

## TP CSS : Mise en place de règles de styles sur un site HTML

Votre entreprise est en train de développer un site de présentation de contenu de type portfolio, capable de donner un modèle de portfolio à remplir pour les futurs clients. Les maquettes ont été réalisées, et l'HTML qui servira de modèle a déjà été produit.

Votre travail est de concevoir une feuille de style afin que le site internet soit conforme à la maquette demandée par l'un des clients.

Le code HTML a déjà été validé par le service qualité de votre entreprise, vous ne pouvez donc pas le modifier (Celui-ci est utilisé comme template pour d'autres portfolio).

### Etape 1 : Mise en route

Quelques consignes vous sont données pour la réalisation de la feuille de style :

- Le code HTML ne doit en AUCUN CAS être modifié, sauf pour ajouter des classes et des id aux éléments.
- Le site doit pouvoir évoluer vers un design responsive (future adaptation pour des terminaux mobiles).
- La police souhaitée par le client est la suivante : Segoe UI
- Afin de respecter les règles de qualité de votre entreprise, vous préférerez exprimer les tailles en 'em' plutôt qu'en 'px'

On vous précise également que l'on souhaite un rendu fidèle à la maquette, mais que vous avez une marge de manœuvre en ce qui concerne la taille des éléments (on ne viendra pas vous embêter pour une poignée de pixels !).

Bien, mettons-nous au travail !

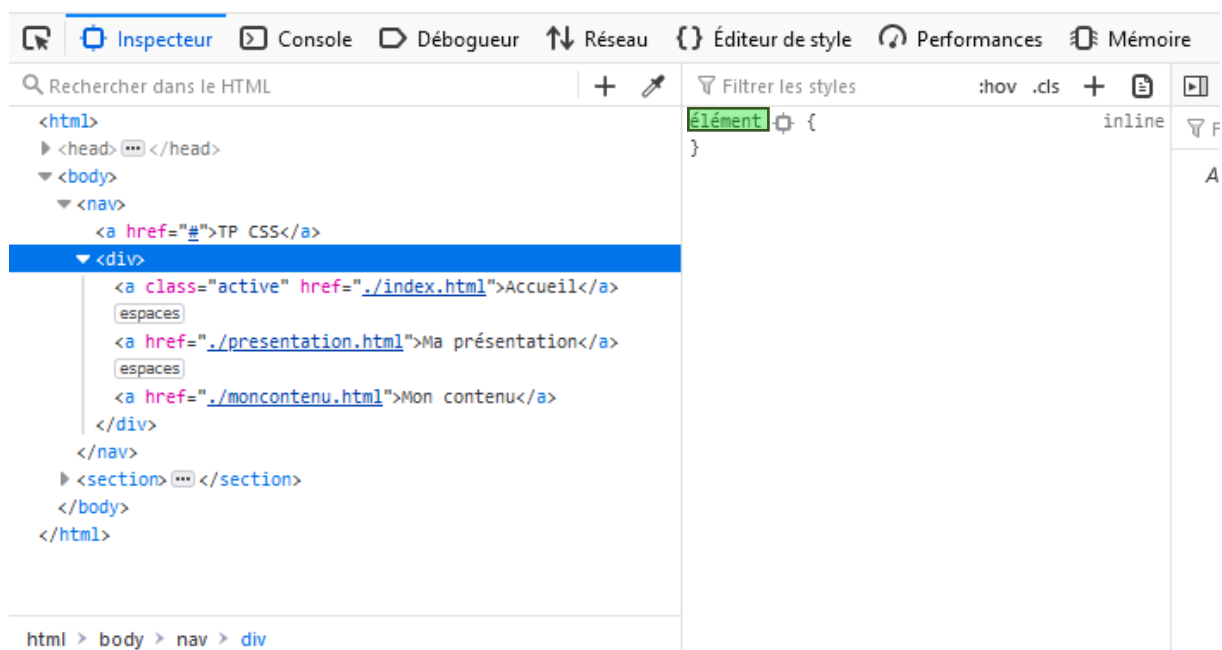
- 1) Prenez un instant pour examiner les spécifications et l'HTML, ainsi que son rendu sur le navigateur. Quelles sont vos remarques ?
- 2) Créez la feuille de style 'tpcss.css'.

Avant de continuer, prenez également un moment pour vous familiariser avec l'inspecteur de votre navigateur :

- Sous Firefox : Clic droit sur la page => Examiner l'élément
- Sous Chrome : Clic droit sur la page => Inspecter

Notez que pour la suite de ce TP, les exemples avec l'inspecteur utiliseront celui de Firefox, mais tous les navigateurs modernes possèdent des inspecteurs, et peuvent accomplir le même travail.

## Bienvenue sur mon site !



Dans le cadre de ce TP, seul l'onglet inspecteur nous sera utile, mais les autres ont également des fonctions très utiles pour nous autres développeurs !

Lorsque vous cliquez sur un des éléments de votre page (sur la page, ou l'arborescence HTML dans l'inspecteur), les règles CSS qui s'appliquent à cet élément s'affichent à droite de l'arborescence HTML de l'inspecteur.

- 3) Sélectionnez un élément `<a>` dans le `<nav>` de la page présentation, puis cliquez sur **élément** dans l'inspecteur. Entrez ensuite la règle 'font-size:30px'. Que se passe-t-il ?
- 4) Cliquez sur la case à cocher qui est apparue à côté de la nouvelle règle. Que constatez-vous ?
- 5) Cliquez droit sur **élément**, puis choisissez 'Copier la règle'. Collez ensuite le résultat sur un éditeur de texte. Que constatez-vous ?

L'inspecteur est un outil très pratique de développement web, et il est constamment utilisé par ces derniers. N'hésitez pas à vous en servir pour tester si vos règles sont appliquées, si vous voulez tenter une modification sans enregistrer, ou vérifiez l'effet d'une règle.

## Etape 2 : La page d'accueil

Bien, commençons par le commencement : récupérons les couleurs du thème de la maquette.

Si vous essayez de deviner, vous allez vous planter ! Utilisons donc une pipette !

- 1) En utilisant le site <https://imagecolorpicker.com/fr> , récupérez le code RGB (dans le format que vous souhaitez) des éléments suivants :
  - a. Le fond du contenu
  - b. La couleur du menu
  - c. La couleur du texte des menus et des contenus
  - d. La couleur du texte de la page Ma Présentation

Bon, il s'agit maintenant d'appliquer les couleurs et la police sur la page d'accueil, et comme il s'agit de propriétés générales, utilisons des sélecteurs d'attributs.

