Usage

Ces fichiers sass seraient utilisé pour la création de site web.

On ne devrait jamais modifier des fichiers sauf 2 couleurs et style.scss. Les couleurs sont des variables qui doivent être utilisé poiur les clients.

Le fichier style.scss load (importe) tous les modules mais on peut commenter et décommenter des élément pour les inclures ou non dans notre projet. Cependant, il ne faut pas changer l'ordre des élément car certains élément dépendent de d'autre.

Si il y. a des modifications spécific à faire, on devrait creer des fichiers distincs selon les modification à faire. Par la suite, si les modifications sont intéressant, nous les inclurons dans les prochaines versions de notre package.

—

Les modifications ne seront pas appliquer au ancien projet puisque ces derniers devraient fonctionnés,

—
Par défault la font est avenir nex mais pour certain client il va falloir changer la font
—

Par défault, vous devrier seulement ajouter les classes a du html
—
Certain élément rest a vérifier donc à mesure qu'on les utilise ils seront tester et corriger au besoin.

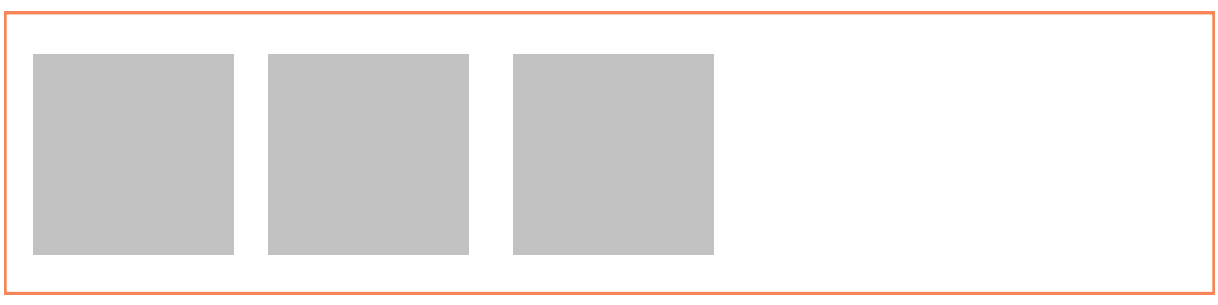
Core Module

row

La disposition se fait en rangées et en colonnes. On a de nouvelles façons de disposer nos éléments.

De base, dans une rangée, les éléments à l'intérieur sont alignés sur une ligne et les éléments se mettent tous en haut à gauche. De plus, la rangée a toujours 100% de la largeur de la page.

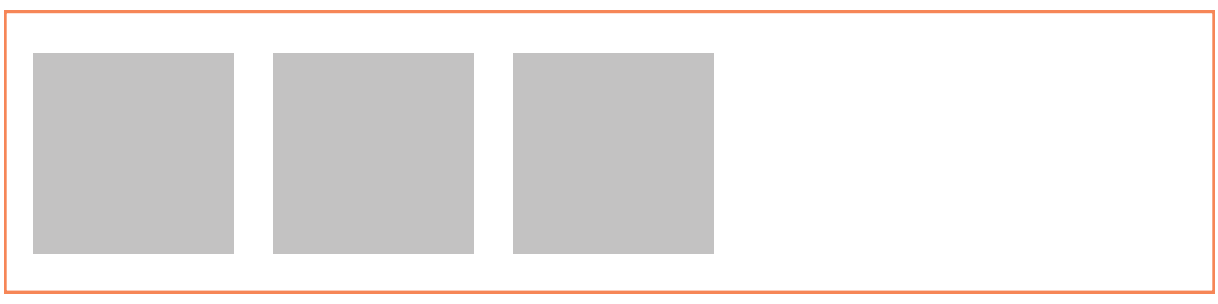
.row



```
<div class="row">…</div>
```

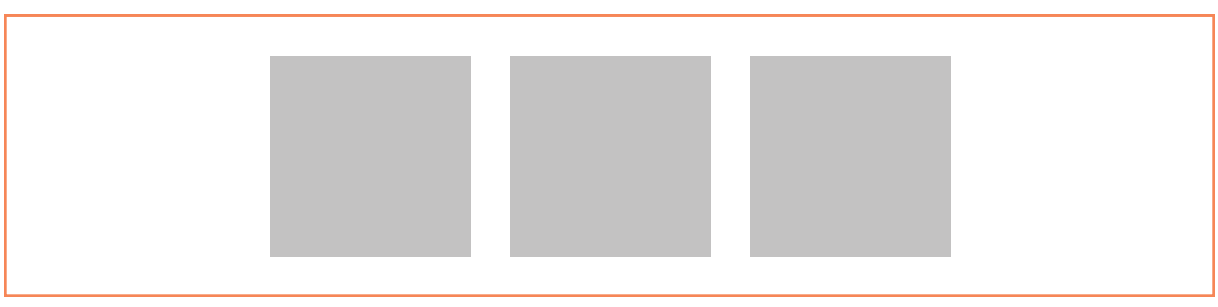
Cependant, selon les cas ont peut disposer les éléments au centre, à droite ou séparés entre eux.

.row.item-left



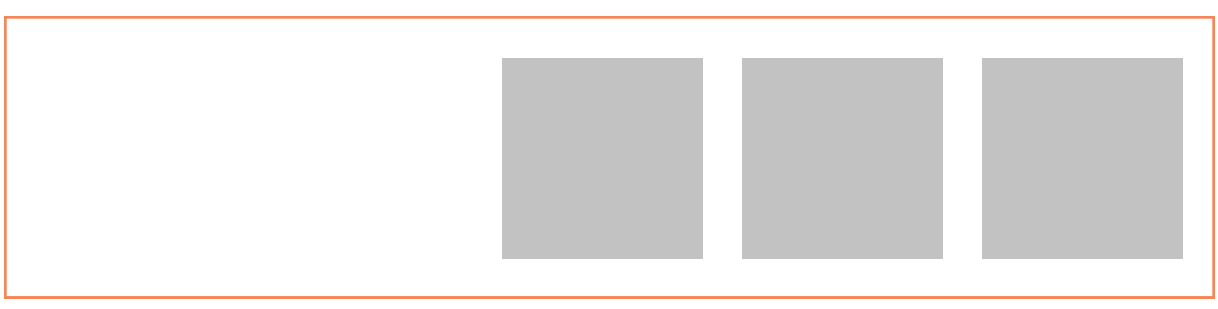
```
<div class="row item-left">…</div>
```

.row.item-center



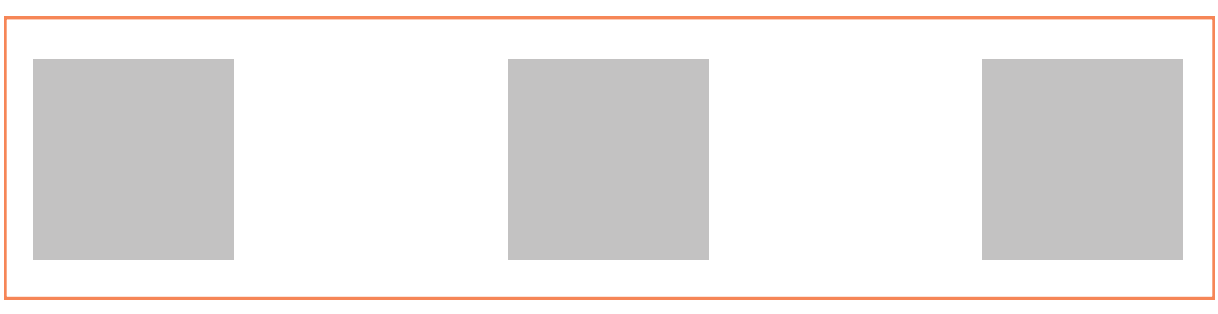
```
<div class="row item-center">…</div>
```

