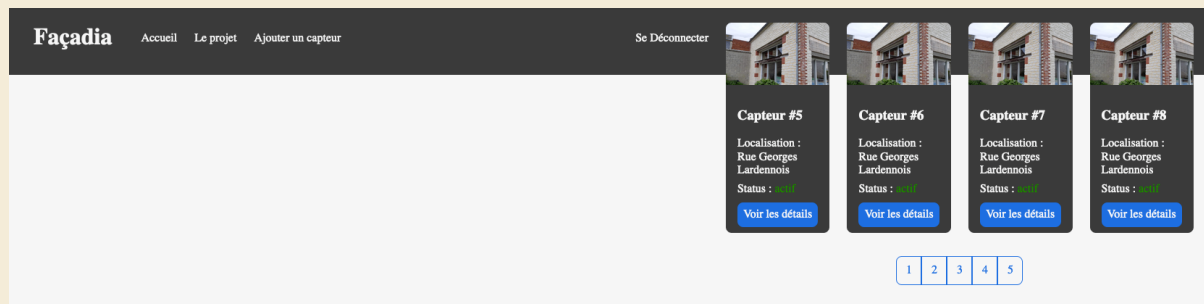
Ils peuvent être causés par de nombreuses raisons et ça tombe bien, nous allons justement voir ces raisons dès maintenant !

1. Les erreurs liées au HTML

Côté HTML, la cause la plus fréquente d'erreur est souvent **l'oubli de fermeture d'une balise**. Ne rigolez pas, cela arrive beaucoup plus souvent que vous le pensez, et à moi le premier !

Une balise non fermée peut entraîner un comportement un peu bizarre côté navigateur : ce dernier va s'emmêler les pinceaux et

votre affichage risque de “sauter”.



Les blocs des différents capteurs sont mal positionnés sur la page

Autre erreur classique : l'utilisation d'une **balise dépréciée ou expérimentale**.

2. Les erreurs liées au comportement du CSS

Est-ce que vous vous souvenez de la signification du terme CSS ? Cascading Style Sheets, ou Feuille de style en cascade en bon français. Et concrètement, qu'est-ce que cela veut dire ? Autre question : est-ce que vous vous souvenez de la règle des priorités (on dit aussi spécificités) en CSS ?

Si je vous rappelle ces deux règles, **c'est parce que de très nombreux bugs côté CSS surviennent parce qu'elles ne sont pas respectées**. Face à un bug côté CSS, il convient donc que vous preniez bien le temps de regarder si une propriété CSS n'est pas surchargée par une autre !



Dans l'exemple ci-dessus, vous pouvez voir une erreur de CSS

pour le texte. L'inspecteur, dans le screenshot de droite, nous indique la ligne ainsi, que le fichier, à laquelle la couleur est surchargée (ici, la ligne 179).

L'autre cas de bug très classique, et très sournois aussi, est la **mauvaise sélection d'un élément HTML** auquel vous faites appel dans votre CSS, par exemple une faute de typographie dans le nom de la classe sélectionnée.

```
/* Sensor Detail Page */

.sensor-detail-main { /* Le bug de sélection est ici
*/

    margin-left: auto;

    margin-right: auto;

    max-width: 1024px;

    width: 80%;

}
```

```
<main class="sensor-details-main">

  <div ckass="sensor-details-main-top-header">

    <h2 class="section-tile">Details du capteur
#7</h2>

  </div>

  <div class="sensor-details-wrapper">

    <div class="sensor-details-img-gallery-
wrapper">

      <img class="sensor-details-img" src=""
alt="" />

      <ul class="pagination-list">

        <li class="pagination-list-item prev-
item">

          <a href="#"><<</a>
```

```

        </li>

        <li class="pagination-list-item next-
item">

            <a href="#">>></a>

        </li>

    </ul>

</div>

</div>

</main>

```

Dans l'exemple ci-dessus, une balise HTML a été mal sectionnée. Cela provoque une cassure du design comme vous pouvez le voir dans l'image ci-dessous. Dans ce type de cas, vous allez devoir chercher à la main avec l'inspecteur d'où peut provenir l'erreur.

