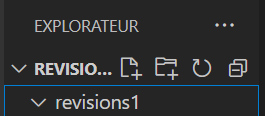
Révision – Balises de bases

Cet exercice de révision concerne la structure de base d’une page HTML et les balises (ou élément) html utilisée fréquemment en HTML.

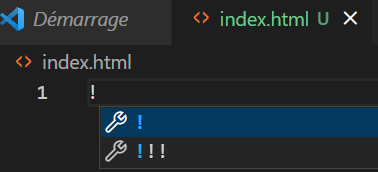
Pour commencer ces révisions, vous pouvez créer un répertoire revisions-html dans lequel vous mettez votre projet revisions1

Voici la structure à obtenir dans Visual Code :



|  |  |
| --- | --- |
| Une fois les dossiers créés et correctement ouverts dans Visual Code (via le menu déroulant Fichier / Ouvrir un Dossier), vous pouvez créer un fichier index.html.  Faites bien attention à le créer dans le répertoire revisions1. |  |

Une fois ouvert, tapez « ! » pour insérer le code HTML de base.



Vous devez obtenir le code suivant.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

</body>

</html>

Quelle est l’utilité de la balise suivante ?

<!DOCTYPE html>

Quelle est l’utilité de la balise suivante ?

<html lang="en">

Quelle est l’utilité de la balise suivante ?

<head>

Quelle est l’utilité de la balise suivante ?

<meta charset="UTF-8">

Quelle est l’utilité de la balise suivante ?

<meta name="viewport" content="width=device-width, initial-scale=1.0">

Quelle est l’utilité de la balise suivante ?