- 2) Appliquer les couleurs et la police en utilisant uniquement les sélecteurs d'attributs suivants dans votre feuille CSS :
  - a. body
  - b. nav
  - c. h1
  - d. a

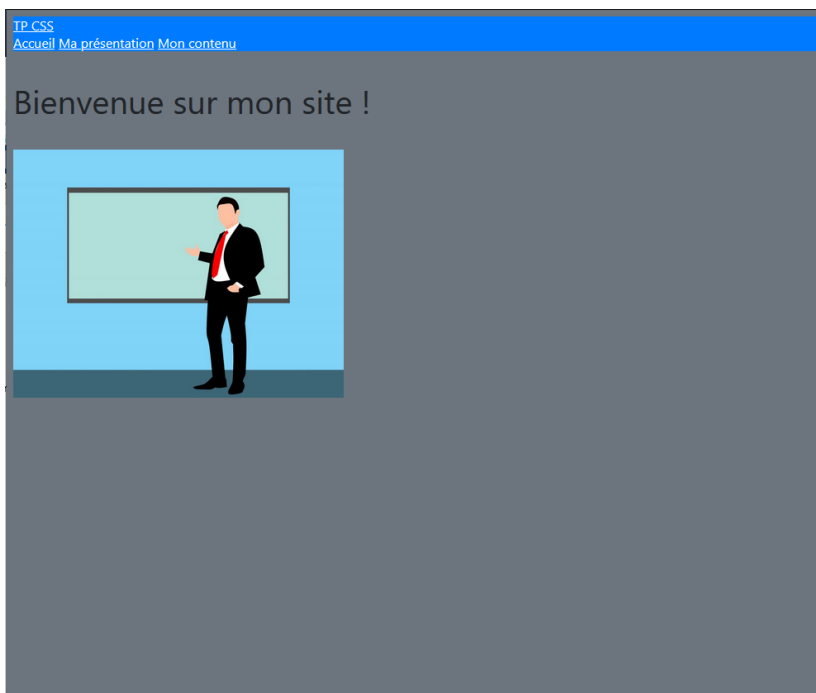
Documentation d'aide pour cette question (Elle provient du site de Mozilla, mais convient à tous les navigateurs récents) :

[Comment modifier la couleur de fond des éléments](#)

[Comment changer les propriétés des textes](#)

[Comment changer la couleur des textes](#)

Vous devez arriver à un résultat ressemblant à cela :



Bon, c'est un début, mais il y a tout un tas de problèmes :

- Il y a des bordures autour de l'écran
- Le menu est sur deux lignes, et ses éléments ne sont pas correctement espacés/dimensionnés/alignés
- Le contenu de la page n'est pas centré

Eh bien traitons les ! Commençons par les bordures. Cherchons le coupable, utilisons l'inspecteur.

- 3) Inspectez le `<body>` de votre page. Les bordures sont-elles démarquées de manière spécifique par votre inspecteur ?

Il est temps de parler de `margin` et de `padding` ! Pour faire simple :

- Les propriétés `margin` permettent de créer de l'espace supplémentaire entre l'élément touché et les autres éléments extérieurs.
- Les propriétés `padding` permettent de créer de l'espace entre l'élément et les éléments qu'il contient.

Documentation : les [Margin](#) et les [Padding](#)

N'hésitez pas à jouer avec la démo présente sur ces pages pour bien comprendre le concept.

Notre `<body>` a donc un `margin` natif de 8. Le fourbe !

- 4) Ecrivez une règle CSS forçant la `margin` du `<body>` à 0.

Les bordures devraient avoir disparues !

Attaquons maintenant le menu, et par la même occasion un gros morceau, le `display flex`.

Documentation : [Le Display Flex](#)

Prenez un moment pour bien lire cette documentation, elle est très importante, car elle va définir l'agencement de toutes nos pages !

- 5) En appliquant le `display flex`, faites-en sorte d'avoir le menu sur une seule ligne.



Petits rappels de CSS :

- `nav a` => Selecteur touchant tous les `<a>` a l'intérieur du `<nav>`.
- `nav div a` => Selecteur touchant tous les `<a>` dans des `<div>`, eux-mêmes dans le `<nav>`.
- `nav > a` => Selecteur qui touche uniquement les `<a>` qui sont directement fils de `<nav>`.

Donc, si vous avez bien suivi, répondez aux questions suivantes :

6) Combien d'éléments touchent les sélecteurs suivants dans la page d'accueil :

- `nav a`
- `nav div`
- `nav > div`
- `nav div a`
- `nav > div > a`
- `nav > a`

Réponses (Enlever le surlignage noir)

- 4 (Tous les `a` dans `<nav>`)
- 1 (Le `<div>` dans `<nav>`)
- 1 (Toujours le même `<div>`)
- 3 (Seulement les 3 `<a>` dans le `<div>` de `<nav>`)
- 3 (Même chose que la précédente)
- 1 (Seulement le `<a>` directement fils de `<nav>`)

Bon, maintenant que vous êtes incollable en selecteurs :

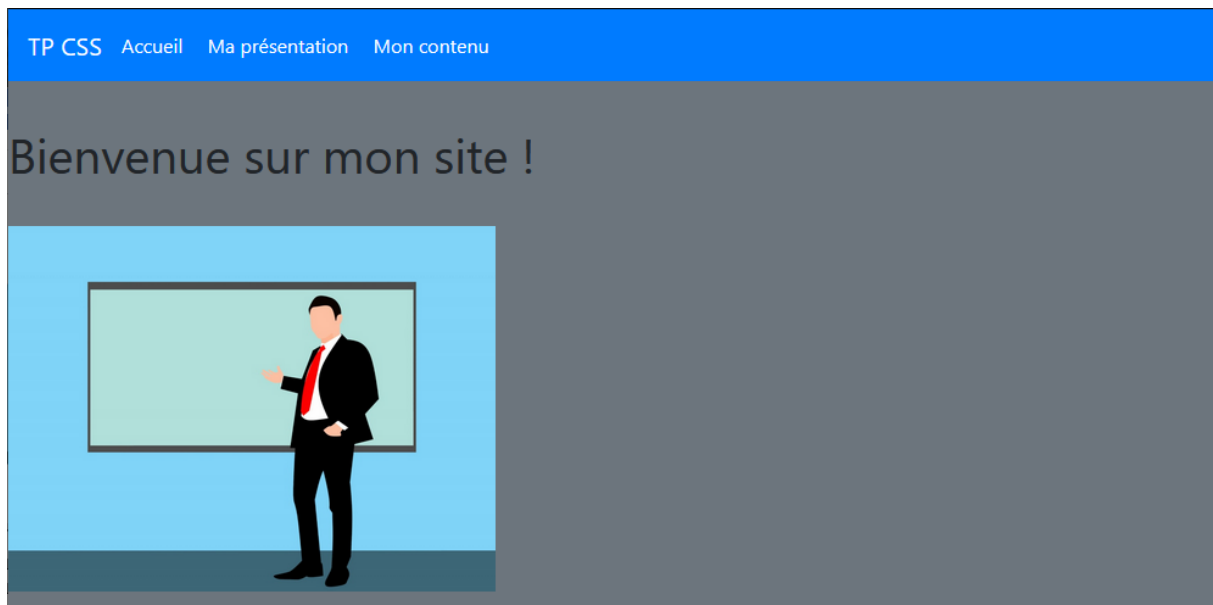
- 7) En utilisant astucieusement les sélecteurs, ainsi que les propriétés d'espacement (margin et padding) et de texte, donnez l'aspect suivant au menu (ignorez l'alignement vertical des items pour l'instant).



Bon, deux détails pour le menu subsistent :

- Les liens sont soulignés
- L'alignement vertical

- 8) Utilisez les propriétés du flex pour aligner verticalement les éléments du menu. (Indice : Vous devez aligner les éléments selon l'axe secondaire)
- 9) En utilisant la propriété text-decoration, supprimer le soulignement des liens.



Bon, et bien le menu c'est bon ! Nous verrons les quelques éléments manquants plus tard !

Maintenant, le contenu.

- 10) En utilisant un display :flex, centrez le contenu de la page d'accueil.

Indice 1 : Cette fois ci, il vous faut un flex vertical ! Il y a donc une propriété par défaut à changer.

Indice 2 : Il s'agit donc de la propriété flex-direction.

Indice 3 : Il faut ensuite aligner le contenu de ce flex selon l'axe horizontal, donc l'axe secondaire (puisque l'axe principal est vertical)



Et bien ma foi, le travail m'a l'air accompli !