Détails du capteur #7



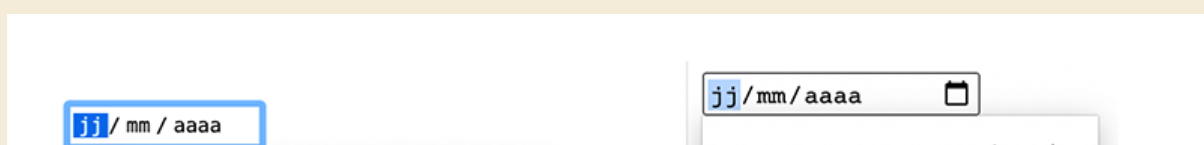
#	Type de données	Valeur des données
1	ID	232124
2	Marque du capteur	facadiaRT
3	Status	Actif
4	Latitude	12.23
5	Longitude	-18.23
6	Température	17
7	Degrée d'humidité	30 %
8	Date de dernière visite	23/02/2021
9	ID du technicien	123456

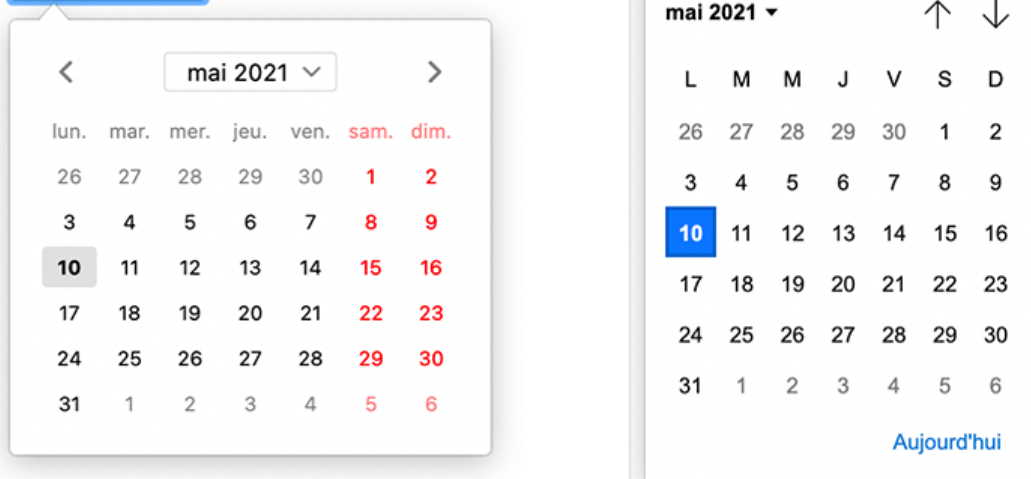
Bulletin météo

Le visuel de la façade à gauche, s'affiche mal

3. Les erreurs liées aux navigateurs

Il arrive régulièrement que des bugs puissent être liés à des **incompatibilités navigateurs**. Par exemple, pendant très longtemps, Firefox ne supportait pas l'input date : Chrome affichait un DatePicker alors que Firefox n'affichait rien ; on devait alors passer par une librairie JavaScript pour harmoniser ce comportement.





Les input de type date sous Firefox et Chrome

Heureusement, l'époque de la forte incompatibilité entre les navigateurs est derrière nous. Cela dit, il existe encore des moments où vous allez devoir gérer ce cas de figure. Par exemple, si vous souhaitez réaliser des animations CSS. Dans ce type de cas, vous allez devoir passer par les "Vendor Prefix".

```
/* ici des vendors prefix utilisés pour la propriété
CSS: transition */

.card {

    -webkit-transition: all 4s ease;

    -moz-transition: all 4s ease;

    -ms-transition: all 4s ease;

    -o-transition: all 4s ease;

    transition: all 4s ease;

}
```

Dans le snippet de code ci-dessus, vous pouvez voir la propriété transition avec un vendor prefix.

Vous avez peut-être déjà vu `-webkit-` , `-moz-` ou même `-ms-` dans une base de code CSS ? Il s'agissait d'un "Vendor Prefix" et il était là pour gérer la compatibilité entre les navigateurs.