.row.item-right



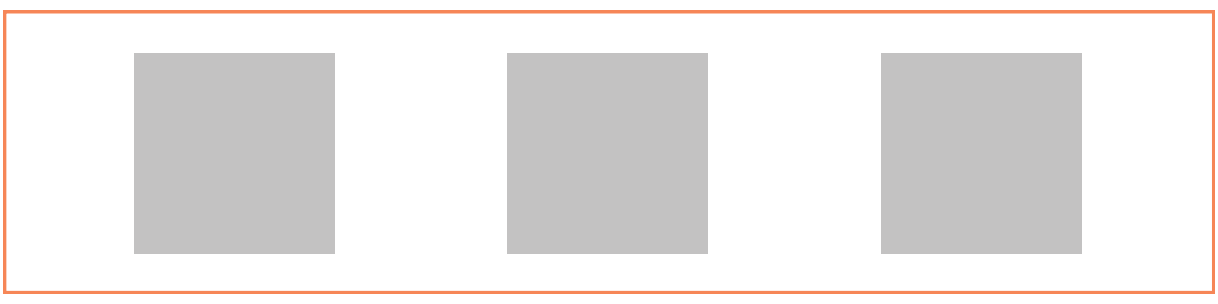
```
<div class="row item-right">…</div>
```

.row.item-split



```
<div class="row item-split">…</div>
```

.row.item-around



```
<div class="row item-around">…</div>
```

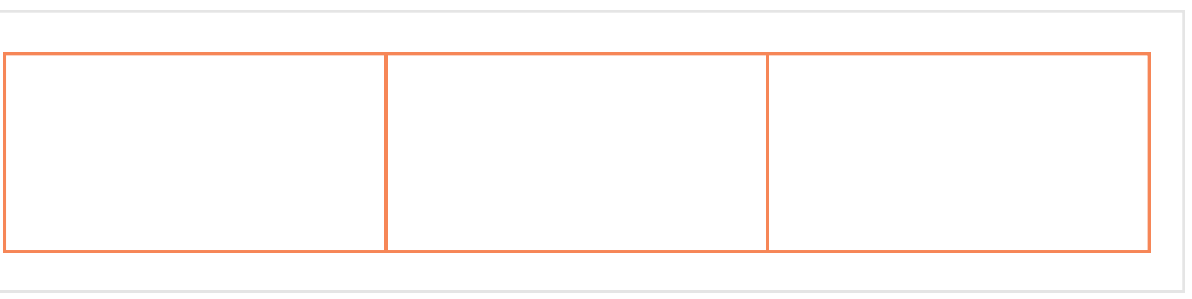
.row.item-top (par défault les items sont centrés)

```
<div class="row item-top">…</div>
```

col

Les colonnes ont aussi évoluées. On a beacoup de possibilités.De base, elles utilisent toute la rangée à part égale.

.col

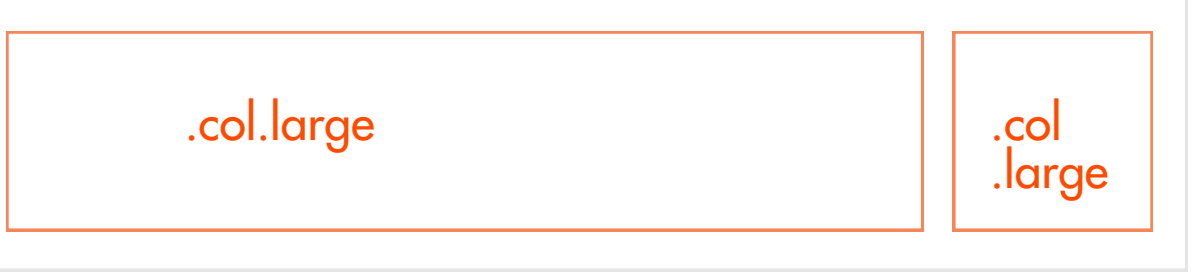


```
<div class="row">
    <div class="col">...</div>
    <div class="col">...</div>
    <div class="col">...</div>
</div>
```

On peut maintenant leurs donner une grandeur prédéterminée de 20% à 100% par tranche de 20. Aussi il y a full qui prend toute la largeur disponible sans auncune marge

.xsmall -> 20%
.small -> 40%
.medium -> 60%
.large -> 80%
.xlarge -> 90%
.full -> 100% (sans marge)

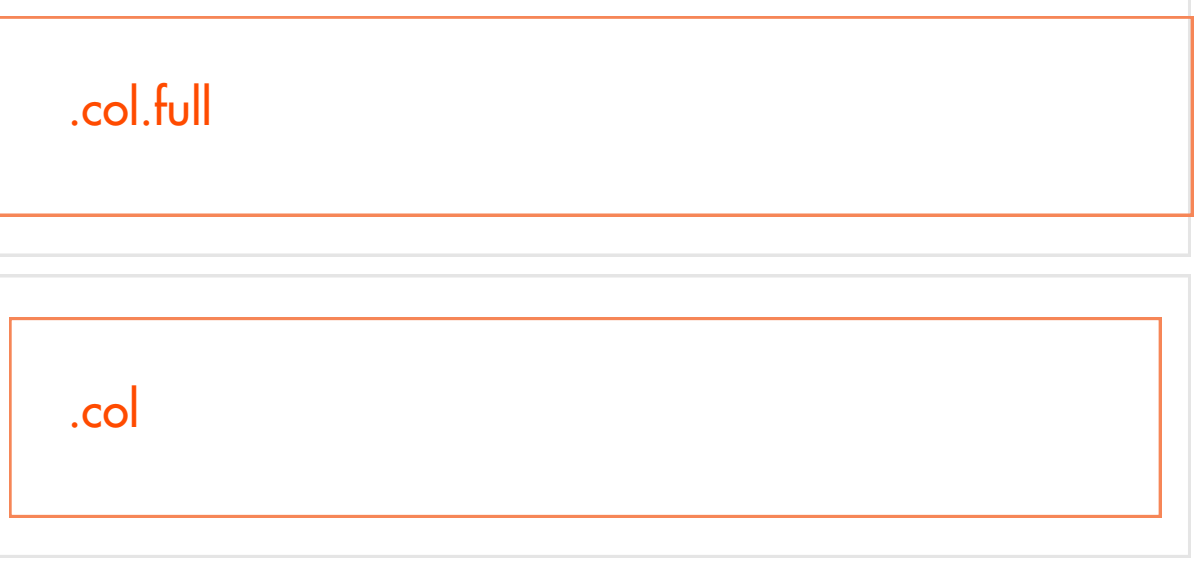
Le premier élément prend l'espace déterminé. Les autres s'ajustent avec ce qui reste. Ça ne peut pas dépasser 100%.




```
<div class="row">
    <div class="col large">...</div>
    <div class="col large">...</div>
</div>
```

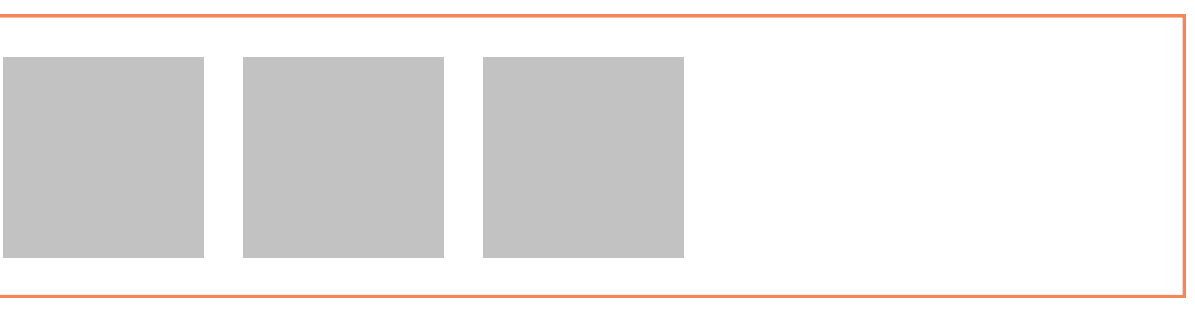
Mais le total peut être plus petit que 100%.