## Etape 3 : La page présentation, ou comment apprécier les règles bien faites

Si vous n'avez pas utilisé des règles portant sur les id et les classes, vous devriez avoir une plaisante surprise en chargeant la page de présentation :



En effet, si vous avez évité d'utiliser des selecteurs de classes et d'ids, comme la page utilise des éléments structurants (<nav>, <section>), cela permet d'appliquer des règles générales bien pratiques qui nous mâchent bien le travail.

Il reste cependant le texte à traiter :

- L'espacement entre les lignes n'est pas le même
- Le texte n'est pas correctement positionné par rapport aux bords

Eh bien, réglons cela. Nous pourrions utiliser une règle générale sur la balise <P>, mais cela risque d'être problématique sur le long terme. Il faut donc trouver une solution.

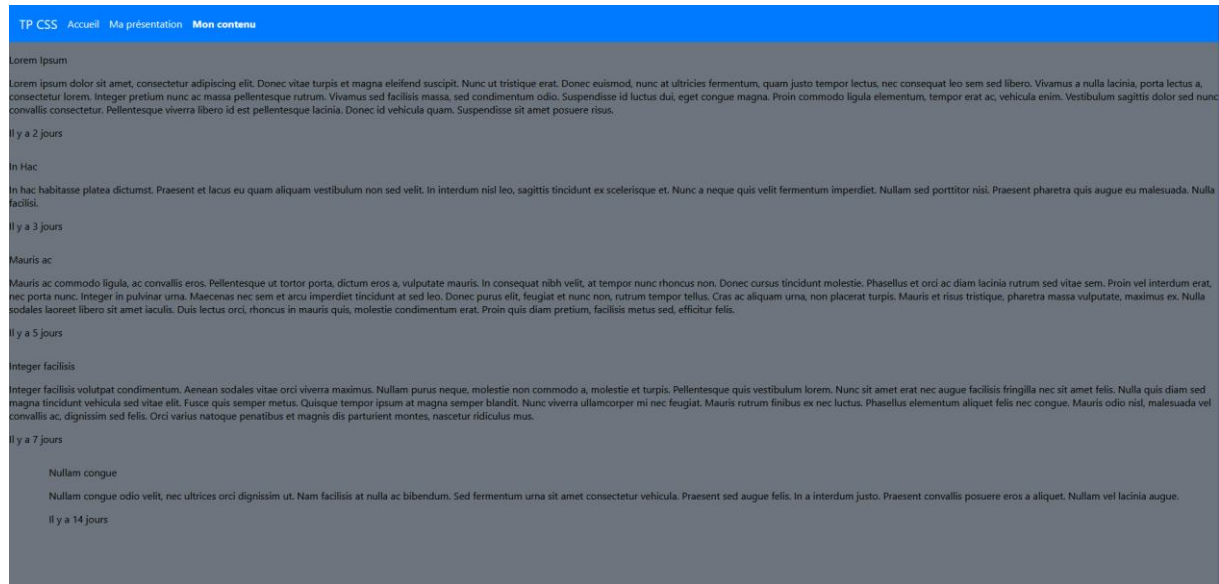
- 11) En utilisant tout ce que vous avez appris jusque-là, reproduisez la page 'Ma Présentation' pour qu'elle soit conforme à sa spécification.

A noter qu'il est possible de faire cela avec une seule règle.



## Etape 4 : Mon contenu

Autant la page Présentation était une histoire rapide, autant il y a du travail pour la page ‘Mon Contenu’ :

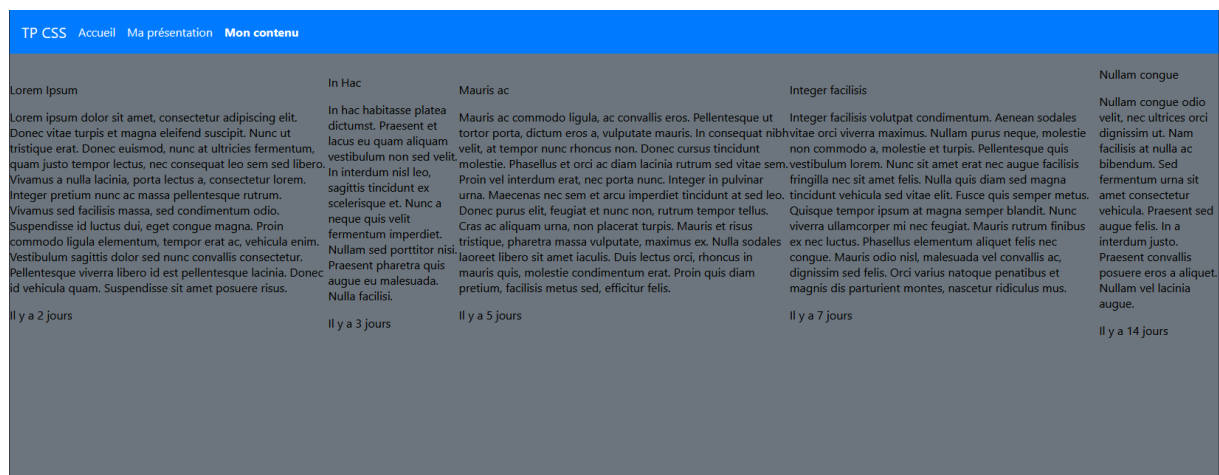


12) Au premier coup d’œil, quel est le problème en ce qui concerne l’alignement sur la page ?

Réponse : L’axe principal est vertical, alors que sur la maquette, l’axe principal est horizontal. Il faut modifier la propriété flex-direction.

Corrigeons ce comportement en le surchargeant à l’aide d’une classe !

13) En utilisant une classe, surchargez le comportement du sélecteur plus général.



Ok, bon ... ça l’effectue pour le moment ^^ . Enfin, il ne devrait y avoir que 4 éléments par ligne ! Utilisons la propriété [flex](#) pour régler le problème ! C’est la possibilité de fixer la taille d’un élément qui nous intéresse ici !

14) En utilisant la propriété flex, faites-en sorte que chaque élément du <section> fasse 25% de la taille totale.

Ah mais non !!!!!

TP CSS   Accueil   Ma présentation   Mon contenu		
<p>consectetur adipiscing elit. eleifend suscipit. Nunc ut id, nunc at ultricies</p> <p>In Hac</p> <p>In hac habitasse platea dictumst. Praesent et lacus eu quam aliquam vestibulum non sed velit. In interdum nisl leo, sagittis tincidunt ex scelerisque et. Nunc a neque quis velit fermentum imperdiet. Nullam sed porttitor nisi. Praesent pharetra quis augue eu malesuada. Nulla facilisi.</p> <p>Il y a 3 jours</p>	<p>Mauris ac</p> <p>Mauris ac commodo ligula, ac convallis eros. Pellentesque ut tortor porta, dictum eros a, vulputate mauris. In consequat nibh velit, at tempor nunc rhoncus non. Donec cursus tincidunt molestie.</p> <p>Phasellus et orci ac diam lacinia rutrum sed vitae sem. Proin vel interdum erat, nec porta nunc. Integer in pulvinar urna. Maecenas nec sem et arcu imperdiet tincidunt at sed leo. Donec purus elit, feugiat et nunc non, rutrum tempor tellus. Cras ac aliquam urna, non placerat turpis. Mauris et risus tristique, pharetra massa vulputate, maximus ex. Nulla sodales laoreet libero sit amet iaculis. Duis lectus orci, rhoncus in mauris quis, molestie condimentum erat. Proin quis diam pretium, facilisis metus sed, efficitur felis.</p> <p>Il y a 5 jours</p>	<p>Integer facilisis</p> <p>Integer facilisis volutpat condimentum. Aenean sodales vitae orci viverra maximus. Nullam purus neque, molestie non commodo a, molestie et turpis.</p> <p>Nullam congue</p> <p>Pellentesque quis vestibulum lorem. Nunc sit amet erat nec augue facilisis fringilla nec sit amet felis. Nulla quis diam sed magna tincidunt vehicula sed vitae elit. Fusce quis semper metus. Quisque tempor ipsum at magna feugiat. Mauris rutrum finibus ex nec luctus. Phasellus elementum aliquet felis nec congue. Mauris odio nisl, malesuada vel convallis ac, dignissim sed felis. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.</p> <p>Il y a 14 jours</p> <p>Il y a 7 jours</p>

