- Rapport de projet TP 1 -

- Février 2023 -

Site vitrine / infos



DE Albérick Nicolas

Sébastien

AFCI LIEVIN DW WM

**Fait par Sébastien**

Formateur : Antoine VERRIN

Table des matières

[**1.Description du projet** 4](#_Toc127193145)

[**2.1. Exigences fonctionnelles** 4](#_Toc127193146)

[**2.2.Exigences non fonctionnelles** 5](#_Toc127193147)

[**2.3 Répartition des taches** 5](#_Toc127193148)

[**2. Analyse et conception** 6](#_Toc127193149)

[**2.1 La conception fonctionnelle** 6](#_Toc127193150)

[**2.2 La conception graphique** 7](#_Toc127193151)

[**2.3 Les langages utilisés** 9](#_Toc127193152)

[**2.4 Les tests** 9](#_Toc127193153)

[**3.** **Création des pages événements** 10](#_Toc127193154)

[**3.1 Présentation du code HTML** 10](#_Toc127193155)

[**3.2 Présentation du code CSS** 11](#_Toc127193156)

[**3.3 Présentation du CSS responsive** 14](#_Toc127193157)

[**4. Création de la page Articles** 17](#_Toc127193158)

[**4.1 Présentation du code HTML** 18](#_Toc127193159)

[**4.2 Présentation du code CSS** 19](#_Toc127193160)

[**4.3 Présentation de CSS responsive** 22](#_Toc127193161)

[**5**. **Conclusion** 24](#_Toc127193162)

**Introduction**

Nous avons donc choisi pour notre projet tutoré, de développer un site parodique local sur la ville de Liévin.

L'objectif est de développer un site Internet faussement journalistique pour faire rire les internautes, en mettant à leurs dispositions des outils de base pour partager des informations humoristiques.

Néanmoins nous suivons certaines caractéristiques d’un journal classique.

Pour mener à bien ce projet, nous avons tout d'abord listé et analysé les exigences requises, afin d’obtenir une expression précise des besoins.

Cette analyse a permis de développer plus efficacement les différentes fonctionnalités. Ensuite, nous avons effectué des tests dans le but de comparer les exigences attendues aux résultats obtenus et d'améliorer ces derniers.

Et enfin, nous avons procédé à un bilan de notre projet.

# **Description du projet**

Puisqu’il s’agit d’un site d’infos parodiques, nous avons choisi pour le nom du site : « Liév’intox ». Nous pensons qu’il illustre immédiatement le thème du site et qu’il représente bien le penchant des internautes, gourmands, pour ce genre de lecture.

Liév’intox devra donc regrouper toutes les fonctionnalités nécessaires au partage des articles humoristiques telles que : des articles du jour, des évènements, des archives, recevoir des emails.

Mais aussi, il devra répondre à des exigences non fonctionnelles par sa qualité et ses performances.

## **2.1. Exigences fonctionnelles**

* + - Accueil

Comme tout site web, le Liév’intox devra avoir sa page d’accueil avec un visuel sur les articles, événements et une ergonomie d’utilisation.

* + - Menu

Le menu est mis en évidence afin d’accéder aux différentes pages et pouvoir accéder aux différentes fonctionnalités facilement.

* + - Evénement

La page événement contient 2 niveaux, les faux évènements à venir et ceux passés.

* + - Articles

Les articles sont rédigés uniquement dans un but à distraire et faire rire le visiteur.

* + - Contact

Un visiteur a la possibilité d’envoyer un message aux administrateurs par le biais du formulaire contact.

* Widget calendrier

A la demande du client, nous avons ajouté un widget « google agenda », ainsi l’internaute aura accès à un calendrier dans le pied de page du site.

* + - Widget météo

Le client souhaite aussi un widget météo, il est localisé sur Liévin.

* + - Visibilité

Les réseaux sociaux du journal sont proposés en pied de page, ainsi le site aura plus de visibilité par les partages des internautes et cela participera aussi au référencement off page.

## **2.2.Exigences non fonctionnelles**

* Exigences de qualité

Afin que le site donne envie aux internautes de le faire connaître à leurs proches et de lui être fidèle, il est important de répondre aux exigences de qualité suivantes :

* Ergonomie efficace

Partager de fausses informations ok, mais le faire avec qualité, la visite doit être un plaisir. La mise en page du site doit faciliter un maximum la démarche à l’aide d’une présentation claire et intuitive. Nous reviendrons sur ce facteur de qualité dans la partie conception.