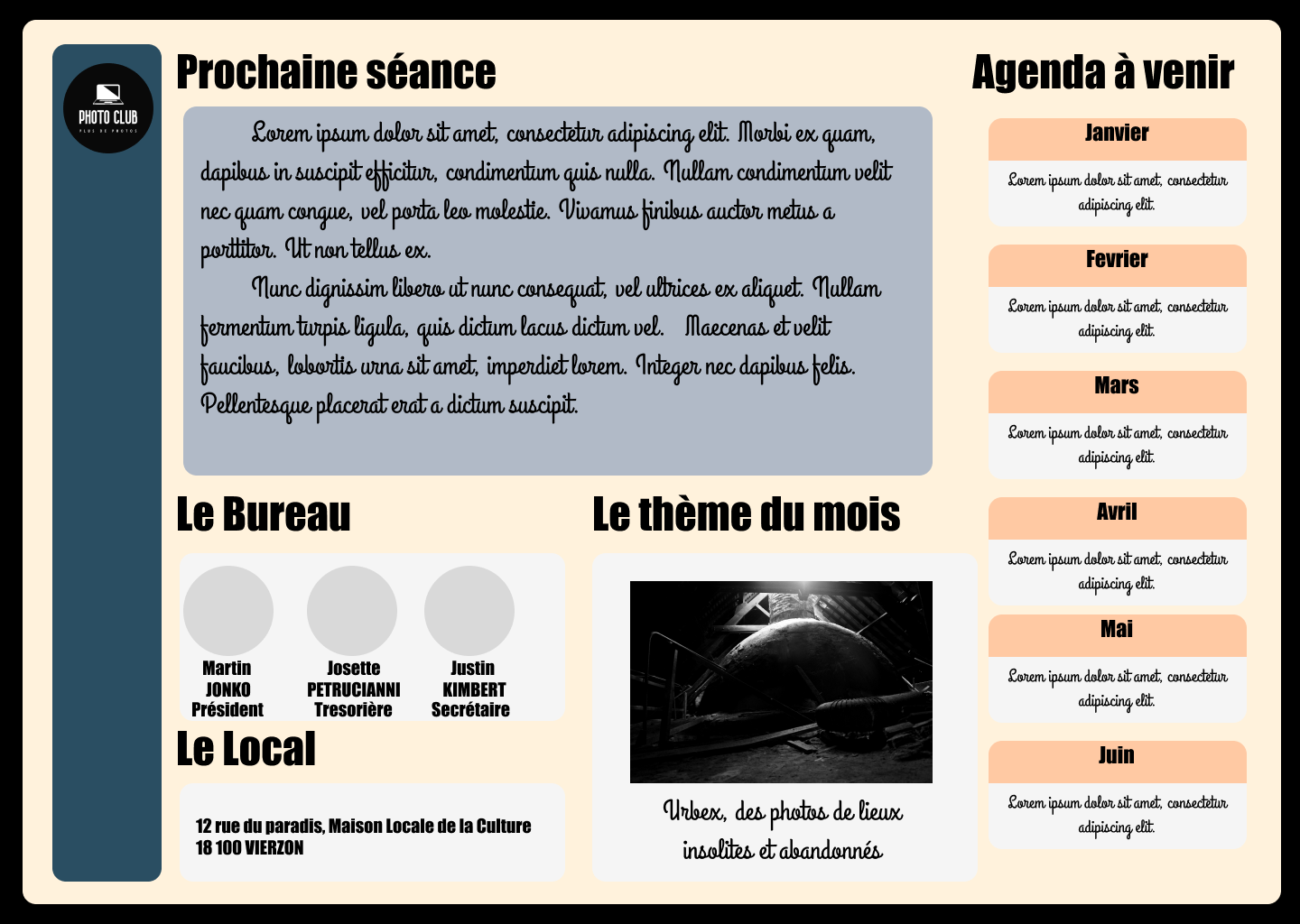
<body>

Projet à développer

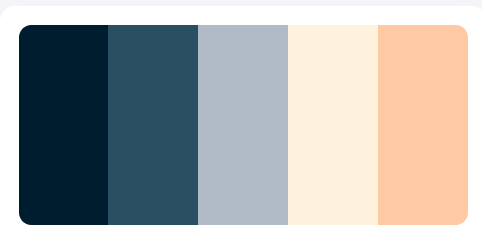
Pour support de ces révisions vous allez devoir reproduire une maquette FIGMA. Vous serez guidé tout au long du processus. Joint avec le sujet, vous aurez les fichiers sources (images, logo et autres).

Dans le répertoire révisions-html créez le projet photo-club. N’oublie pas de pousser le tout sur GitHub !

Voici la maquette :



Palette de couleurs :



#001C2F noir

#2A4E62 bleu gris

#B1BAC7 gris

#FFF2DC saumon clair

#FFC9A3 saumon foncé

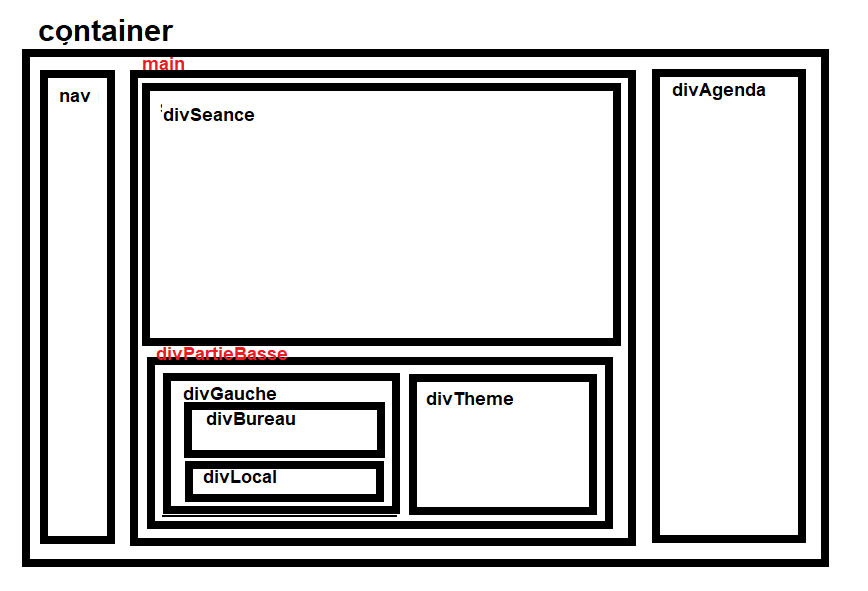
# Découpage de la page en DIV

Vous le savez, tout est DIV (HEADER,MAIN,FOOTER,NAV et autres). Il s’agit de conteneurs qui prennent toute la largeur de la page et s’ajuste en hauteur au contenu. Avec FLEX, vous avez remarqué que vous pouviez changer ce comportement.