Il est suggéré de finir la colone avec un .col (sans size). Ce dernier va remplir l'espace restant.



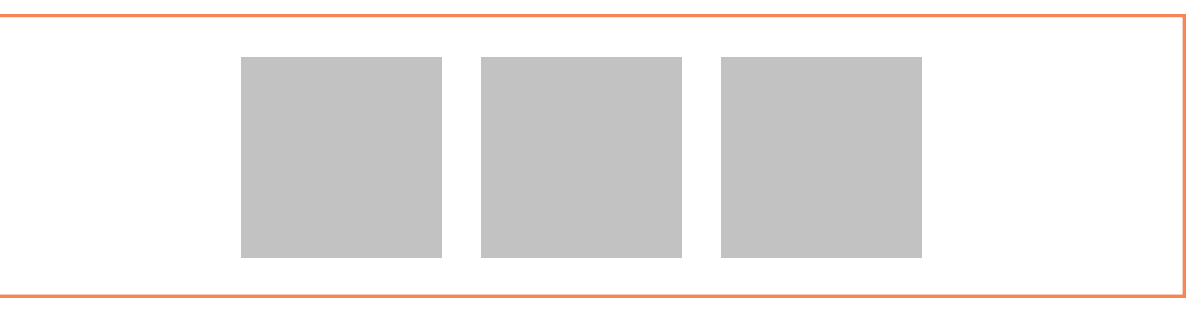
Cependant, selon les cas ont peut disposer les éléments au centre, à droite.

.col.item-left



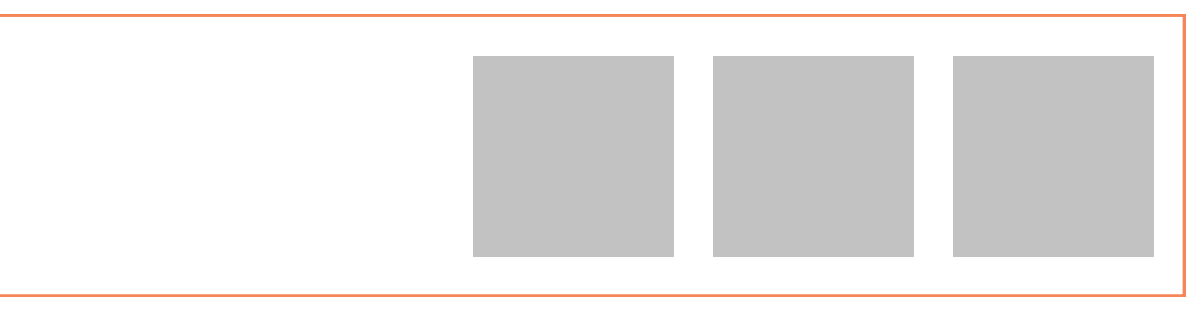
```
<div class="col item-left">...</div>
```

.col.item-center



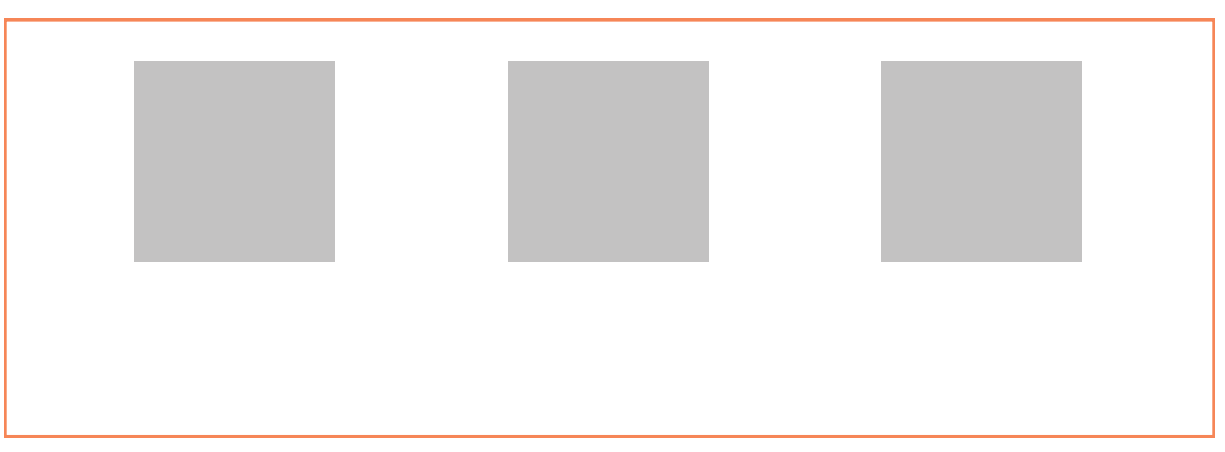
```
<div class="col item-center">...</div>
```

.col.item-right



```
<div class="col item-right">...</div>
```

.col.item-top (par défault les items sont centrés)



```
<div class="col item-top">...</div>
```

.col.auto

Prend exactement la largeur des éléments à l'intérieur peut importe la largeur. C'est pour être sur d'avoir un pur item split ou item around dans la rangée

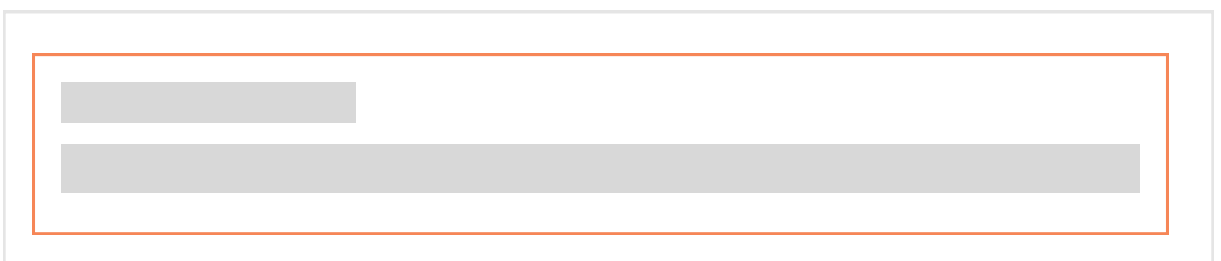
```
<div class="col auto">...</div>
```

.wrap .nowarp

On peut maintenant choisir la disposition des éléments dans la colone. Par default, les éléments, labels et inputs par exemple, vont s'aligner les uns par dessus les autres.

Pratiquement, en UX, on suggère de mettre les labels au dessus des inputs pour être plus rapide à remplir par les utilisateurs.

.col.size.wrap



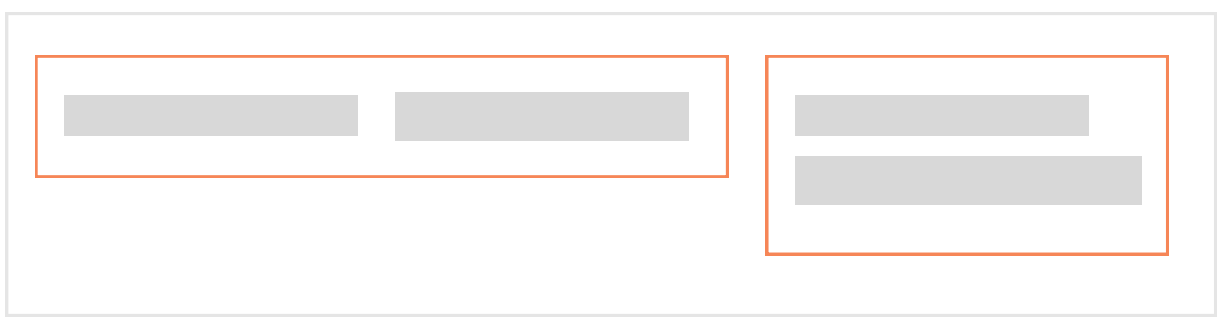
```
<div class="row">
    <div class="col xlarge wrap">
        ...
    </div>
</div>
```

Si on met nowrap les élément ce centre sur une même ligne. Selon les attributs des éléments le sizing devrait s'ajuster.