* Interface graphique

Les différentes couleurs et choix typographiques doivent permettre à un utilisateur de repérer les différentes fonctionnalités qui s’offrent à lui.

* Contact

Pour toutes questions ou informations, l’utilisateur doit pouvoir contacter le webmaster.

* + - Exigences de performance

Il faut aussi prendre en compte les exigences quantitatives

## **2.3 Répartition des taches**

Lors de notre réunion en début de projet, nous avons établi une répartition des taches pour chacun d’entre nous.

Le maquettage et la création de la page Accueil comprenant la barre de navigation et le pied de page pour Albérick.

La création des différents fichiers HTML et CSS, l’écriture de la présentation du projet et la création de la page Contact pour Nicolas.

Enfin la création des pages Evenements et de la page Articles pour moi.

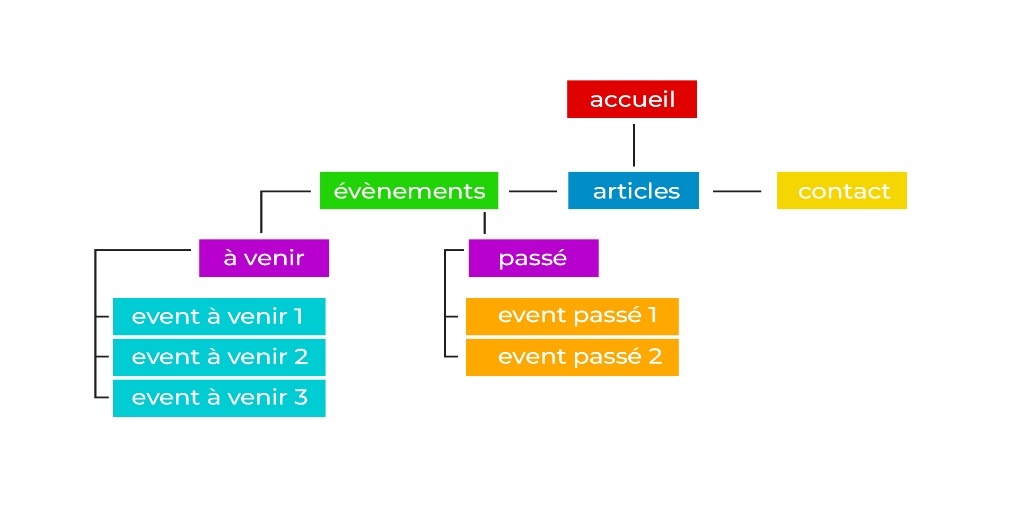
# **2. Analyse et conception**

## **2.1 La conception fonctionnelle**

* La structure du site

Le site web est basé sur une structure simple. Un accueil et des pages articles et évènements. Aucune identification ou inscription n’est demandée, il y a juste à naviguer sur le site et lire.

Ces pages sont présentées ainsi.



Nous avons donc choisi ce type de schéma d'organisation puisqu'il s'adapte facilement à nos objectifs, qui sont d'une part, une facilité de navigation pour l'utilisateur en se familiarisant très rapidement à ce schéma, et d'autre part une organisation soignée de nos informations et fonctionnalités.

* La structure des pages

Toutes les pages du site Liév’intox ont une même structure.



## **2.2 La conception graphique**

L’ergonomie est un point important. Son objectif principal est de faciliter l’utilisation du site par les utilisateurs. Pour cela, nous nous imposerons certaines contraintes :

* Accessibilité des navigateurs

Le site Liév’intox doit être fonctionnel sous les navigateurs les plus utilisés, c’est-à-dire : Chrome, Firefox, Opéra et Brave. Les différentes fonctionnalités développées devront donc être testées sur les quatre navigateurs.

* Le système de navigation

La navigation doit être simple, même pour les usagers débutants. Il est donc indispensable que l’apprentissage du site ne demande pas des efforts à l’utilisateur. L’utilisation de menu qui répertorie les principales fonctionnalités permet à l‘utilisateur de minimiser son effort mental.