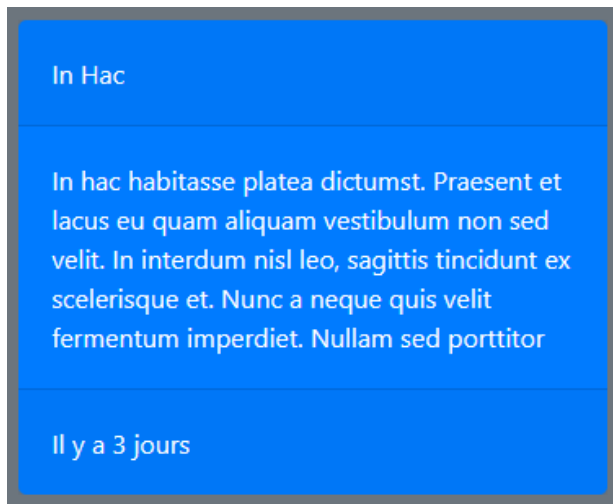
Une barre de défilement horizontale ! Et puis quoi encore ?

Le flex ne retourne pas automatiquement à la ligne lorsqu'il est à court de place, il faut le lui indiquer avec l'une de ses propriétés.

15) En utilisant la [documentation](#) sur le display : flex, forcez le retour à la ligne.

TP CSS   Accueil   Ma présentation   Mon contenu		
<p>Lorem Ipsum</p> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec vitae turpis et magna eleifend suscipit. Nunc ut tristique erat. Donec euismod, nunc at ultricies fermentum, quam justo tempor lectus, nec consequat leo sem sed libero. Vivamus a nulla lacinia, porta lectus a, consectetur lorem. Integer pretium nunc ac massa pellentesque rutrum. Vivamus sed facilisis massa, sed condimentum odio. Suspendisse id luctus dui, eget congue magna. Proin commodo ligula elementum, tempor erat ac, vehicula enim. Vestibulum sagittis dolor sed nunc convallis consectetur. Pellentesque viverra libero id est pellentesque lacinia. Donec id vehicula quam. Suspendisse sit amet posuere risus.</p> <p>Il y a 2 jours</p> <p>Nullam congue</p> <p>Nullam congue odio velit, nec ultrices orci dignissim ut. Nam facilisis at nulla ac bibendum. Sed fermentum urna sit amet consectetur vehicula. Praesent sed augue felis. In a interdum justo. Praesent convallis posuere eros a aliquet. Nullam vel lacinia augue.</p> <p>Il y a 14 jours</p>	<p>Mauris ac</p> <p>Mauris ac commodo ligula, ac convallis eros. Pellentesque ut tortor porta, dictum eros a, vulputate mauris. In consequat nibh velit, at tempor nunc rhoncus non. Donec cursus tincidunt molestie. Phasellus et orci ac diam lacinia rutrum sed vitae sem. Proin vel interdum erat, nec porta nunc. Integer in pulvinar urna. Maecenas nec sem et arcu imperdiet tincidunt at sed leo. Donec purus elit, feugiat et nunc non, rutrum tempor tellus. Cras ac aliquam urna, non placerat turpis. Mauris et risus tristique, pharetra massa vulputate, maximus ex. Nulla sodales laoreet libero sit amet iaculis. Duis lectus orci, rhoncus in mauris quis, molestie condimentum erat. Proin quis diam pretium, facilisis metus sed, efficitur felis.</p> <p>Il y a 5 jours</p>	<p>Integer facilisis</p> <p>Integer facilisis volutpat condimentum. Aenean sodales vitae orci viverra maximus. Nullam purus neque, molestie non commodo a, molestie et turpis.</p> <p>Pellentesque quis vestibulum lorem. Nunc sit amet erat nec augue facilisis fringilla nec sit amet felis. Nulla quis diam sed magna tincidunt vehicula sed vitae elit. Fusce quis semper metus. Quisque tempor ipsum at magna feugiat. Mauris rutrum finibus ex nec luctus. Phasellus elementum aliquet felis nec congue. Mauris odio nisl, malesuada vel convallis ac, dignissim sed felis. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.</p> <p>Il y a 7 jours</p>

Pour la suite du sujet, j'appellerai cet élément une carte :

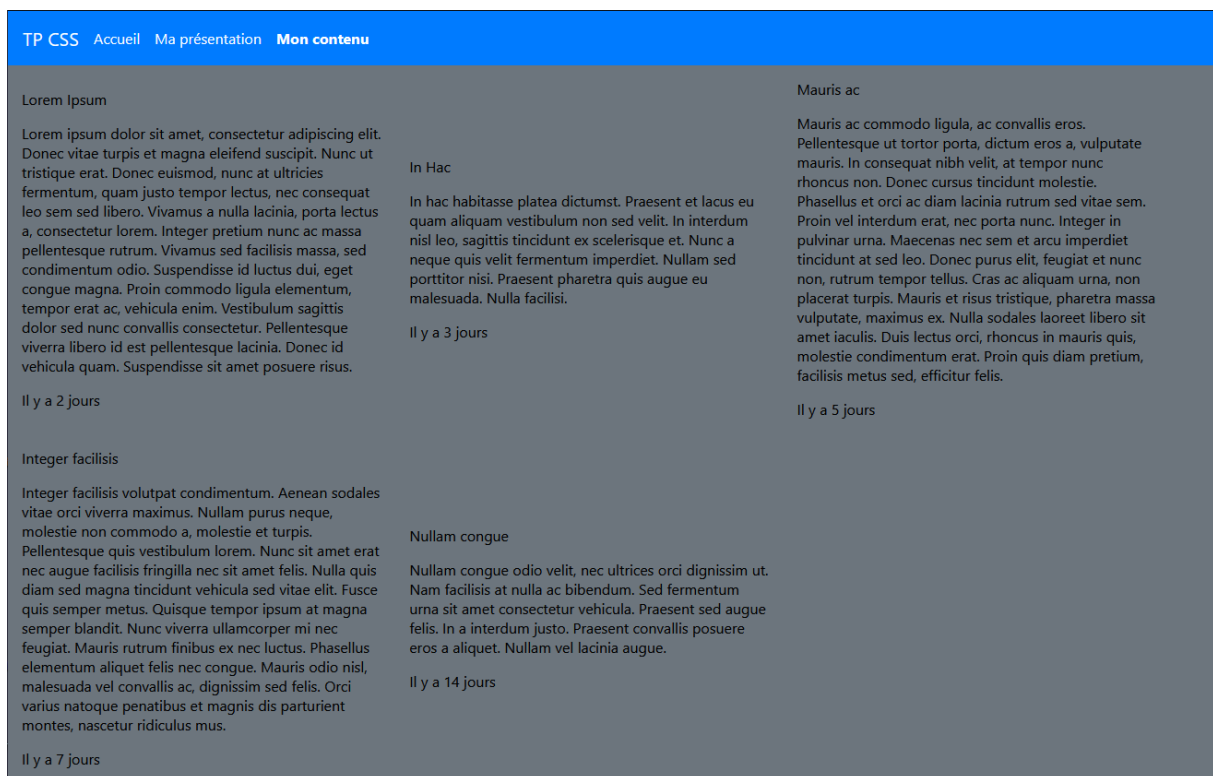


Il y a donc 5 cartes présentes dans le code HTML.