Malgré cela, cela reste des boites rectangulaires ! Et il faut penser les alignements

## Découpage de la page

Voici le pré-découpage sommaire de la page en général. Nous avons créé les premières DIV. Nous allons pouvoir les positionner afin d’obtenir ce visuel.



## Mise en place HTML

Il nous faut mettre l’arborescence des divs qui correspond. Pour le moment nous ne nous préoccupons pas des attributs de mise en forme

Voyons le code HTML :

<body>

    <div id="container">

        <nav>nav</nav>

        <main>

            <div id="divSeance">Séance</div>

            <div id="divPartieBasse">

                <div id="divGauche">

                    <div id="divBureau"> bureau</div>

                    <div id="divLocal">local</div>

                </div>

                <div id="divTheme">theme</div>

            </div>

        </main>

        <div id="divAgenda">agenda</div>

</div>

</body>

N’oubliez pas que vous pouvez taper div#divSeance et Visual Code tape automatiquement le div avec l’identifiant choisi !

Ajoutons un fichier CSS lié (raccourci : link :csss )

<link rel="stylesheet" href="style.css">

N’oubliez pas ! cette ligne se met bien dans HEAD (à ne pas confondre avec HEADER)

Ajoutons une bordure épaisse à tous nos div !

div

{

    border: 3px solid black;

}

nav

{

    border: 3px solid black;

}

main

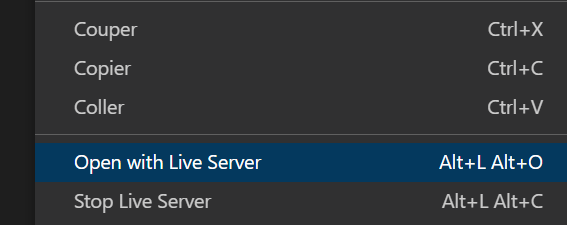
{

    border: 3px solid black;

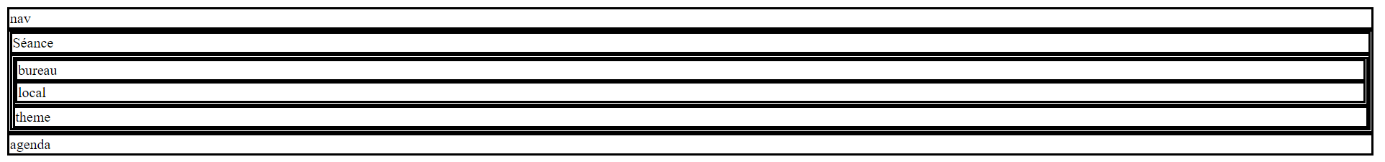
}

## Tester !

Click droit dans la page de code HTML, open with live server ! Cette manipulation permet de ne jamais oublier de mettre à jour la page dans le navigateur test !



Le résultat ? Moche !



# Mise en place des éléments

Il va être temps de mettre du flex ! Mais quels divs sont à faire flexer ? Rappelez-vous on ajoute l’attribut display :flex au PARENT pour que les enfants s’ajustent à la largeur disponible

Les parents nominés sont :

* Container
* Main
* divPartieBasse
* divGauche

Il ne faut pas oublier que Main et divGauche seront alignée en colonne !

Il est entendu que lorsque l’on mettra le détails du contenu, par exemple, pour l’agenda, d’autres divs seront ajouté et flexeront !