La barre de fichiers, qui est un menu horizontal statique est donc composé de :

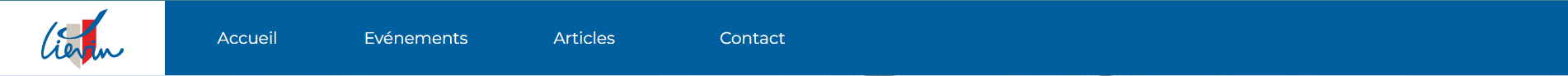
- accueil

- évènements

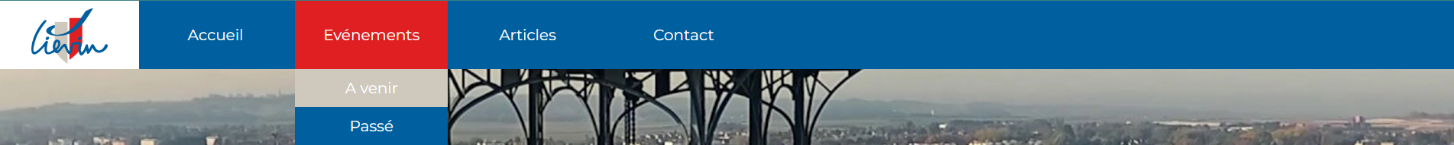
- articles : derniers articles créés

- contact

Barre de menu.



Voici ce qu’il se présente en naviguant sur le menu en allant jusqu’au sous-menu.



Les couleurs de la chartre graphique y sont respectées

Apparence du site

* La charte graphique

Afin de donner à l’utilisateur des repères au sein du site et de faciliter sa visite, il doit y avoir une cohérence entre chaque page. La charte graphique détermine les différentes règles graphiques et ergonomiques.

Elle contient :

* Les dimensions des pages
* Les couleurs à employer (le texte, les liens, les liens actifs, les boutons…)
* Les styles à employer (famille, taille…)
* Les types d’images…

* + Choix des couleurs

Pour réaliser l’image du nom « Liév’intox », nous avons repris le code couleur du logo de la ville Liévin.

Avec le plug-in chrome SMART COLOR PICKER, nous avons pu avec une pipette numérique connaitre le code couleur des 4 couleurs principales.



La police google Montserrat a été choisie

* + Contenu

Chaque page doit avoir un titre propre permettant à l’utilisateur de savoir sans réfléchir dans quelle section du site il se trouve.

Les phrases simples et les textes courts sont privilégiés car cela favorise une lecture rapide pour l’utilisateur.

## **2.3 Les langages utilisés**

* HTML

Nous avons utilisé le HTML.

* CSS

Afin de manipuler la présentation, nous avons utilisé une feuille de style CSS.

Voici les raisons : D’une part, il permet d’alléger le code source écrit en HTML, puisque tout ce qui est relatif à la présentation est géré dans un fichier séparé. Ce qui entraîne donc un chargement plus rapide des pages, qui est après manipulé par la feuille de style.

Et d’autre part, il permet de nous retrouver plus facilement dans notre code et ainsi facilite les modifications à effectuer, puisqu’au lieu d’avoir à modifier toutes les pages unes à unes, nous avons juste à modifier le fichier CSS.

Exemples d’utilisation :

- Tous les choix graphiques

- Pages différentes lors que l’utilisateur s’est authentifié ou non.

## **2.4 Les tests**

Afin que le site soit de qualité, il est important de réaliser des tests. Ces tests ont pour but de détecter les erreurs éventuelles et permettent de les corriger pour qu’il n’y ait pas d’impact sur le fonctionnement du site.