Essayons de mettre un peu d'espace entre les différentes cartes en utilisant le padding.

16) Ajoutez du padding (vertical et horizontal) entre chaque carte.

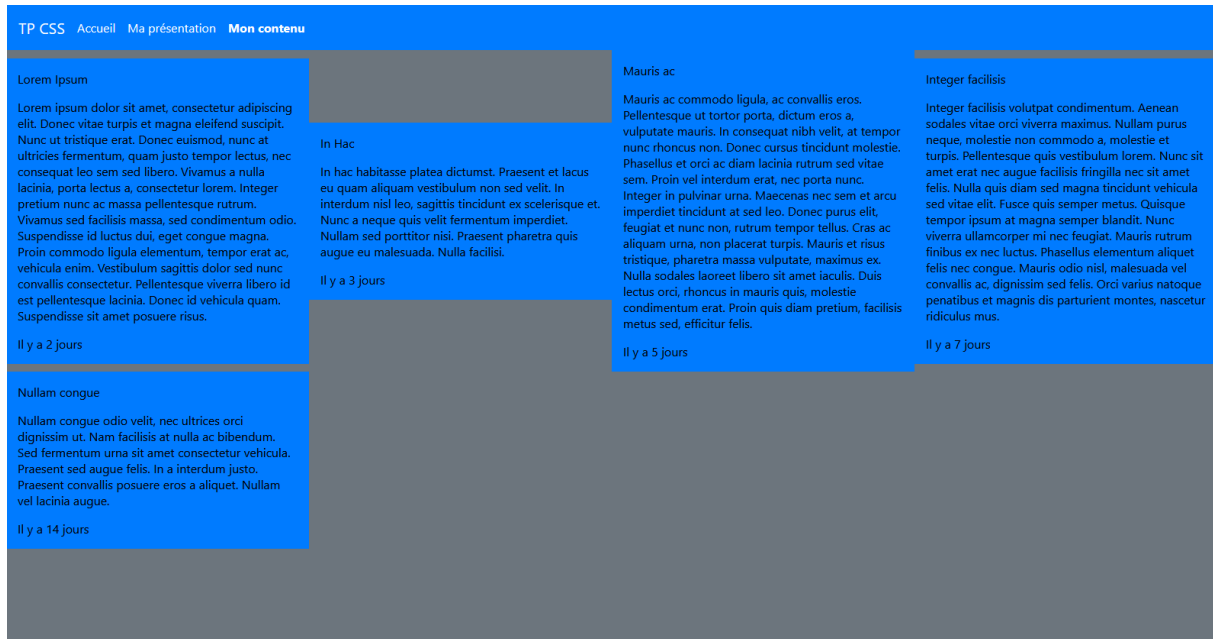
Et là, un problème intéressant arrive !



Il y a un retour à la ligne ! Eh oui, comme nous avons rajouté du padding, les éléments font désormais 25% de la taille de l'écran + le padding horizontal que vous avez rajouté ! Donc le retour à la ligne s'opère. Heureusement, CSS nous apporte une solution simple à ce problème : [Le box-sizing](#).

Ce dernier nous permet, lors du dimensionnement de nos éléments, de prendre en compte le padding dans le rendu final pour ne pas dépasser de l'élément parent.

- 17) En utilisant la propriété box-sizing, permettre l’affichage de 4 éléments par ligne.
- 18) Ajoutez la couleur de fond de chaque carte.



Bon, ça commence à ressembler à quelque chose ! Mais pas mal de problèmes encore :

- La couleur du texte
- L’espacement entre chaque carte
- L’alignement des cartes
- La taille des cartes

Avant de nous repencher sur l’espacement (car pour rappel, nous avons rajouté du padding), réglons le problème simple de la couleur.

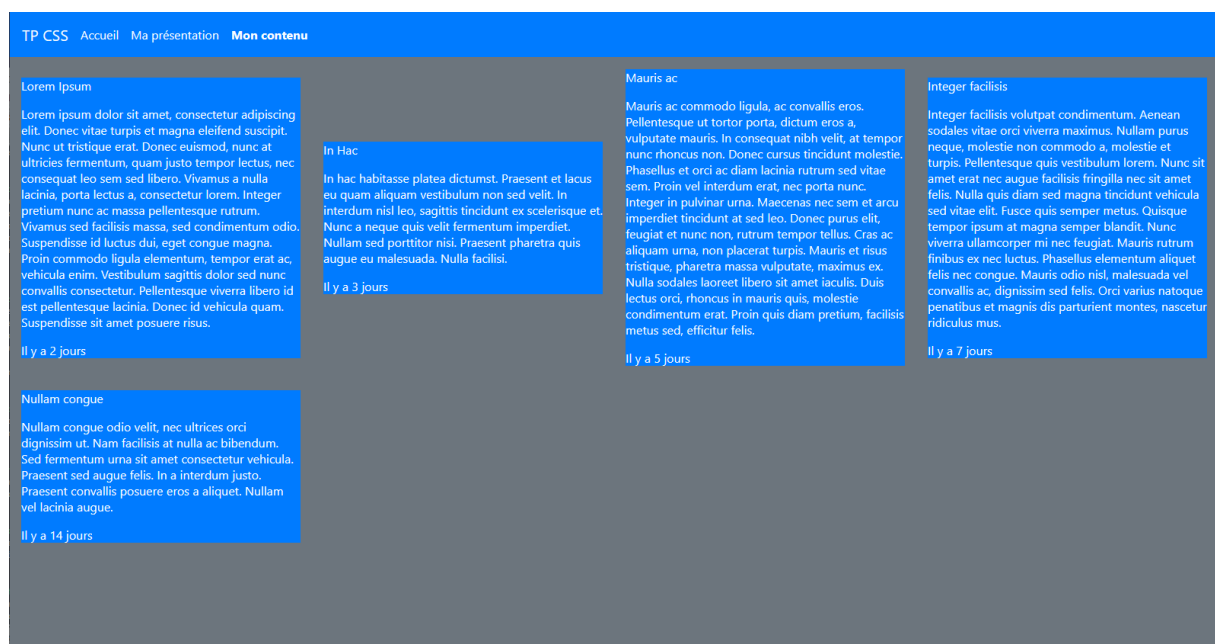
- 19) En utilisant le principe de la surcharge, changer la couleur du texte pour correspondre à la maquette.

Bon, l’espacement. Oui, nous avons mis du padding, mais le padding, c’est l’espacement entre l’élément et ses éléments intérieurs ! Et on ne peut pas utiliser de margin, sinon le flex-wrap ne fonctionne plus !

Heureusement, il n’y a qu’à appliquer la couleur de fond des cartes à l’un des éléments internes aux cartes pour donner l’illusion d’un espacement.

- 20) Appliquez le fond à un élément qui permet de créer un espace visuel entre chaque carte.
- 21) Ajouter une espace entre le menu et les cartes, ainsi qu’entre chaque ligne de carte.