Voici les petites modifications apportées au CSS :

main {

    border: 3px solid black;

    display: flex;

    flex-direction: column;}

#container {

    display: flex;}

#divPartieGauche {

    display: flex;

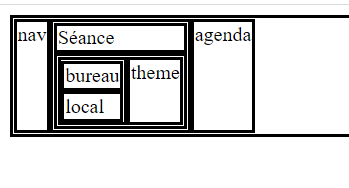
    flex-direction: column;}

#divPartieBasse {

    display: flex;}

Mais qu’obtient-t-on ? Page suivante !

Et voila ! TADAM !



# Dimensionnement des éléments

Il faut donner au container la hauteur maximale (vous vous rappelez du min-height :100vh ?) mais également donner 20% à nav, 30% à agenda et le reste pour le bloc du milieu !

container

{

    min-height: 100vh;

}

Pour ce faire : On peut utiliser flex-grow

nav {

    border: 3px solid black;

    flex-grow: 1;

}

main {

    border: 3px solid black;

    display: flex;

    flex-direction: column;

    flex-grow: 10;

}

#divAgenda

{

    flex-grow: 2;

}

Après il faut mettre flex-grow :1 sur les divs suivant :

divPartieGauche

#divPartieBasse

#divSeance

#divTheme

#divBureau

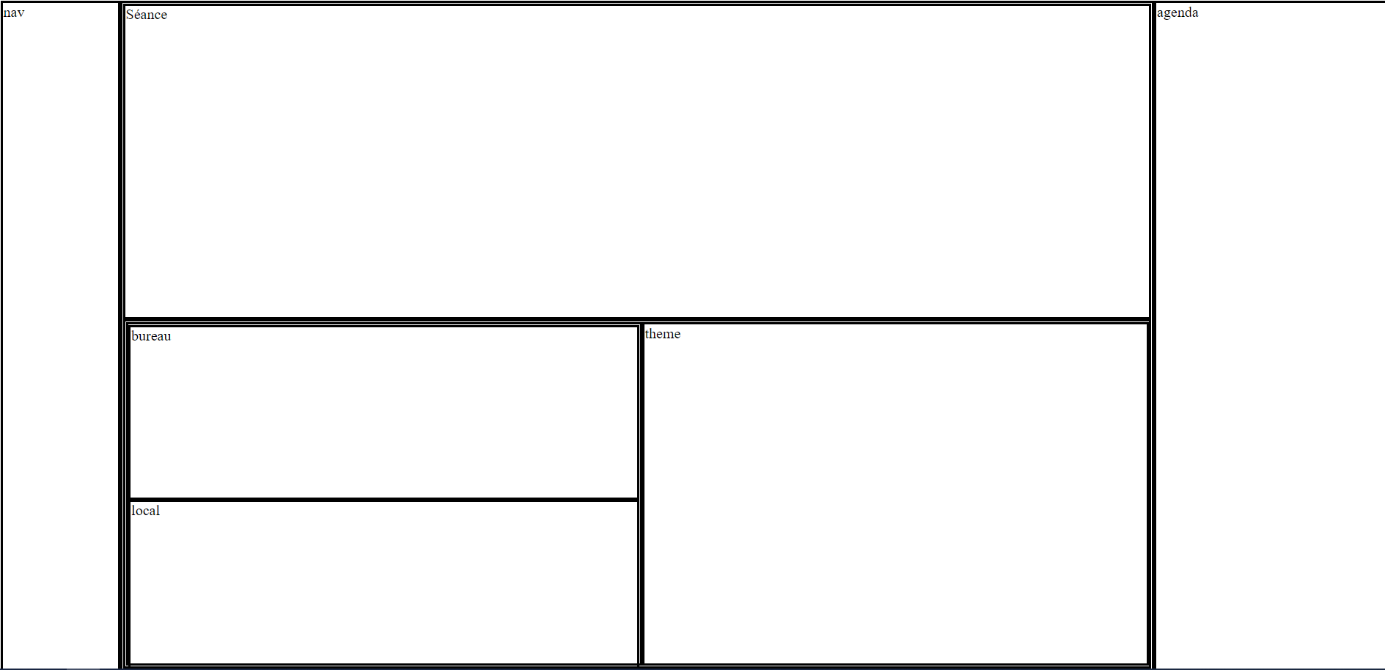
#divLocal

#divGauche

Il faut également fixer la hauteur de la divBureau et de la divLocal :

height: 50%;

Qu’avons-nous ? Quasiment ce que l’on veut !