Pour cela au cours du développement, nous avons effectué 4 types de tests :

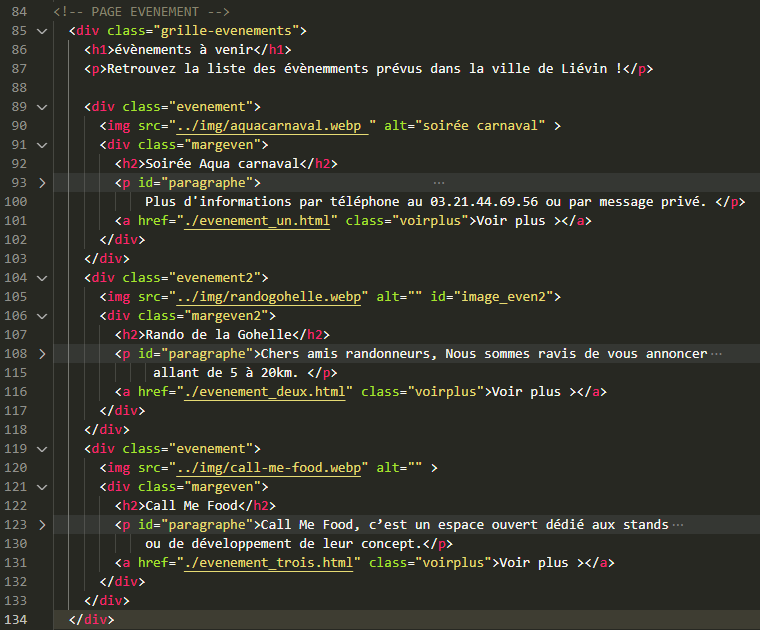
* Des tests unitaires
* Des tests fonctionnels
* Des tests d’utilisabilité
* Des tests responsifs

# **3.** **Création des pages événements**

Voici la maquette prévue pour nos pages évènements



## **3.1 Présentation du code HTML**



J’ai commencé par créer un conteneur dans une balise « div » ayant pour attribut class et de valeur *grille-evenements* » pour y placer les différents articles.

Ensuite j’ai créé le titre dans une balise <h1> afin d’avoir le titre de la page et un paragraphe dans la balise <p> pour mettre une description de la page.

Chaque article est contenu dans une balise « div class=*evenement* » ou « div class=*evenement2* » pour le CCS ensuite.

Les images utilisées sont en format WEBP pour permettre un chargement plus rapide des pages car il est plus léger que les formats JPEG.

La composition de chacune de ses div est la même à savoir l’image de l’article contenu dans la balise <img> suivi d’un conteneur *margeven* dans lequel est placé le titre de l’article dans une balise <h2> et le texte qui possède l’*id=paragraphe* et contient un descriptif de l’article. La fin de ses div se terminent par un bouton qui renvoi vers la page dédiée à l’article grâce à la balise <a href= >

## **3.2 Présentation du code CSS**

Commençons par le conteneur « div class=*grille-evenements* »



Tout d’abord on remarque que les sélecteurs sont précédés soit d’un point qui renvoie à une class du fichier HMTL soit d’un # qui lui renvoi à une id.

Je donne une valeur sur la largeur de 1540 pixels grâce à la propriété width au conteneur des articles avec *grille-evenements* { width : 1540 px}

J’applique ensuite une marge auto de 5% à cet élément avec le margin : 5% auto.

Le *grille-evenements h1* contient un font-size : 58px et une text-transform : uppercase qui permet de donner aux lettres contenues dans le <h1> d’avoir une taille de 58pixels ainsi qu’une écriture en majuscule.

Le *grille-evenements p* contient un font-size : 25px ainsi qu’un margin-top qui permet un décalage de -10px par rapport au titre <h1>.

Le *grille-evenement h2* permet de donne aux titres des articles une taille de 45px et une marge inférieure de 15px.

Enfin le texte des articles qui sont contenu dans l’*id* *paragraphe* aura une taille de 23 px ainsi qu’une marge inférieure de 20px.



Les *evenement* sont tous les deux un display : flex qui permet de faire en sorte que les éléments à l’intérieur de ces balises soient en ligne plutôt qu’en colonne à la suite.

Le margin-top : 40px permet d’avoir une séparation en haut de 40 px.

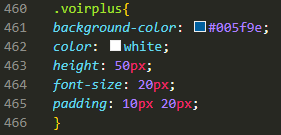
On remarquera le flex-direction : row-reverse dans le *evenement2* qui permet d’inverser l’ordre des éléments pour avoir un décalage entre les éléments compris dans le .*evenement*.



Le margin-left et le margin-right contenu dans les *margeven* et *margeven2* permet d’avoir une séparation de 20px entre le texte et l’image du conteneur.



On termine le CSS des pages événements avec la partie du CSS consacrée au bouton



Le background-color permet de donner la couleur bleue comprise dans la chartre graphique préétablie par le client.