Indice (22) : Utiliser la propriété margin-top



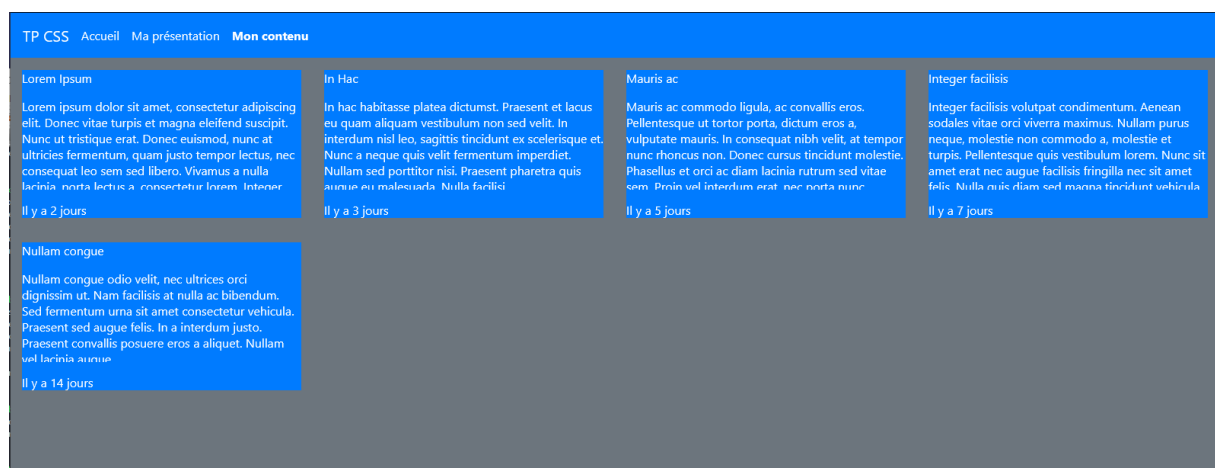
On y arrive ^^, mais il est temps de régler un problème que nous avons laissé depuis longtemps de coté. Dans la maquette, le contenu de chaque carte est limitée à une certaine hauteur : 120px. Appliquons cette règle !

## 22) Fixer la taille du texte (hors titre et pied) à 120px.



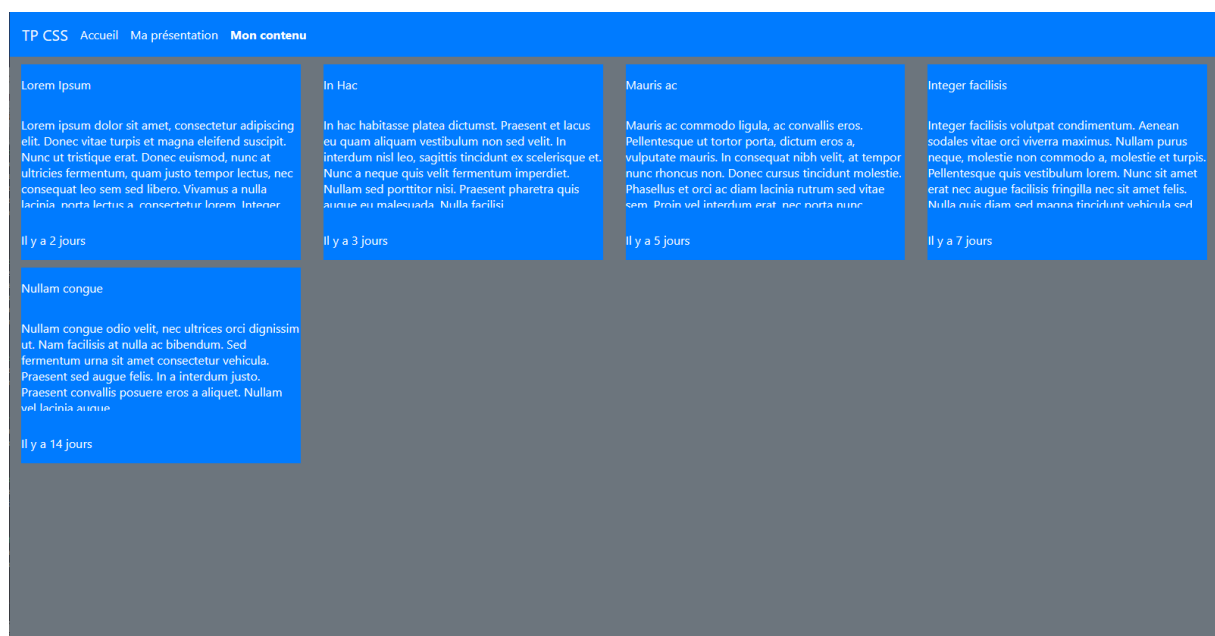
Bon, les résultats sont mitigés, mais explicables ! Par défaut, en HTML, quand un élément manque de place pour s'afficher, il grandit pour s'adapter à son contenu, mais comme nous avons fixé une taille insuffisante, le contenu dépasse, on dit qu'il est en [overflow](#).

## 23) Modifiez le comportement de l'overflow pour correspondre à la maquette.



Ah, on se rapproche. Bon, pour l’alignement des éléments, utilisons du flex !

24) Appliquez un display : flex à l’élément portant le fond bleu, et paramétrez-le de manière à obtenir le résultat suivant :

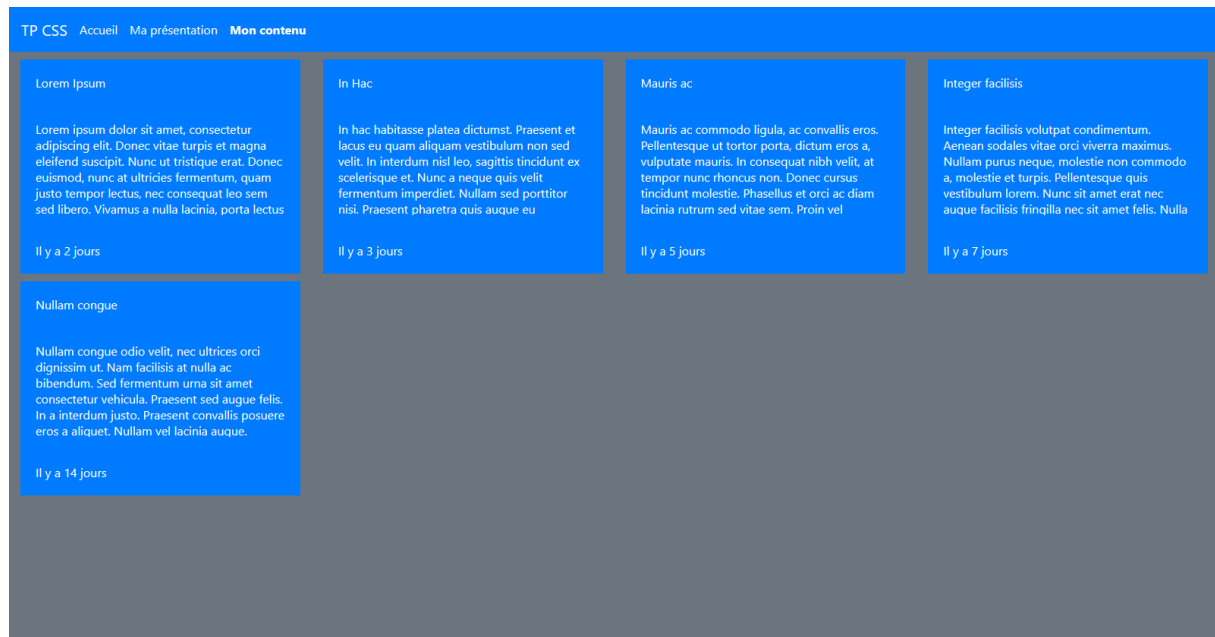


Bon, il est temps de décoller notre texte du bord de nos cartes !

25) Appliquez une/des règles CSS permettant le résultat suivant :

Indice 1 : [REDACTED]

Indice 2 : [REDACTED]



Nous avons parcouru beaucoup de chemin. Il ne nous reste plus qu'à peaufiner ce joli travail !