On constate que les bordures font un peu « bugger », mais globalement on est bon !

# Mise en forme des éléments

Un petit peu de décoration, les marges autour, les bordures arrondies !

## Coloriser le fond

body

{

    /\*mise en couleur et en forme\*/

    background-color: #001C2F ;

}

## Configurer le container

#container {

    display: flex;

     /\*mise en couleur et en forme\*/

    background-color: #FFF2DC;

    border-radius: 15px;

    padding: 1%;

    margin: 1%;

    min-height: 90vh;

}

## Améliorer la nav

nav {

   /\* border: 3px solid black;\*/

    flex-grow: 1;

    /\*mise en couleur et en forme\*/

    background-color: #2A4E62 ;

    border-radius: 15px;

    padding: 1%;

}

Voyons ce que cela donne !



Pas mal non ? Nous allons maintenant remplir les différents containers :

* La barre de navigation
* La partie fiche séance
* Les infos bureau
* Les infos sur le local
* L’image du thème en cours
* L’agenda.

# Mise en place de la navigation

Il nous faut ajouter l’image du logo et le mettre en forme ! Facile, c’est la balise IMG !

<nav><div id="logoClub"></div></nav>

Grace à l’idée nous allons redimensionner l’image mais également lui donner le côté rond !

IL faut aussi ôter le flex-grow :1 sur le style de la nav.

nav {

   /\* border: 3px solid black;\*/

    /\*flex-grow: 1;\*/

    /\*mise en couleur et en forme\*/

    background-color: #2A4E62 ;

    border-radius: 15px;

    padding: 1%;

}

Pour l’image ronde voici le code !

#logoClub{

    border-radius: 100%;

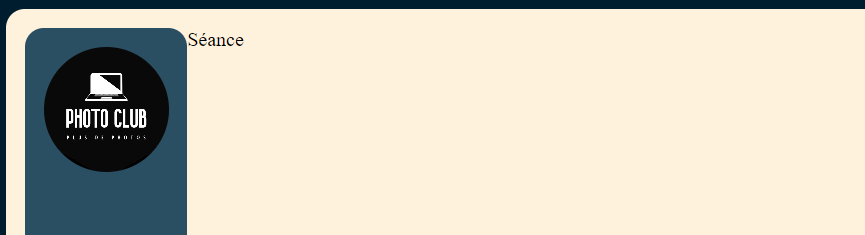
    background-size : cover; background-repeat: no-repeat;

    background-image : url(assets/icons/logo-photoclub.png);

    height: 100px;   width: 100px;    border: none;

}

Perfecto !



# Partie fiche séance

Concernant la séance, on voit bien un titre H1 et un paragraphe de texte. Il suffit de mettre le titre et ensuite ajouter le paragraphe, lui mettre un fond, des border radius et un peu de padding !

## Ajout du titre et du paragraphe

On ajoute le H1 et le P dans le div « divSeance » existant ! Pour le lorem : lorem20 est le raccourci (on peut changer le nombre suivant le nombre de mots désirés)

<div id="divSeance">

                <h1>Séance</h1>

                <p class="pGris">Lorem ipsum dolor sit amet consectetur adipisicing elit. Sunt cupiditate quaerat sed animi recusandae ducimus debitis nobis molestiae voluptatum repellendus.

                    Lorem ipsum dolor sit amet consectetur adipisicing elit. Voluptatem, iure.

                </p>

            </div>

## Mise en forme du divSéance

#divSeance

{

    flex-grow: 1;

    padding: 1%;

}

## Paragraphe de texte grisé

.pGris

{

    background-color: #B1BAC7 ;

    border-radius: 15px;

    padding: 1%;

}

Ca ressemble davantage, même si l’agenda se fait écraser : il est vide, pas de panique !