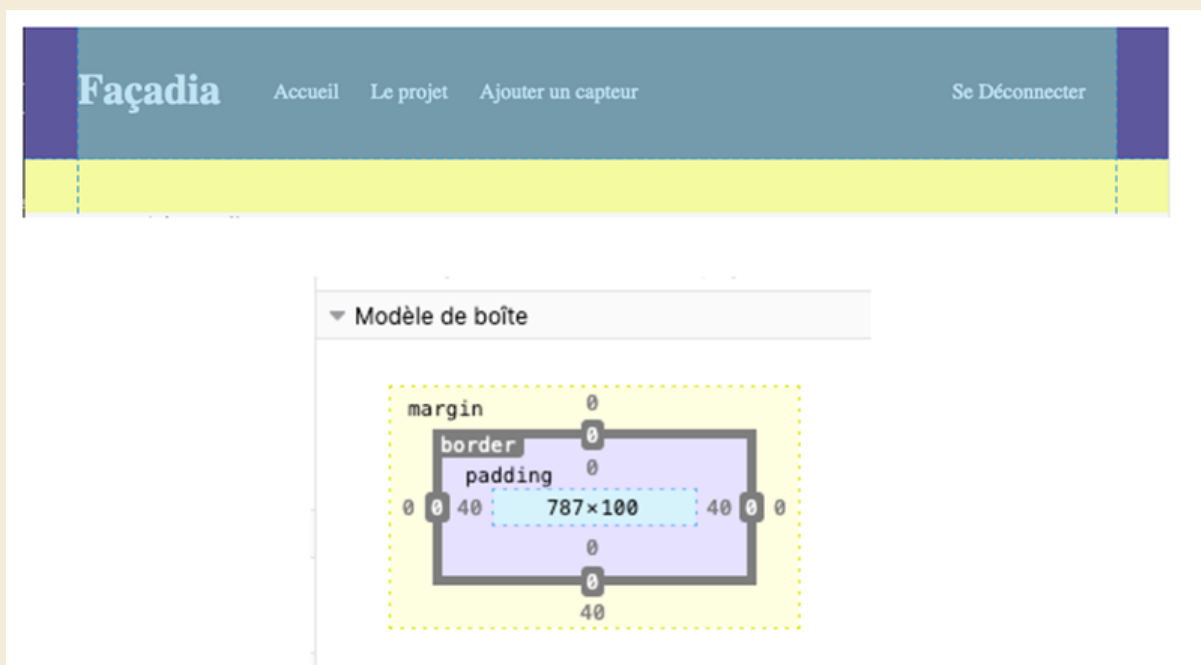
Analysez votre bug

Utiliser l'inspecteur

Je ne vous présente plus l'inspecteur ! 😊

Ce formidable outil va nous permettre d'**analyser votre code HTML et CSS** et plus particulièrement :

- de changer la résolution ;
- de changer le type de devices ;
- d'inspecter les modèles de boîte, ou Box Model, et de voir précisément les margin et padding utilisés ;
- de voir si certaines règles CSS sont surchargées ou ne sont pas appliquées.



Un exemple du modèle de boîte sous Firefox, pour le header de notre site Façadia.

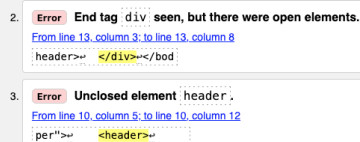
Utiliser un validateur HTML et CSS

Recourir à un validateur peut parfois être une solution miracle.

En effet, il arrive d'être totalement coincé devant un bug : on ne comprend pas d'où il vient et comment le résoudre. On a essayé l'inspecteur, on a parcouru notre code une bonne dizaine de fois mais rien n'y fait.

Dans ce type de cas, je vous invite à copier/coller votre code dans

un [validateur HTML](#) et/ou dans un [validateur CSS](#) et voir si des erreurs et warnings remontent. Les validateurs pourront vous montrer des éléments que vous n'aviez pas vus, comme, par exemple, une balise HTML qui n'est pas correctement fermée :

- 
- A screenshot of an HTML validator interface showing two error messages. The first error, labeled '2. Error', states 'End tag <div> seen, but there were open elements.' and points to 'From line 13, column 3: to line 13, column 8' in the code snippet 'header><div></div></body>'. The second error, labeled '3. Error', states 'Unclosed element <header>.' and points to 'From line 10, column 5: to line 10, column 12' in the code snippet 'per"><header>'.
2. **Error** End tag `<div>` seen, but there were open elements.
From line 13, column 3: to line 13, column 8
`header><div></div></body>`
 3. **Error** Unclosed element `<header>`.
From line 10, column 5: to line 10, column 12
`per"><header>`