- 26) Appliquer un espacement entre les lignes de chaque paragraphe
- 27) Créez des arrondis aux coins des cartes

Documentation :

### [Arrondir les coins](#)

Il ne nous reste que les contours légèrement plus foncés des entêtes et des pieds de carte. Pour cela, je vais vous donner une astuce : la transparence !

On peut représenter une couleur selon le format rgba (Rouge, Vert, Bleu, Alpha). Le canal alpha gère la transparence :

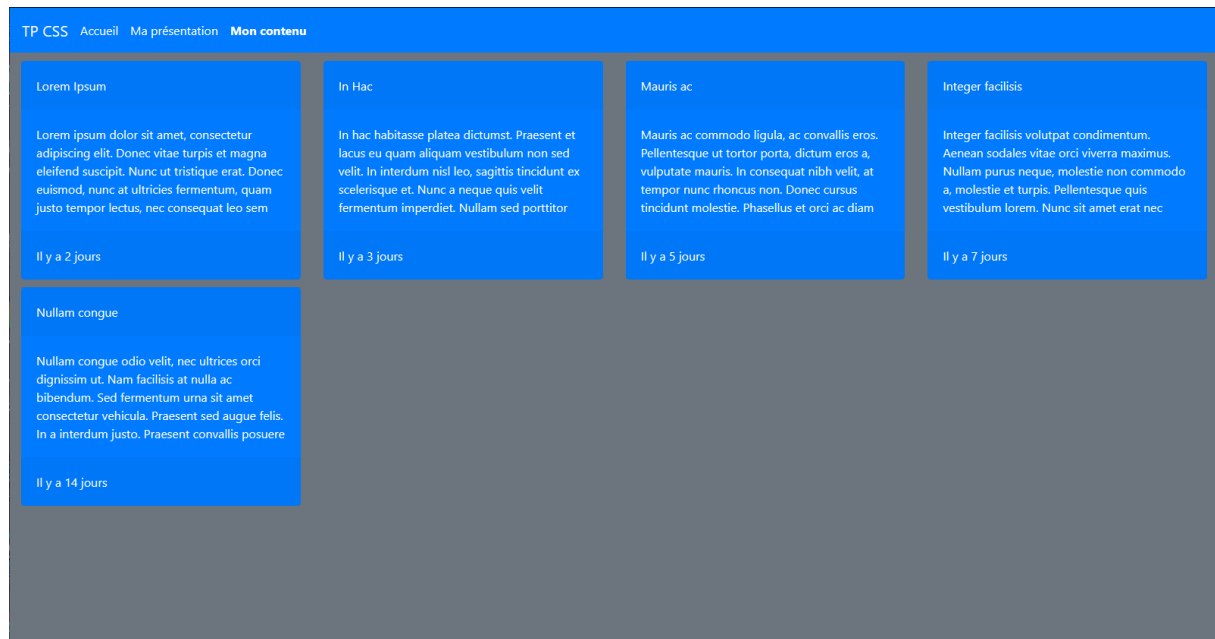
- A 1, il est complètement opaque
- A 0, il est complètement transparent

L'idée est de créer un fond gris (en utilisant la propriété background-color) suffisamment transparent pour donner l'effet que la couleur est un peu plus terne.

- 28) Créez le petit effet foncé sur les pieds et les entêtes des cartes.

Indice 1 : Les éléments sur lesquels vous devez appliquer les règles sont les `<div>` qui contiennent les entêtes et les footers. N'hésitez pas à créer une classe appliquée à ces éléments.

Indice 2 : La valeur du canal alpha (pour la transparence) est 0.03



Il ne reste plus que la séparation nette entre les éléments de la carte, et nous sommes bon ! Pour cela, utilisons des [bordures](#) !

29) En utilisant des bordures, créez la séparation entre les éléments de la carte.

Indice 1 : 

Indice 2 :   


Félicitations ! On dirait la maquette maintenant !

## Etape 4 : Les spécificités du menu

Il y a quelques éléments du menu qui ne sont pas encore gérés !

- La page active du menu est écrite en gras.
- Le menu doit rester affiché sur la page même lorsque l'on défile une page.
- Lorsque l'on passe la souris sur un élément du menu, celui-ci passe progressivement en gras.

Allons-y progressivement, tout d'abord, la page active !

30) En utilisant les propriétés CSS des textes, mettre la page courante dans le menu en gras.

Indice 1 :   


Maintenant, le menu fixe ! La meilleure page pour tester cela est la page présentation.

31) En utilisant les différentes [positions](#), faites-en sorte que le menu reste toujours au sommet de l'écran.



Il se peut que, selon la solution que vous adoptez, le menu s’affiche par-dessus la page, même quand vous êtes au sommet de celle-ci. Si vous rencontrez ce problème, il vous suffit de mesurer la hauteur de votre menu, et de rajouter une margin adéquate à votre contenu.

Bien, maintenant, il s’agit d’avoir les items du menu qui se mettent en gras lorsqu’on les survole !

Pour cela, il faut faire appel aux [pseudo-classes](#) CSS.

32) En utilisant les pseudos-classes, faites en sorte que le texte se mette en gras lorsque l’on le survole.

Bien, il ne reste plus qu’a faire en sorte que cela arrive progressivement, faisons un tour du côté des animations CSS ! Le sujet est vaste, donc nous n’utiliserons que les [transitions](#) dans ce TP.

33) En utilisant la transition, faites-en sorte que la mise en gras soit progressive.

Eh bien, en voilà un beau menu !

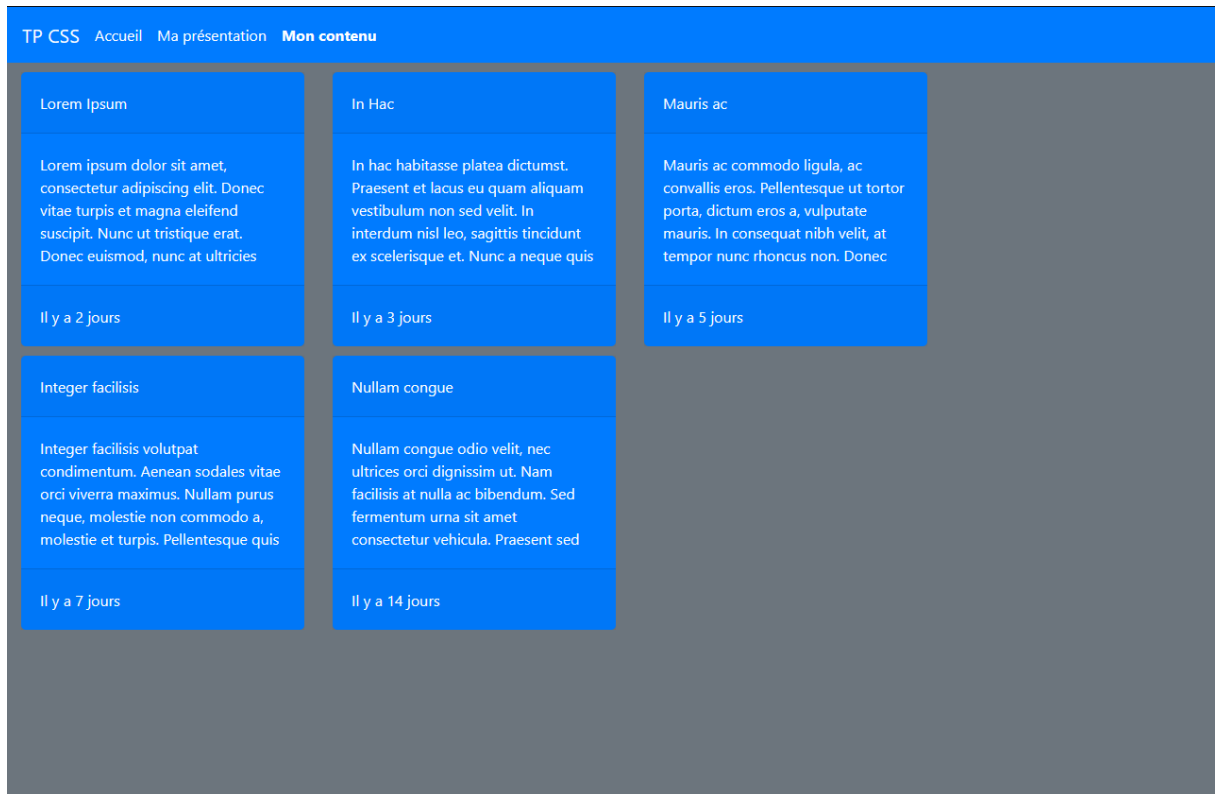
## Etape 5 : Les médias queries, ou comment gérer le responsive