.col.size.nowrap

```
<div class="row">
    <div class="col xlarge nowrap">
        ...
    </div>
</div>
```

On peut mettre une colone en wrap et l'autre nowrap mais c'est vraiment pas à prélivégier. En mobile, tout wrap.



```
<div class="row">
    <div class="col nowrap">     </div>
    <div class="col wrap">   </div>
</div>
```

Font

Les polices sont prédéterminées dans le fichier font-face. Elles sont dans un ordre de priorité, Avenir Next, Helvetica, Verdana et sans-serif.

Ce sont des placeholders, donc on peut les mettre dans notre code css sans trop de difficulté.

```
h1{
@extend %avenir-heavy;
…
}
```

%avenir-ultra

%avenir

%avenir-medium

%avenir-demibold

%avenir-bold

%avenir-heavy

On peut voir un aperçu plus détaillé des font dans le dossier font. Il y a un HTML avec les différentes tailles des fonts.

Conseil: on essaye de jouer avec le contraste par exemple si le titre est en médium le texte ne devrait pas être régulier (etencore moins médium) on essaye d'avoir un contraste d'au moins 2 tailles (médium à ultra).

Titre

La grosseur des titres et des textes est déjà prédéterminée. On a de h1 à h6 comme titre. La typo est au choix du client mais de base on a une typo sans serif. Les class mobiles sont prédéterminées.

La grosseur des autres éléments de texte est aussi pédéterminée selon l'élément. De base, le texte dans un p serait de 1em soit environ16px.

h1
39.063 px

h2
31.25 px

h3
25 px

h4
20 px

h5
16 px

h6
12,8 px

h1
31.25 px

h2
25 px

h3
20 px

h4
18 px

h5
16 px

h6
12,8 px

Media Querie

Présentement on a pas de hack spécial pour les différents browser. Tout est gérer par Autoprefixer.

Il gère ie11 et plus et les 2 dernieres versions des navigateurs plus modernes.

Les classes sont déjà configurées pour les différentes tailles d'écran.

Voici les breakpoints:

640px	$iphone-land
768px	$tablet-portrait
1024px	$tablet-land $desktop
1025px	$desktopnotouch
1920px	$desktop-ultimate

Certains media queries sont utilsés pour les image haute résolution, mais des plugins Gulp et des mixins règlent ces problèmes.

Il y a aussi un fichier fake_mediaquery.scss qui doit être importé à la fin de tout. Il créer la classe .mobile-only, .tablet-and-up et .desktop-only. Ces classes affichent des éléments seulement en mobile, tablet ou desktop.

Couleur

On se limite a 5 couleurs bases pour nos styles.
On utilise les noms:

black
white
accent
primary
secondary

.bgcolor-primary
.textcolor-black

Et 3 couleurs pour les « validations »

success warning alert

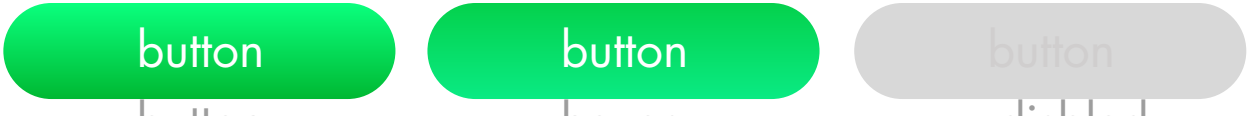
Il y a aussi des couleur pour les « brands » suivante:

twitter
googleplus
linkedin
vimeo
flickr
foursquare
facebook
pinterest
youtube
instagram
dribbble

Button

Que se soit un input ou un bouton avec un type submit les boutons ont 3 états. Leur taille est gérée par les données à l'intérieur du button. La couleur est gérée par les 2 variables $accent et $accent-light. Il y a aussi un classe .gray qui permet de créer un bouton gris mais pas disabled.

on peut aussi dans les préférence du bouton _elementBTN.scss choisir si le bouton a les coins ronds ou carrés.



button hover disbled

```
<button type="submit">button</button>
— ou —
<input type="submit" value="button" />
```

Layout component

.accordion

Accordion cache les colonnes ou les sections sous lui. Il peut se placer dans une rangée ou une colonne. Il va prendre 100% de la largeur disponible. Par défault l'accordéon n'a pas d'icône. Le titre doit absolument être un h2.

.accordion

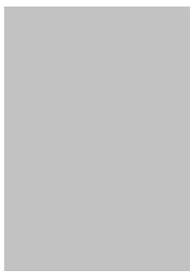
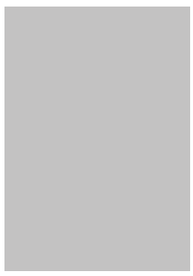
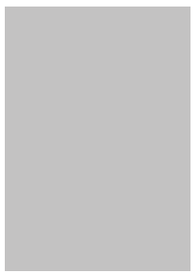
Titre

```
<div class="row accordion">
    <h2>titre</h2>
    <div class="row">
        ...
    </div>
</div>
```

En ajoutant .open les éléments apparaîssent. Il faut créer ou modifier un javascript qui ajoute la classe open.

.accordion .open

Titre

```
<div class="row accordion open">
    <h2>titre</h2>
    <div class="row ">
        ...
    </div>
</div>
```

Si on utilise accordion-icon au lieu un icône se place à la droite du titre.

.accordion-icon

Titre

Hero image mixin

Le mixin hero image permet de gérer les images responsives dans les «heros» comme cover image. Comme dans le site d'Analystik ou Signder.

Cependant, vous devez aller dans modules/component/layout/hero-image et changer le fichier _element.scss

Vous devez simplement créer la nouvelle class avec le mixin

```
.nom_de_la_class{
    @cover_image('image_path',extention,cover )
}
```

nom_de_la_classe sera à votre discretion
image_path sera le chemin pour l'image par exemple
'/Content/images/nom_de_image'
extention sera le type d'extention (jpg, png ou gif)
cover est le type ou size peut être cover ou contain.

Par la suite, vous utilisez votre nouvelle classe dans un div.

Sticky header

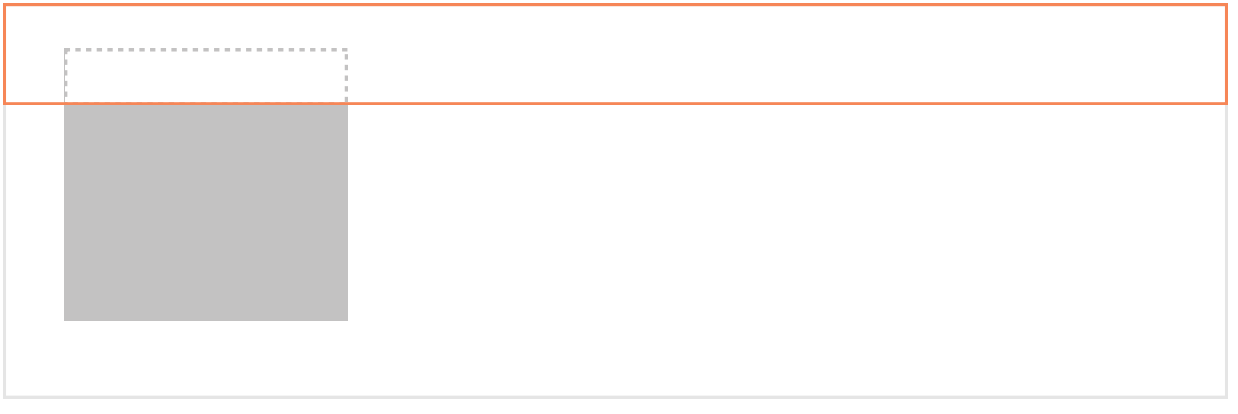
Le sticky header permet d'avoir un menu toujours accessible, peu importe où on se trouve dans la page.