Message du validateur suite à une balise mal fermée

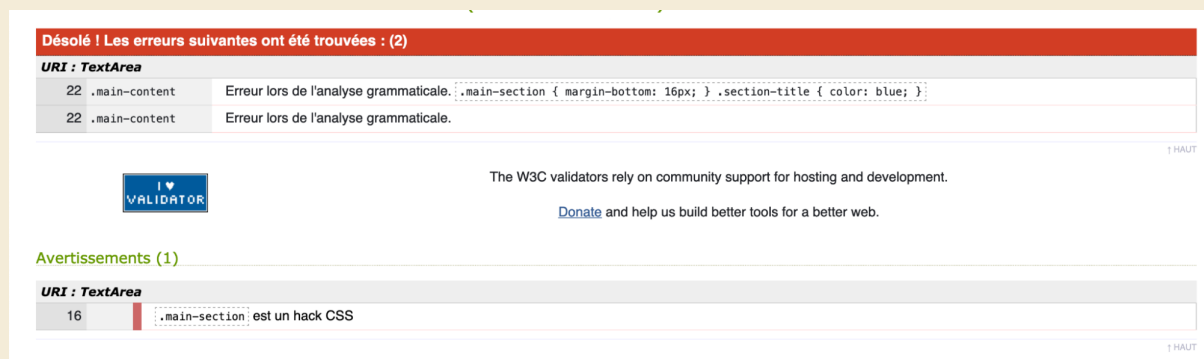
Le validateur HTML me montre que j'ai oublié de fermer la balise `header` dans mon code HTML.

Testons à présent ce bout de code dans le validateur CSS :

```
.main-wrapper {  
    background-color: #eee;  
    padding: 8px;  
}  
  
.main-header {  
    display: flex;  
    align-items: center;  
    justify-content: center;  
}  
  
.main-content {  
    .main-section {  
        margin-bottom: 16px;  
    }  
}  
  
.section-title {  
    color: blue;  
}
```

Il manque une accolade pour le sélecteur `.main-content` .

Regardons à présent le message d'erreur, que m'affiche le validateur CSS :



Le message d'erreur dans le validateur CSS

Comme vous pouvez le constater, le message d'erreur du validateur n'est pas très clair et ne nous donne pas beaucoup d'informations sur l'erreur : il ne m'indique pas clairement qu'une accolade est manquante.

Je vous invite dès à présent à coder quelques lignes de HTML et CSS, vous pouvez même prendre le code du projet Façadia et le modifier : essayez de mal fermer des balises, d'oublier des points virgules sur des propriétés CSS, d'employer des propriétés CSS inconnues et regardez comment le validateur se comporte.

Utiliser CanIUse

Parfois les validateurs ne vont pas vous remonter des erreurs, mais des **warnings** ! Et ces warnings peuvent être super intéressants pour analyser votre code et comprendre ce qui ne va pas.

Par exemple, peut-être que votre propriété CSS a besoin de “Vendor Prefix” et que vous avez oublié de le mettre dans votre projet. Ou alors, vous vous servez peut-être d'une propriété qui, soit n'est plus utilisée, soit est peu supportée par les navigateurs.

Dans ce cas-là, [le site CanIUse](#) est votre ami. Il va vous permettre d'analyser la propriété en détail : son pourcentage d'utilisation, les versions de navigateur supportées, etc.

À vous de jouer !



C'est l'heure de l'exercice et de mettre en application tout ce que vous avez vu dans ce chapitre ! Vous allez détecter un bug d'affichage.

Changez la branche de votre projet pour vous mettre sur la [partie-2/chapitre-1/section-3/exercice](#). Il semblerait que notre tableau contenant les données d'un capteur soit cassé et ne s'affiche pas comme il faut. Utilisez les outils que nous avons vu durant cette partie pour analyser le problème.

Dans la vidéo ci-dessous, je vais vous montrer comment je l'analyse.

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 boîte (Box Model).

Maintenant que vous connaissez les outils pour analyser les bugs, il est temps d'apprendre à les résoudre.