Le color : white permet de faire apparaitre les lettres du texte bouton en blanche et le font-size : 20px de leurs donner une hauteur de 20px.

La height : 50px donne au bouton une hauteur de 50px.

Le padding : 10px 20px permet d’élargir le bouton de 10px en haut et en bas et de 20px à gauche et à droite.

## **3.3 Présentation du CSS responsive**



Tout d’abord cette ligne nous indique que pour les écrans ayant une résolution inférieure à 420px, toutes les lignes de code de CSS qui suivront seront celles prises en compte pour l’affichage de notre page.

Voici la maquette prévue pour nos pages évènements



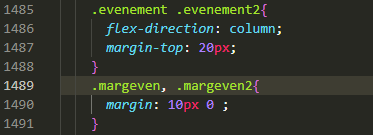
**

Ce qui change par rapport aux lignes de CSS de l’affichage sur écran :

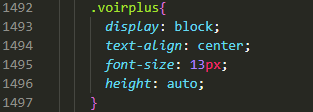
La largeur du conteneur *grille-evenements* passe de 1540px à 400px afin de correspondre à la largeur maximale définie dans notre @media screen and (max-width : 420 px).

Cette fois on met une marge automatique pour mieux s’adapter et un padding de 3%.

On modifie la taille des polices des différentes parties afin qu’elles s’intègrent mieux au format mobile.



Pour l’orientation des différents articles j’utilise la propriété flex-direction : column afin qu’ils se présente tous en colonne.



Pour les boutons on réduit la police à 13px.

Le display : block transforme le bouton en élément de bloc. Il s'insère alors dans le contenu comme une section distincte, avec un saut de ligne avant et après.

Le text-align permet d’avoir le texte centrer au milieu du bouton.

La height : auto permet au bouton de s’adapter son contenu.

# **4. Création de la page Articles**

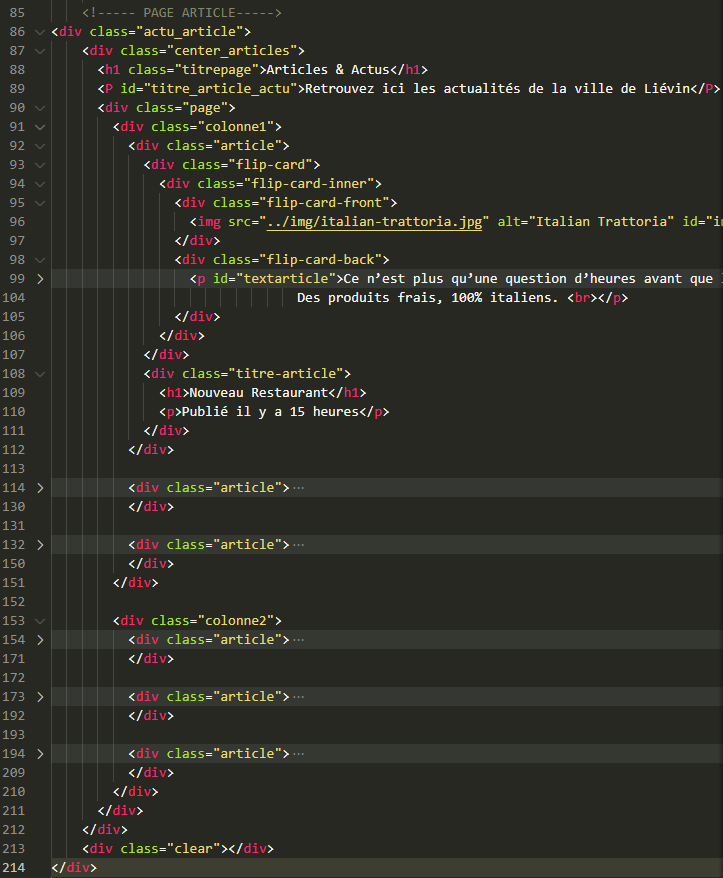
Voici la maquette prévue pour notre page Articles



Pour cette page, j’ai utilisé un système de flip-card qui montre l’image de l’article et qui se retourne au survol de la souris laissant apparaître la description de l’article.