# Infos du bureau

Pour le bureau, cela donne un div par portrait. Ce div contient un div rond et deux paragraphes (centrés)

Voici la structure de base, à dupliquer 3 fois :

<div class="membreBureau">

                        <div class="rond"></div>

                        <p class="nomBureau"></p>

                        <p class="titrePoste"></p>

                    </div>

On va se contenter de mettre rond en rond, avec couleur de fond et aligner le texte avec les paragraphes

Pour membrebureau, il faut un flex en column pour centrer tous les éléments : cela sera plus joli !

.membreBureau{

    display: flex;

    flex-direction: column;

    align-items: center;

}

.Pour le portait à venir, il nous faut du radius !

.rond{

    border-radius: 100%;

    background-color: gray;

    width: 50;

    height: 50;

}

Pour finir, il nous faut du text-align : center pour centrer le texte des paragraphes. A ce stade, on ne va pas chercher les polices !

.nomBureau{text-align: center;}

.titrePoste{text-align: center;}

Pour le fond, petite modif sur le divBureau

#divBureau

{

    flex-grow: 1;

   height: 50%;

   align-items: center;

   display: flex;

 background-color: whitesmoke;

    border-radius: 15px;

}

On a plus qu’à ajouter les trois personnes !

Voilà ce que cela donne :

<div id="divGauche">

                    <h1>Bureau</h1>

                    <div id="divBureau">

                        <div class="membreBureau">

                            <div class="rond"></div>

                            <p class="nomBureau">Martin JONKO</p>

                            <p class="titrePoste">Président</p>

                        </div>

                        <div class="membreBureau">

                            <div class="rond"></div>

                            <p class="nomBureau">Josette PETRUCCIANI</p>

                            <p class="titrePoste">Trésorière</p>

                        </div>

                        <div class="membreBureau">

                            <div class="rond"></div>

                            <p class="nomBureau">Justin KIMBER</p>

                            <p class="titrePoste">Secrétaire</p>

                        </div>

                    </div>

                    <div id="divLocal">local</div>

                </div>

Côté taille, ce n’est pas fou, mais on va arranger plus tard !

# Infos sur le local

Il s’agit d’un titre et d’un texte ; rien de plus !

<div id="divLocal"><h1>Le Local</h1><p class="pBlanc">12 rue du paradis</p></div>

Côté classe, il faut mettre du blanc, du radius et du padding !

.pBlanc

{

    background-color: whitesmoke;

    border-radius: 15px;

    padding: 1%;

}

Bon plus qu’à ajouter le thème et l’agenda

# La photo du thème

Sans surprise, nous retrouvons un titre, une image et un paragraphe, le tout dans un div blanc

<div id="divTheme">

     <h1>Thème de la semaine</h1>

     <div class="pBlanc" id="divPhoto">

     <img class="photos" src="assets/images/IMG\_4468.jpg" alt="">

     <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Quasi, veritatis.</p>

     </div>

</div>

On réutilise une des classes !

Pour l’image :

.photos

{

    max-width: 300px;

}

Il nous manque un alignement pour le div !

#divPhoto {

    /\*il faut le mettre en flex pour aligner les choses dedans!\*/

    display: flex;

    flex-direction: column;

    align-items: center;

}

# Agenda !

Ici, il s’agit d’un titre et d’une série de paire de div inclus dans un div (pour faire les espaces !

Pour ajouter un mois voici le html

 <div id="divAgenda">

            <h1>Agenda à venir</h1>

            <div class="boxMois">

                <div class="boxMoisTitre"> Janvier</div>

                <div class="boxMoisSujet">Lorem ipsum dolor sit, amet consectetur adipisicing elit. Doloribus, commodi?</div>

            </div>

 </div>

Le code remplace le divAgenda bien sûr !

Côté mise en forme Voir en page suivante le CSS

Pour gérer l’espacement et la bordure simplement, voici le bon code. Overflow permet de faire que le div boxMois « coupe » ce qui dépasse des divs enfants. Testez pour voir sans overflow ! Les bords arrondis sont supprimés visuellement.

