

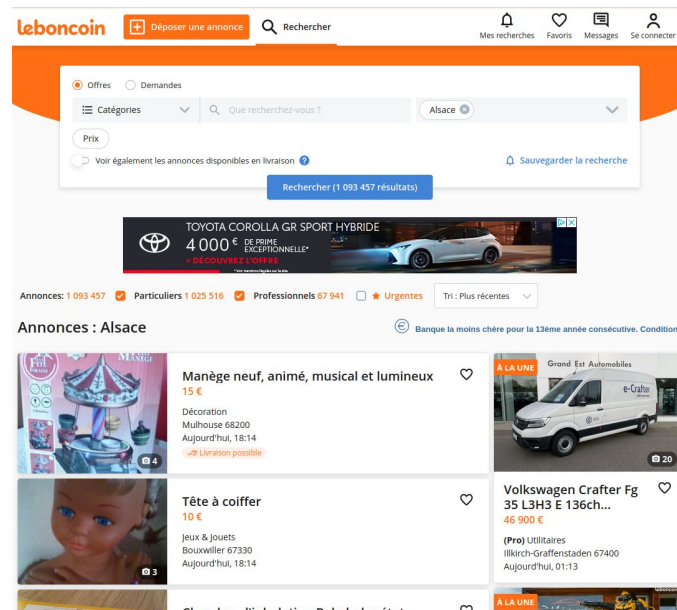
# HTML et CSS

Les petits conseils d'Hugo

A dark blue diagonal gradient bar that starts from the bottom left corner and extends towards the top right corner, covering the lower half of the page.

# Front end, back end?

- Front end : création de l'interface graphique (HTML, CSS, JS...)
- Back end : création de l'interaction entre les bases de données et l'utilisateur (PHP, MySQL...)



Site leboncoin.fr

# HTML

Langage permettant de structurer son site web

# HTML – Langage de balisage

- Un code HTML par type de page
- HTML se code avec des balises
- Les balises permettent de représenter un élément dans la page web
- On les retrouve facilement sur internet, en général on les apprend avec la pratique
- Quelques exemples : div, form, h1, a...

```
1 <div>
2   <h1>Titre stérile</h1>
3   <p>Paragraphe qui ne sert vraiment à rien suivi d'une image !</p>
4   
5 </div>
```

*Exemple d'un code HTML*

# HTML – Attributs

- Les attributs permettent de donner plus d'informations sur une balise
- Beaucoup d'attributs sont spécifiques à une balise en particulier, donc il est important de lire la doc des balises (par exemple les attributs src et alt pour la balise img)
- Deux attributs peuvent être utilisés par toutes les balises : l'identifiant (id) et la classe (class)
- L'identifiant ne doit désigner qu'un seul et unique objet dans une page web (IMPORTANT)
- La classe désigne un ensemble d'objets (mais cela peut être du coup un seul objet)
- Nous verrons leur utilisation plus tard

```
1 <div class="section">
2   <h1 id="sterile-title">Titre stérile</h1>
3   <p>Paragraphe qui ne sert vraiment à rien suivi d'une image !</p>
4   
5 </div>
```

*Exemple d'un code HTML avec attributs*

# HTML – Mes p'tits conseils

- Une page HTML doit être simple, épurée, rapide à comprendre
- Avant de démarrer, avoir une idée du rendu final (imagination, schéma, outil de visualisation en ligne...)
- Mettre des commentaires pour expliquer les parties les plus importantes (le top du top est de commenter votre code... avant de coder!)
- Astuce commentaires : TODO
- On ne donne d'identifiant et de classe à un objet (une balise) uniquement si c'est nécessaire (un code HTML seul ne doit contenir aucune classe et aucun identifiant, ils viendront avec le CSS et le JS)
- Les identifiants et les classes en kebab-case!
- Il existe la balise style : nous verrons dans la partie CSS pourquoi il ne faut JAMAIS l'utiliser
- Donner l'attribut alt aux images (svp, faites le, car tout le monde oublie), ainsi que leur dimensions
- Éviter une invasion de balises div (sinon c'est galère à corriger les bugs), faites au plus simple, même si les balises div restent des balises très puissantes
- Pas nécessaire de tester visuellement le HTML dans votre navigateur, plus important de vous concentrer uniquement sur votre code : on ne test visuellement que le CSS et le JS

# CSS

Langage permettant de mettre en page  
et rendre joli votre site web

# CSS – Langage de sélecteurs

- Le principe est simple : on va sélectionner des “éléments” par leur type de balise, identifiant ou classe, pour leur affecter/modifier de nouvelles propriétés de style
- Les sélections précises sont prioritaires sur les sélections plus générale en cas de conflit
- Pour sélectionner un type de balise, on écrit le type de la balise directement (ex : `img {...}`)
- Pour sélectionner un identifiant, on écrit `#` suivi de l’identifiant (ex : `#sterile-title {...}`)
- Pour sélectionner une classe, on écrit `.` suivi du nom de la classe (ex : `.section {...}`)
- Pour sélectionner une sélection dans une autre, on les écrit à la suite (ex : `.section img {...}`)
- Pour faire de la multi-sélection, on met une virgule (ex `#sterile-title, p {...}`)
- On peut accéder à des pseudo-classes avec `:`, il est recommandé de lire la doc (ex : `p:hover {...}`)

```
1 #sterile-title,  
2 p {  
3     font-weight: bold;  
4 }  
5  
6 .section {  
7     color: black;  
8 }  
9  
10 #sterile-title {  
11     color: red;  
12 }
```