.sticky

```
<div class="row sticky">
    ...
</div>
```

Le menu s'accroche à une rangée. De base, le menu n'est pas visible. En ajoutant la class visible, à l'aide d'un javascript (positionnement de la fenêtre), le menu se fixe au haut de la page. Le menu est toujours au-dessus des éléments de la page.

```
<div class="row sticky visible">
    ...
</div>
```



Element component

Blockquote

Tout ce qui est du blockquote est déjà prêt. L'élément de la citation et l'auteur; la couleur du texte, la grosseur et le style.

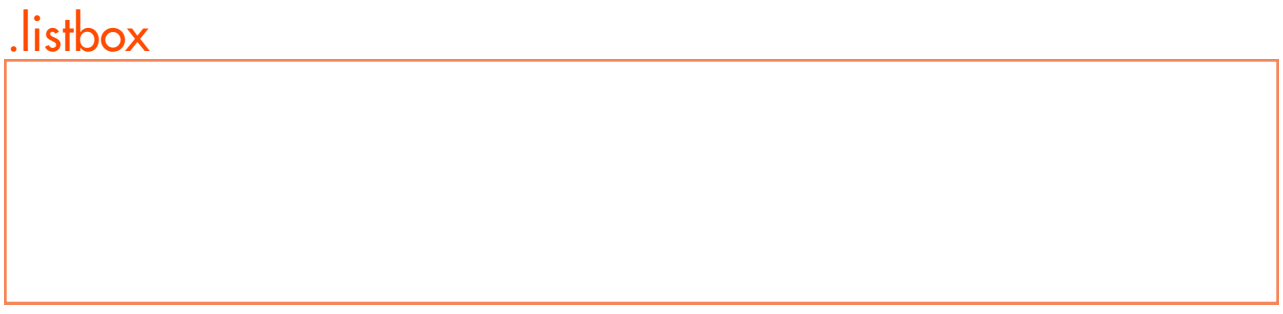
```
<blockquote>
    Less is more
    <cite>Ludwig Mies Van Der Rohe</cite>
</blockquote>
```

> Less is more
>
> *- Ludwig Mies Van Der Rohe*

Liste

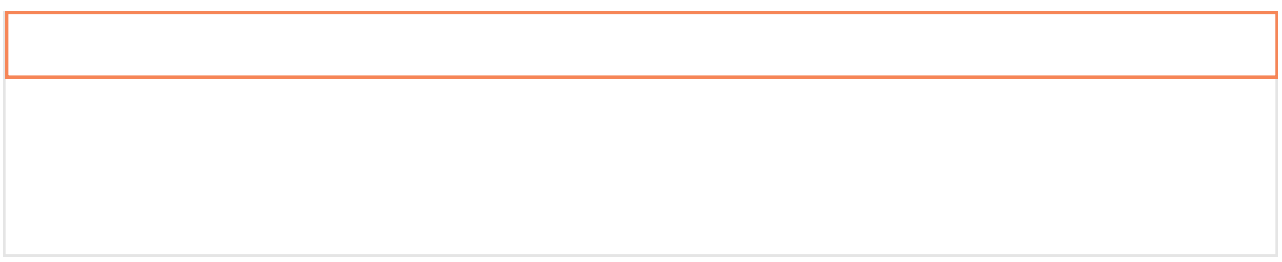
On découpe tous les éléments. Les cellules sont des div. independants.

.listbox



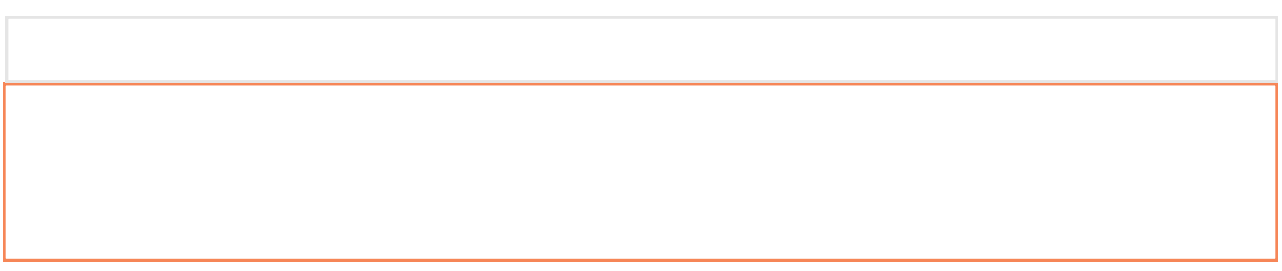
L'élément listbox découpe seulement le cadre de la liste.

.head



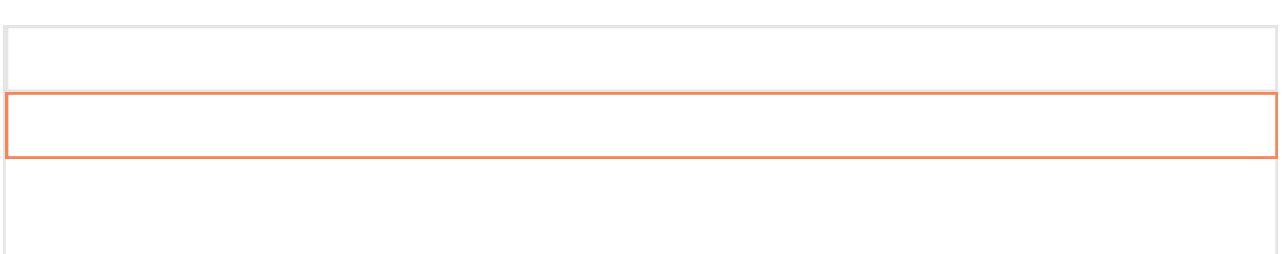
Le head est une rangée spéciale qui crée les cellules d'en-têtes.

.listbody



Le listbody est le groupe de rangées de la liste sans le header.

a.row



Crée les autres rangées de la liste.

.col.size

Seulement les col.size sont utilisées dans les listes.

```
<section>
    <div class="listbox">
        <div class="head">
            <div class="col xsmall">Id</div>
            <div class="col small">Name</div>
            <div class="col medium">Note</div>
        </div>
        <div class="listbody hsmall">
        <a class="row">
            <div class="col xsmall left">6</div>
            <div class="col small">John Doe</div>
            <div class="col medium">Client modèle</div>
        </a>
        <div>
    </div>
</section>
```

Les sizes des colonnes du head doivent se répéter dans les autres colonnes des rangées au dessous.

Aux colonnes peuvent aussi être ajoutées la classe left ou right pour aligner les items à gauche où à droite.