.boxMois{

    border-radius: 15px;

    margin-top: 5%;

    overflow: hidden

}

On modifie la mise en forme du titre de la case « mois »

.boxMoisTitre{

    text-align: center;

    background-color: #FFC9A3 ;

    padding: 1%;

}

On modifie la mise en forme du texte contenu dans la case moisSujet

.boxMoisSujet

{

text-align: justify;

background-color: whitesmoke;

padding: 1%;

}

# Etape finale ?

Pour arranger définitivement le design, deux retouches :

Pour les titres

h1

{

    font-weight: bold;

}

Pour les espaces autour des rectangles blancs

    margin: 2%;

à mettre dans

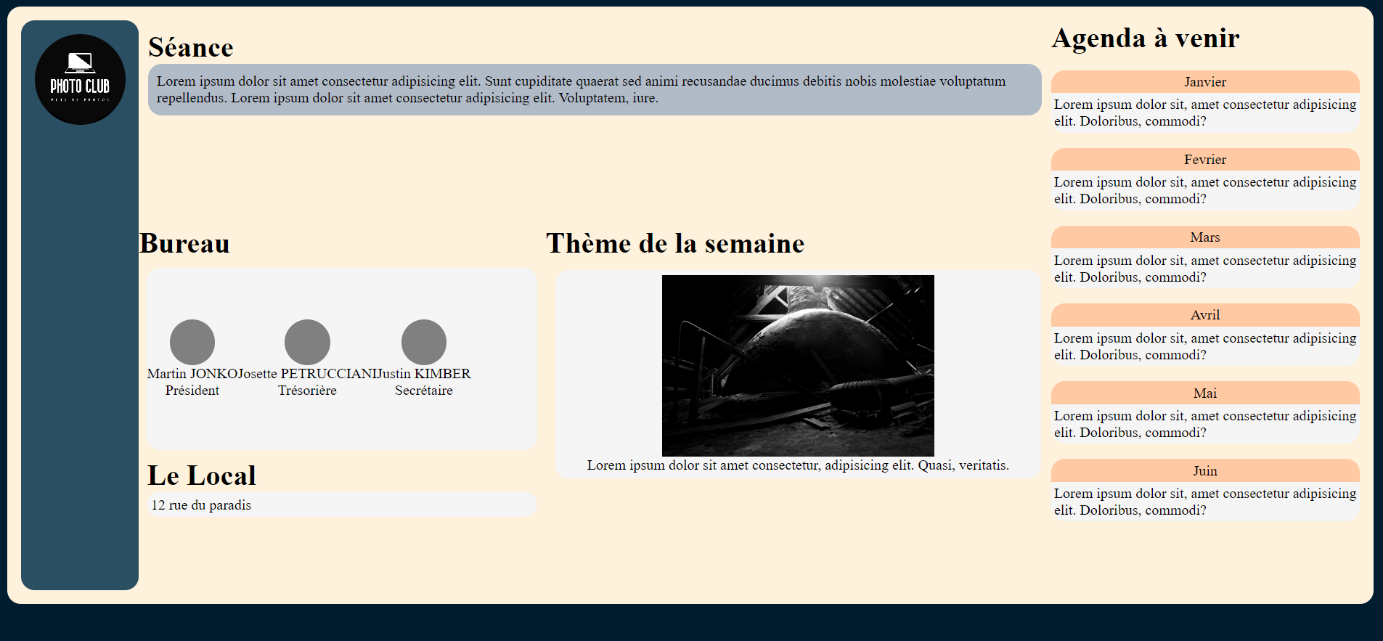
#divBureau

#divLocal

#divPhoto

Quel résultat avez-vous ? Comparons avec le mien !

Tadam !



# Etape finale : rendre responsive !

En résumé : tout ce qui est en flex avec flex-direction : row passe en flex-direction :column. Avec un peu de chance tout ira bien !

@media screen and (max-width:700px) {

    #container {flex-direction: column;

    }

    #divPartieBasse {flex-direction: column;

    }

}

Bon, ce n’est pas de la chance, c’est de l’anticipation !

Le resultat final en dernière page