Exemple d'un code CSS



# CSS – Comment kon s'organise?

- De manière à garder un code HTML pur et léger, on ne va jamais utiliser l'attribut style, ni de balise style (c'est une convention, alors par Jupiter, respectez là) : on utilisera que des feuilles CSS
- On utilise une feuille CSS qu'on nomme par convention main.css, qui va contenir tous les éléments de style que l'on peut retrouver sur tout le site web
- On utilise une feuille CSS par page HTML créée
- Lorsqu'une page CSS commence à devenir trop importante, on la découpe en plusieurs feuilles selon des catégories
- Voilà ce qui arrive quand on ne s'organise pas : notre projet pour le site web d'ITS

# CSS – Mes p'tits conseils

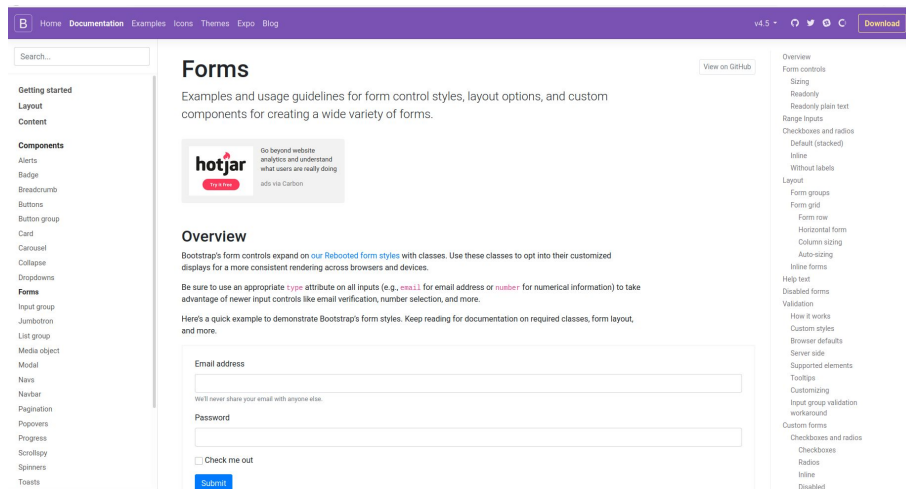
- Faire simple, épuré : ne pas chercher à mettre trop d'éléments de style
- IMPORTANT : on va privilégier en CSS les classes par rapport aux identifiants dès que possible
- On sélectionne un id uniquement dans la feuille CSS d'une page HTML et si on est archi sûr que l'élément est unique
- On met des commentaires (faites moi confiance, c'est vraiment utile)
- On précise dans les commentaires si le sélecteur en question prend son sens avec d'autres sélecteurs
- On évite de créer des modifications de style qui se répètent
- On crée plusieurs classes pour pouvoir généraliser des styles : btn, btn-rounded, magic...
- On fait le CSS toujours après le HTML, pas pendant!
- Il est très conseillé de vérifier visuellement, à chaque ajout de style, le rendu sur le navigateur
- On va toujours préférer le relatif à l'absolu, le pourcentage aux pixels
- Utilisez flex et grid : c'est très utile !
- On fait en permanence du responsive design : il faut que notre CSS ait un bon rendu sur tous les navigateurs et les appareils (on peut utiliser les Media Queries)
- Outil pour déboguer le CSS : inspecter l'élément
- On apprend beaucoup avec la pratique !

# Bootstrap

Un framework de CSS : très joli sans se prendre la tête

# Bootstrap – Du CSS déjà écrit

- Bootstrap est un énorme fichier CSS, codé uniquement avec des sélecteurs de classe
- On peut faire pleins de trucs très utiles : des formulaires, des navbars, des carrousels...
- Pas besoin de les apprendre par coeur, il faut juste aller sur leur site officiel, et adapter!



The screenshot shows the Bootstrap 4.5 documentation page for the 'Forms' section. The page has a purple header with navigation links: Home, Documentation, Examples, Icons, Themes, Expo, Blog, and a version indicator 'v4.5' with a 'Download' button. A search bar is located below the header. On the left, there is a sidebar menu with categories like 'Getting started', 'Layout', 'Content', 'Components', and 'Forms'. The main content area is titled 'Forms' and includes an introductory paragraph, an advertisement for Hotjar, an 'Overview' section explaining Bootstrap's form controls, and a form example with fields for 'Email address' and 'Password', a 'Check me out' checkbox, and a 'Submit' button. On the right, there is a table of contents for the 'Forms' section, listing various form controls and their sub-features.

# Bootstrap – Mes p'tits conseils

- Si le CSS se fait après le HTML, quand on fait du Bootstrap, il est conseillé de le faire pendant le codage du HTML
- Comme le CSS, il faut faire des contrôles visuels réguliers
- Important : même si c'est possible, on ne modifie pas directement les classes de Bootstrap dans le code CSS
- Inspirez vous salement de leurs templates sur leur site officiel, et adaptez le à votre site (beaucoup plus simple, c'est fait pour ça)
- Dans le CSS, quand on crée un sélecteur de style qui a du sens uniquement avec des classes Bootstrap, on le précise dans les commentaires

# Conseils généraux

Les points les plus importants à retenir!

# Les points les plus importants à retenir

- Des commentaires, s'il vous plaît!
- Des pages HTML pures, claires et simples
- On privilégie des classes aux identifiants pour le CSS (sauf cas précis)
- On structure les feuilles CSS pour éviter des pages CSS gigantesques
- On utilise Bootstrap pour se simplifier la vie
- Cherchez pas à faire des fioritures, pas trop d'éléments graphiques
- On ne met que des images et des icônes libres de droits
- On communique bien avec son équipe pour éviter des problèmes de convention
- Le web, c'est avant tout de la pratique!

# Vos questions?

Puis après on passe à la pratique sur un site de poti-animaux ;)