Le listbody peut avoir des size de hauteur qui équivaut à un nombre de ligne

.hxsmall -> 4 lignes de haut

.hsmall -> 6 lignes de haut

.hmedium -> 10 lignes de haut

.hlarge -> 15 lignes de haut

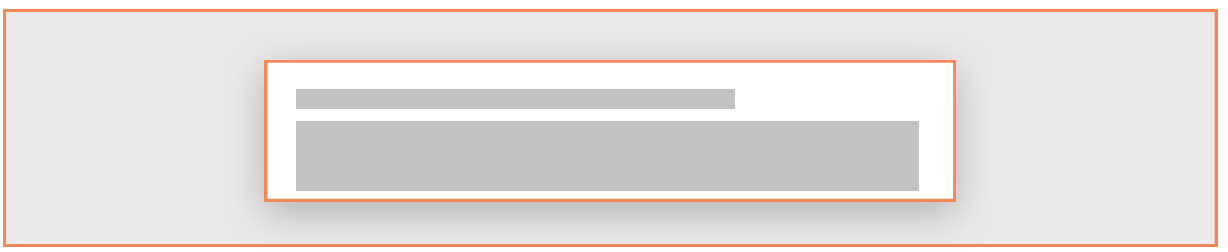
.hxlarge -> 20 lignes de haut

.modal-container

Cette classe permet de créer des modals. Le modal n'a pas besoin d'être dans un row ou col.

Le background et l'animation sont créés, il manque juste un javascript qui rajoute open pour afficher le modal.

```
<div class="modal-container">
    <div class="modal">
        ...
    </div>
</div>
```



```
<div class="modal-container open ">
    <div class="modal">
        ...
    </div>
</div>
```

On peut afficher un x à droite du modal, en enlevant les commnetaires autour du fichier « x-on-modal » dans le fichier styles.scss

progress

Cet élément permet de créer des barres de progression avec un tag html seulement.

Si le propress bar est dynamique, il faut créer un javascript qui permettra de mettre les données dans le propress bar.

```
<progress value="57" max="100"></progress>
```

Le dégrader a une couleur « de fin » dépendant de la valeur attribuée au progress bar.

```
<progress value="28" max="100">
</progress>
```

Ce principe de couleur fonctionne seulement dans Chrome et Safari.

Dans Firefox, le progress bar est uniquement bleu avec 3d

Dans Edge et IE, le progress bar est uniquement bleu sans 3d

.tooltip

La classe tooltip s'accompagne de -left ou -right, pour la direction, si le tooltip est petit. Il permet de positioner le tooltip à gauche ou à droite. Cependant, quand le texte est long, le tooltip prend toute la largeur de la page.

Le tooltip se place simplement dans un div.


```
<div class="tooltip-left">...(texte)...</div>
```

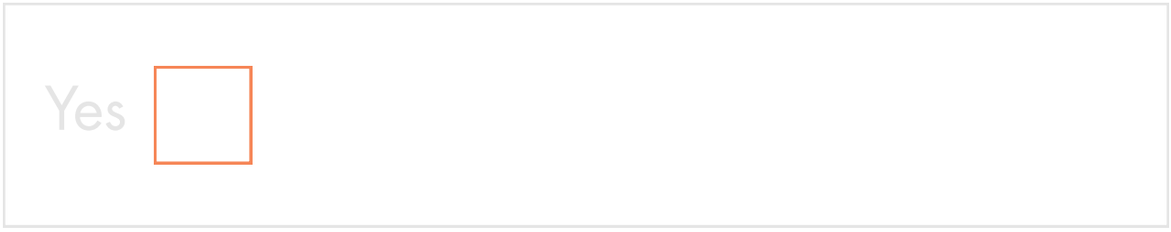
Lorsque l'on passe la souris sur le carré ce dernier disparaît, et le texte, à l'intérieur du div, apparaît.

Form component

Checkbox et radiobutton

Pour avoir un element custom pour les checkboxs et radio buttons, on doit changer un peu le code. Les « for » et les « id » sont essentiels. On peut créer un autre div si on ne veut pas l'importer dans la colonne.

.checkbox




```
<div class="row">
    <div class="col checkbox"
        <label>Yes</label>
        <input type="checkbox" id="yes"  />
        <label for="yes"></label>
    </div>
</div>
```

.radiobutton


```
<div class="row">
    <div class="col radio">
        <label>Homme</label>
        <input type="radio" name="gender' id="homme"/>
        <label for="homme"></label>

        <label>Femme</label>
        <input type="radio" name="gender" id="femme"/>
        <label for="femme"></label>

        <label>Autre</label>
        <input type="radio" name="gender" id="autre"/>
        <label for="autre"></label>
    </div>
</div>
```

Si on veut le label à droite, on peut simplement le mettre à la fin du block au lieu du début.

Label

Il n'y a pas beaucoup de particularités avec eux. Ils ont toujours une largeur auto. Ils prendront toujours la place disponible pour être sur une ligne. Un label devrait avoir un nombre limité de mot, 2 ou 3 maximum.

label


```
<div class="row">
    <div class="col"
        <label></label>
        <input type="text"  />
    </div>
</div>
```

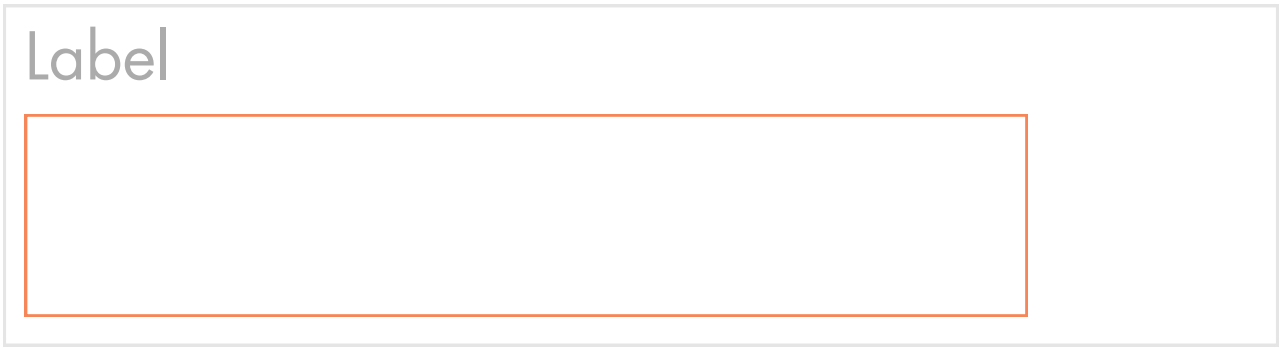
On devrait écrire un label avec un « for » pour un meilleur UX. Lorsque l'on click sur le label avec un «for», le curseur va directement dans le input associé.

label


```
<div class="row">
    <div class="col"
        <label for="name"></label>
        <input type="text" id="name"   />
    </div>
</div>
```

Textarea

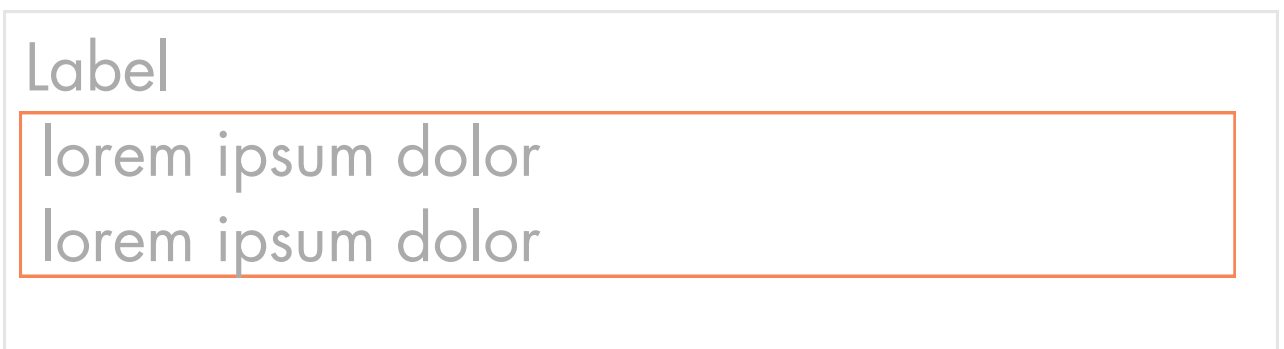
textarea



Le label devrait toujours être en haut du textearea.

```
<div class="row">
    <div class="col">
        <label>Label</label>
        <textarea></textarea>
    </div>
</div>
```