Cette présentation est faite pour donner du dynamisme au site.

## **4.1 Présentation du code HTML**



Tous les éléments seront compris dans une grande div ayant pour class *actu\_article*.

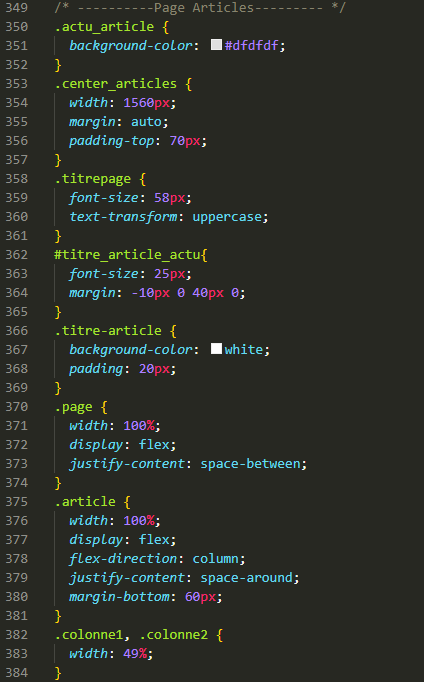
Dans ce dernier j’ajoute un second conteneur *center\_articles* qui me servira à centrer les éléments plus tard. Il contient le titre de la page ainsi que les différents articles contenus dans la div *page*.

Les articles sont répartis dans 2 conteneurs *colonne1* et *colonne2*.

Chaque article est présenté sous forme d’une flip-card :

* L’image de l’article est placée à l’avant de la flip-card dans la *flip-card-front* .
* Le texte descriptif de l’article est à l’arrière dans la *flip-card-back.*

## **4.2 Présentation du code CSS**



Le sélecteur *.actu\_article* me permet d’afficher une couleur d’arrière-plan grâce à la propriété background-color de couleur grise issue de la chartre graphique.

Le sélecteur *.center\_articles* permet au titre de la page et aux articles d’être réparti dans une « fenêtre » de largeur 1560px grâce aux propriétés width : 1560px et margin : auto. Le padding-top sert à garder une marge supérieure de 70px afin d’aérer la page.

Le sélecteur *.titrepage* donne une hauteur au titre de 58px et une écriture en majuscule avec la valeur uppercase de la propriété text-transform.

Le sélecteur *#titre\_article\_actu* donne une hauteur au texte de 25px. J’utilise des valeurs -10px 0 40 px 0 sur le margin pour aligner le texte descriptif du titre sur le titre.

Le sélecteur *.titre-article* permet de donne un fond blanc au titre des différents articles ainsi qu’un espacement de 20px.

Le sélecteur *.page* permet uniquement de contenir les deux .*colonne1* et .*colonne2*.

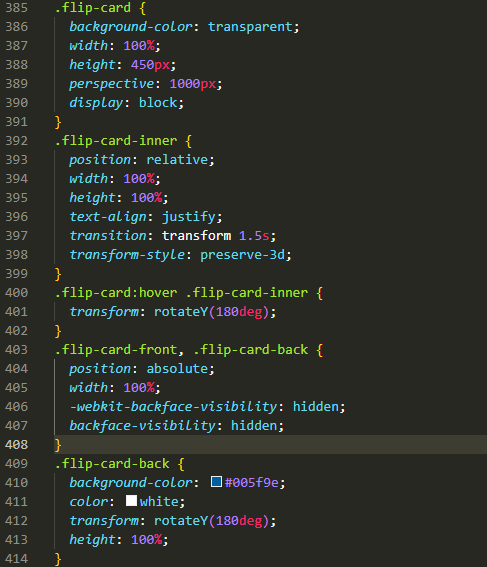
Je lui donne une largeur de 100% avec la propriété width et une disposition flex pour y placer les articles après.

Les *.colonne1* et *.colonne2* ont une largeur de 49% afin de laisser une séparation de 2% entre les deux.

Les *.article* compris dans les div *colonne* auront une largeur de 100% de la colonne, avec une disposition en flex en colonne donnée avec les propriétés display :flex et flex-direction :column.

Le margin-bottom permet d’avoir un espace sous chaque article afin d’aérer la page.

Voyons la mise en place du CSS utilisée pour les flip-card.