Le textarea prend toute la largeur disponible dans la colonne.
On peut aussi mettre la hauteur du textarea avec des classes.
On applique directement ces classes au textarea. Les classes sont line1 à line10




```
<div class="row">
    <div class="col">
        <label>Label</label>
        <textarea class="line2"></textarea>
    </div>
</div>
```

À vérifier

.curency et pourcent

Ces deux classes sont déjà prêtes il faut simplement mettre un label vide à la fin. On applique la classe directement sur la rangée ou la colonne. Sauf que la colonne ne peut pas wrapper. Donc, il est plus sécuritaire de l'utiliser dans un div séparé.


```
<div class="row">
    <div class="col wrap"
        <label>…</label>
        <div class="currency">
            <input type="number"  />
            <label></label>
        </div>
    </div>
</div>
```

Le $ ou le % se rajoute automatiquement suite au dernier label. Le premier label doit être à l'extérieur du div. Si on veut tout mettre sur une rangée, on a juste à mettre .no-wrap à la colonne.


```
<div class="row">
    <div class="col nowrap">
    <label>pourcentage</label>
        <div class="pourcent"
            <input type="number"/>
            <label></label>
        </div>
    </div>
</div>
```

À vérifier

.input-group

Cette classe permet de recréer le même principe que .pourcent ou .currency, mais en créant des éléments personnalisés en avant et un bouton à la fin si nécessaire.

Label

usd send

```
<div class="row">
    <div class="col wrap">
        <label>...</label>
        <div class="input-group">
            <label>usd</label>
            <input type="number"  />
            <input type="submit" value="send" />
        </div>
    </div>
</div>
```

Le premier label se style automatiquement. Le bouton à la fin, aussi, le texte choisit peut être modifié.

Il est possible aussi de faire le même principe sans bouton à la fin en utilisant la classe .input-group-noSubmit.
Il faut alors supprimer le bouton à la fin dans le code.

Label

usd

```
<div class="row">
    <div class="col wrap">
        <label>...</label>
        <div class="input-group-noSubmit">
            <label>usd</label>
            <input type="number"  />
        </div>
    </div>
</div>
```

Template specific

.accordeon

La classe accordeon ressemble beaucoup à l'accordéon dans les layouts compoenents.Il va prendre 100% de la largeur disponible. Le titre doit absolument être un h2. Mais il y a une icône et se place à la droite un titre dans le template.

.accordeon

Titre

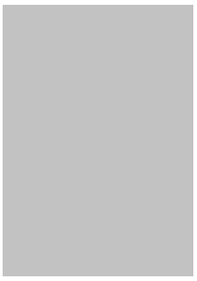
```
<div class="accordeon">
    <h2>Titre
        <span class="icon">
            <i class ="fas fa-angle-double-down"></i>
        </span>
    </h2>
    <div class="row">
        ...
    </div>
</div>
```

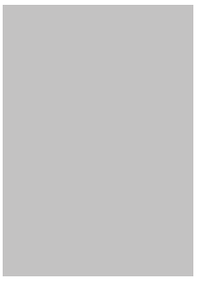
La classe .icon est nécessaire pour le bon affichage de font-awesome. Cette font créée une icône pour les éléments.

En ajoutant .open les éléments apparaissent. Il faut créer ou modifier un javascript qui ajoutera la classe open.

.accordeon .open

Titre





```
<div class="accordeon open">
    <h2>Titre
        <span class="icon">
            <i class=""fas fa-angle-double-down"></i>
        </span>
    </h2>
    <div class="row">
        ...
    </div>
</div>
```

.enbref

En bref ressemble à la classe accordéon, mais elle est un bloque totalement différent qui permet de voir certaines informations. Son fonctionnement y est identique.

.enbref

Titre

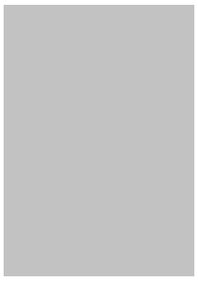
```
<div class="enbref">
    <h2>Titre
        <span class='icon'>
            <i class="fas fa-angle-double-down"></i>
        </span>
    </h2>
    <div class="row">
        ...
    </div>
</div>
```

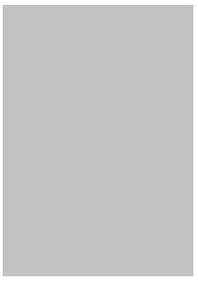
La classe .icon est nécessaire pour le bon affichage de font-awesome. Cette font créée une icône pour les éléments.

En ajoutant .open les éléments apparaissent. Il faut créer ou modifier un javascript qui ajoutera la classe open.

.enbref .open

Titre





```
<div class="enbref open">
    <h2>Titre
        <span class="icon">
            <i class=""fas fa-angle-double-down"></i>
        </span>
    </h2>
    <div class="row">
        ...
    </div>
</div>
```

.infobox

L'infobox est une des classes clés du template. Elle sert pour la création de boîtes de contenu qui seront remplies. Elles servent pour tous les éléments qui ne sont pas des boutons ou dans la sidebar. Le triangle est déjà intégré. Ces boîtes remplissent les colonnes à 100%.

.infobox

Titre

```
<div class="col infobox">
    <div class="header">
        <h2 class="infotitle">Titre</h2>
    </div>
    <div class="row">
        <div class="col">
            ...
        </div>
    </div>
</div>
```

Comme les boîtes remplissent les colonnes à 100%,s'il y a plus qu'une boîte, les boîtes s'aligneront les unes a coté des autres.

La classe .infotitle permet de mettre plusieurs éléments dans le header qui vont se séparer uniformément dans la boîte (un à gauche l'autre à droite).

.infobox

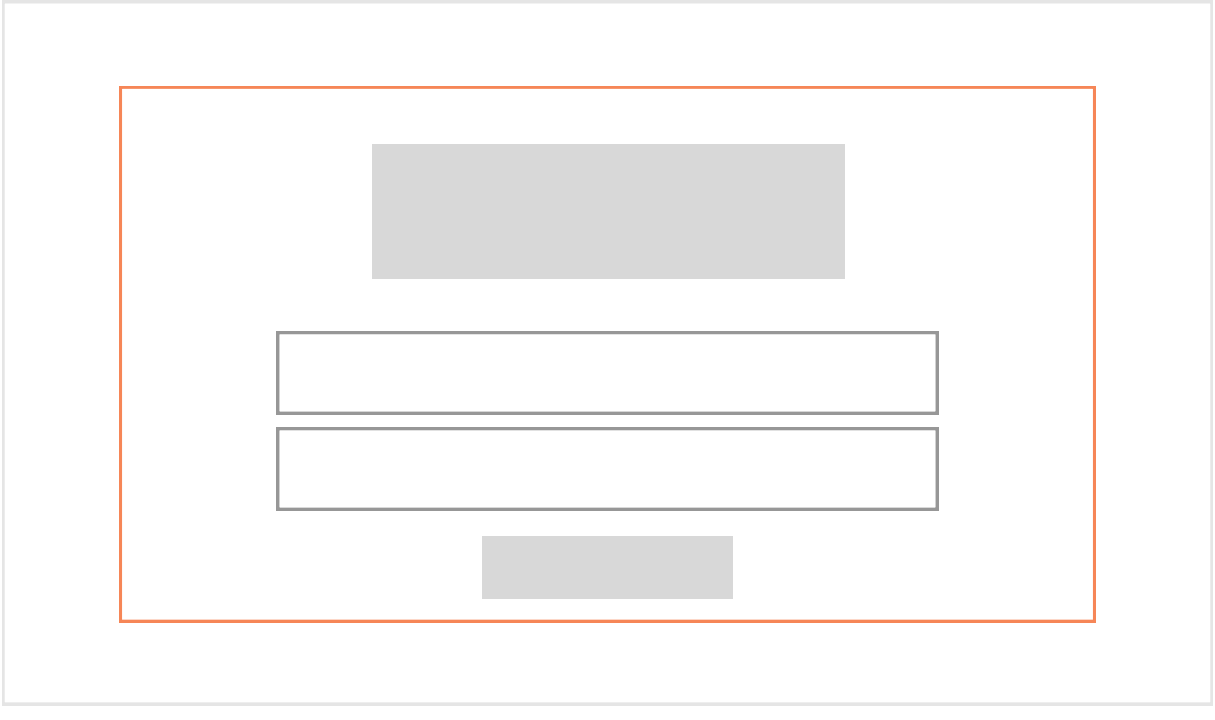
Titre Titre

```
<div class="col infobox">
    <div class="header">
        <h2 class="infotitle">Titre</h2>
        <h2 class="infotitle">Titre</h2>
    </div>
    <div class="row">
        <div class="col">
            ...
        </div>
    </div>
</div>
```