Tout d’abord le sélecteur *.flip-card* a un arrière-plan transparent avec la propriété background-color :transparent. On lui donne une largeur de 100% de la colonne grâce à la valeur 100% de la propriété width et une taille de 450px avec la propriété height.

La propriété perspective :1000px est combinée avec la propriété transforme-style :preserve-3d du sélecteur *.flip-card-inner* pour créer un effet de transformation. C’est la propriété transform :rotateY(180deg) qui donne l’effet de rotation horizontale.

L’image de l’article est contenue dans le sélecteur *.flip-card-front*.

Le texte de l’article est contenu dans le sélecteur *.flip-card-back*, il a une taille de police de 23px de couleur blanche et une marge interne de 20px pour laisser un peu de marge. Il apparait avec un arrière-plan bleu issu de la chartre graphique.

La rotation entre l’image et le texte se fait au passage du pointeur souris grâce au *.flip-card :hover* qui déclenche la rotation.

La face cachée de la flip-card l’est grâce à propriété backface-visibility :hidden.

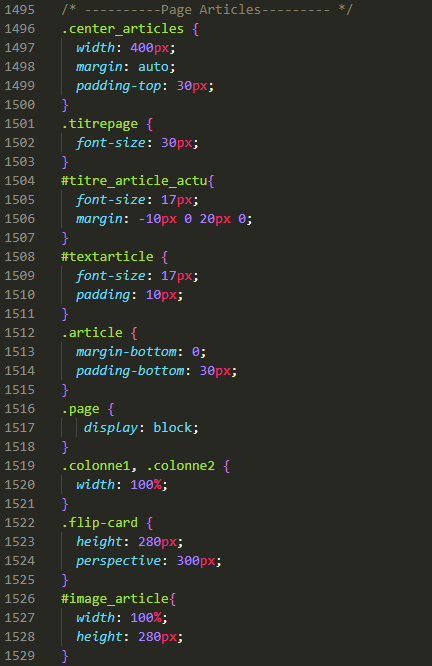
## **4.3 Présentation de CSS responsive**



Tout d’abord cette ligne nous indique que pour les écrans ayant une résolution inférieure à 420px, toutes les lignes de code de CSS qui suivront seront celles prises en compte pour l’affichage de notre page.

Voici la maquette prévue pour notre page articles





On modifie la valeur de largeur à 400pixels dans le sélecteur *.center\_articles* afin de s’intégrer aux 420pixels définie dans notre point de rupture et ainsi garder une marge de 10pixels à droite et à gauche.

Le padding-top nous permet d’avoir d’aérer la page.

Cette fois les articles seront présentés en forme d’une seule colonne avec le changement de display :flex à display :block dans le sélecteur *.page.*

La taille des éléments compris dans les *flip-card* sont réduit à une hauteur de 280pixels.

# **5**. **Conclusion**

Ce premier projet de groupe est un site Web d’un journal de Liévin, un site vitrine / infos. Il permet de mettre en relation plusieurs enseignements utilisés sous une dimension professionnelle.

Tout d’abord, ce projet nous a permis d'appliquer les connaissances que nous avons acquises durant ces deux premières semaines mais aussi nos connaissances personnelles, telles que la modélisation, les différents langages de programmation, la gestion de projet mais aussi la communication.

Grâce à celui-ci chaque membre de l'équipe a pu renforcer ses connaissances mais aussi apporter aux autres membres son savoir et ses compétences afin d'harmoniser l'efficacité de l'équipe.

A travers des méthodes de travail et des outils, ce projet nous a permis de nous immerger dans un univers professionnel.

Il est vrai que de créer un site et respecter un cahier des charges rendent un projet intéressant et professionnel mais il y a aussi toutes les démarches qui ne sont pas visibles et qui rendent enrichissante une telle expérience : écouter l’opinion de chacun des membres de l'équipe, savoir communiquer et argumenter afin d’opter pour les meilleurs choix, s’organiser sur les plans personnels et collectifs, gérer les imprévus, respecter des délais pour ne pas gêner ses collègues et pour ne pas retarder tout le projet.

Le projet nous apporte donc à chacun une idée sur l’organisation dans le monde professionnel et qui permettra de nous adapter plus facilement lors de notre stage.