.login-screen

Il sert principalement à créer la boîte login. Le logo, les inputs et le bouton est configuré grâce aux autres .scss mais ils ont une position prédéterminée dans l'écran.

.login-screen



```
<div class="login-screen">
      <div class="row">
         <div class="col item-center">
            <img srcset="
                  images/logo.png 357w,
                  images/logo@2x.png 714w,
                  images/logo-@3x.png 1071w"
                  src="images/logo.png" alt="Logo" />
            <input type="text" placeholder="Username" />
            <input type="password" placeholder="Password" />
         </div>
      </div>
      <div class="row item-center">
         <button>Login</button>
      </div>
   </div>
```

nav

Cet élément sert à créer la barre de navigation dans le haut de la page. Utilisé conjointement avec menu-mobile cela permet d'avoir une navigation fonctionnelle sur tous les formats.

Plusieurs choses changent d'un client à l'autre, mais le code, ici-bas, crée une navigation, le logo, 2 onglets, un bouton et une image de profil.

nav

```
<nav>
    <div class="row">
        <div class="col xsmall">
            …(mobile stuff)
          <div class="logo tablet-and-up">
           <img src="images/logo-emalaya.png" alt="Logo" />
           </div>
        </div>
        <div class="col">
            <div class="row bar-right">
                <div class="menu-top">
                    <a><span class="icon-menu">
                        <i class=" FA_ICON "></i>
                        </span>Element 1
                    </a>
                    <a><span class="icon-menu">
                        <i class=" FA_ICON "></i>
                        </span>Element 1
                    </a>
                </div>
                <button class="button-header">
                    …
                </button>
                <div class="col xsmall">
                    <div class="profile">
                    <span class="profile-name desktop-only »>
                    …</span>
                    <div class="profile-image"></div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </nav>
```

.title

Cette classe crée un titre dans la section de droite. Le titre prend 100% la section de droite

.title

Titre

```
<h1 class="title">titre</h1>
```

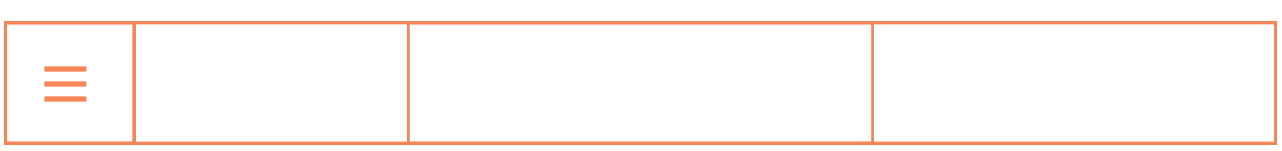
.menu-mobile

Le menu mobile change les fonctionnalités du menu pour permettre l'affichage sur les petits écrans (plus mince que 768px). Techniquement, il devrait avoir la classe .mobile-only et le menu normal .tablet-and-up pour ne pas avoir un double affichages. Les onglets du menu normal disparaissent dans le menu mobile.

.menu-mobile

```
<div class="col xsmall">
    <div class="hamburger-menu mobile-only">
        <div class="burger">
            <i class="fas fa-bars"></i>
        </div>
    </div>
<div class="logo tablet-and-up">
    ...
</div>
—
<div class="menu-mobile mobile-only">
    <div class="col">
        <h3 class="title-menu">
            <a>
                <span class="icon">
                    <i class="far fa-file-alt"></i>
                </span>Fiches de temps
            </a>
        </h3>
            <ul class="list-menu">
                <li><a>Élément 1</a></li>
                li><a>Élément 2</a></li>
            </ul>
        </div>
    </div>
```

.menu-mobile.open

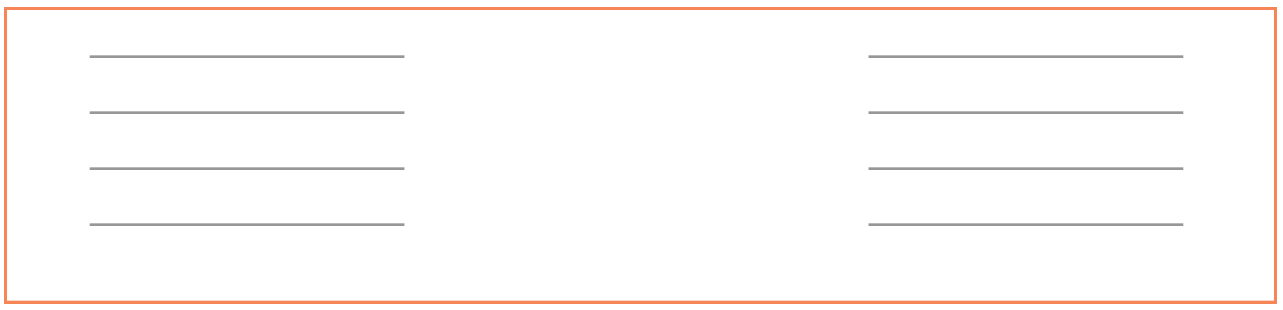


À l'aide d'un javascript, on ajoute la classe .open sur menu-mobile, ce dernier s'affiche une boite avec les liens.

.list temp

List temp permet de creer une liste d'objet dans le template.
Une liste de compagnie par exemple.
Chaque ligne a une couleur différente pour permettre de séparer les objets rapidement. Aussi la hauter est varisble selon le nombre d'élément

.list-temp



```
<a class="list-temp">
    <div class="row">
        <div class="col">
            ...
        </div>
        <div class="col">
            ...
        </div>
    </div>
</a>
```

Chaque a va donner un nouvel objet clickable. La liste sera zebrée par ligne.

À vérifier

Template specific Sidebar

.intertire-icon

Cet élément crée le titre d'une liste


```
<aside class="col xsmall sidebar tablet-and-up">
                <h5 class="intertitre-icon"><span class="fas fa-bolt"></span>Titre</h5>
                <ul class="sidebar-list">
                    <li><a class="selected">element 1</a></li>
                    <li><a>element 2</a></li>
                    <li><a>element 3</a></li>
                </ul>
            </aside>
```

.sidebar-list

Une simple liste de liens pour la sidebar. Lorsque l'on ne veut pas d'élément spécial, mais juste des liens.

.sidebar-list


```
<aside class="col xsmall sidebar tablet-and-up">
                <h5 class="intertitre-icon"><span class="fas fa-bolt"></span>Titre</h5>
                <ul class="sidebar-list">
                    <li><a class="selected">element 1</a></li>
                    <li><a>element 2</a></li>
                    <li><a>element 3</a></li>
                </ul>
            </aside>
```

L'élément avec la classe .selected sera plus foncé que les autres.