Bon, si vous avez tenté de réduire la taille de votre fenêtre dans la page « Mon contenu », vous aurez pu vous apercevoir d’un problème :



Bon, réglons le problème rapidement :

34) Imposez à vos cartes une largeur minimale de 300px.



Ce n'est pas très esthétique ! Il faudrait qu'une fois que l'écran tombe en dessous d'une certaine taille, on puisse changer la propriété flex des cartes ...

Heureusement, CSS vous couvre avec les [medias queries](#) !

Le sujet est un peu feuillu, mais pour faire simple, les medias queries sont des conditions d'application de règles CSS. Par exemple :

```
.maClasse{
  color: white;
}

@media (max-width : 600px){
  .maClasse{
    color: blue;
  }
}
```

Considérons le code CSS suivant :

- Tant que mon écran à une largeur supérieure à 600px, la classe .maClasse qui s'applique est celle en **dehors** de la media query.
- Des que mon écran est réduit à une largeur inférieure à 600px, c'est la classe .maClasse à **l'intérieur** de la media query qui s'applique.

Les classes dans les media queries surchargent les comportements qui sont en dehors si elles s'appliquent (elles sont prioritaires).

Autre exemple (que vous pouvez tester sur le TP si vous le souhaitez) :

```
body{
  color : white
}

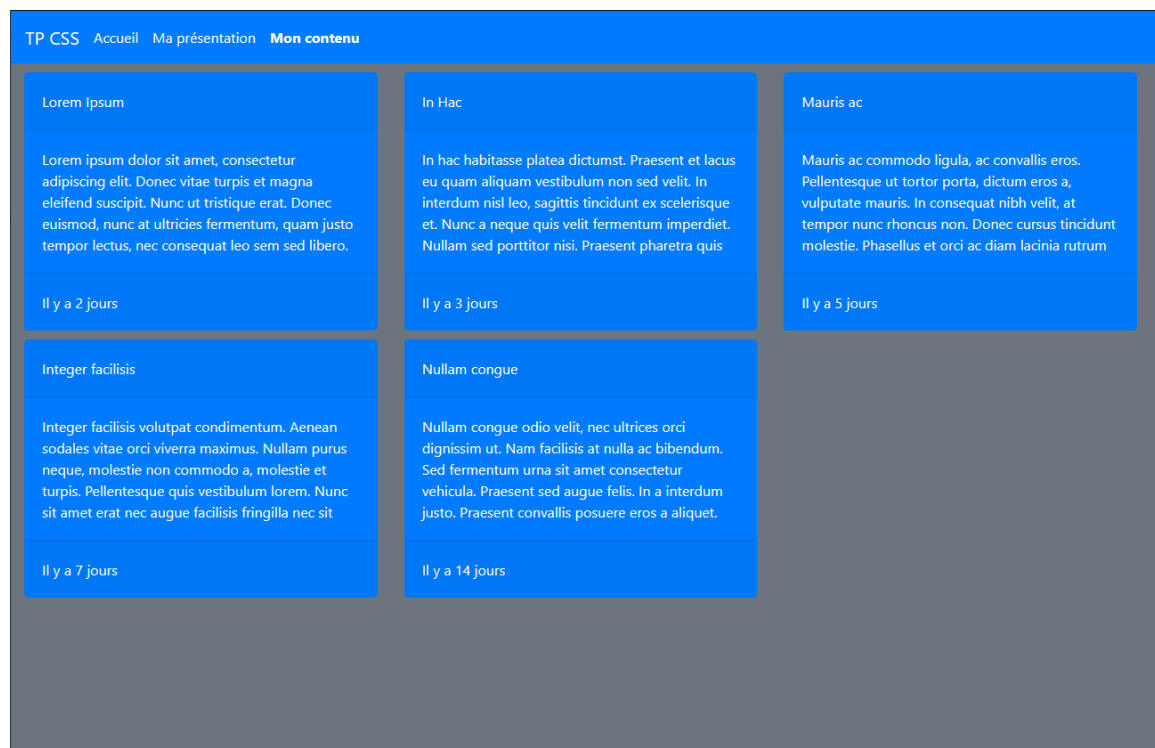
@media (min-width: 1200px) and (max-width : 1600px){
  body{
    color: blue;
  }
}
```

Ici :

- Tant que l'écran à une taille supérieure à 1600px, le texte sera blanc.
- Tant que l'écran à une taille entre 1200px et 1600px, le texte sera bleu.
- Tant que l'écran à une taille inférieure à 1200px, le texte redeviendra blanc.

N'hésitez pas à tester sur votre site.

35) En utilisant les medias queries, modifiez la propriété flex des cartes en fonction de la taille de l'écran pour obtenir un résultat comme ci-dessous :



Lorem Ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
 Donec vitae turpis et magna eleifend suscipit. Nunc ut  
 tristique erat. Donec euismod, nunc at ultricies  
 fermentum, quam justo tempor lectus, nec consequat  
 leo sem sed libero. Vivamus a nulla lacinia, porta lectus

Il y a 2 jours

In Hac

In hac habitasse platea dictumst. Praesent et lacus eu  
quam aliquam vestibulum non sed velit. In interdum nisl  
leo, sagittis tincidunt ex scelerisque et. Nunc a neque  
quis velit fermentum imperdiet. Nullam sed porttitor nisi.  
Praesent pharetra quis augue eu malesuada. Nulla

Il y a 3 jours

Mauris ac

Mauris ac commodo ligula, ac convallis eros.  
Pellentesque ut tortor porta, dictum eros a, vulputate  
mauris. In consequat nibh velit, at tempor nunc rhoncus  
non. Donec cursus tincidunt molestie. Phasellus et orci  
ac diam lacinia rutrum sed vitae sem. Proin vel interdum

Il y a 5 jours

## Integer faciliis

Integer facilisis volutpat condimentum. Aenean sodales vitae orci viverra maximus. Nullam purus neque, molestie non commodo a, molestie et turpis. Pellentesque quis vestibulum lorem. Nunc sit amet erat nec augue facilisis fringilla nec sit amet felis. Nulla quis

Il y a 7 jours

Nullam congue

Nullam congue odio velit, nec ultrices orci dignissim ut. Nam facilisis at nulla ac bibendum. Sed fermentum urna sit amet consectetur vehicula. Praesent sed augue felis.

Lorem Ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing  
 elit. Donec vitae turpis et magna eleifend suscipit.  
 Nunc ut tristique erat. Donec euismod, nunc at  
 ultricies fermentum, quam justo tempor lectus, nec  
 consequat leo sem sed libero. Vivamus a nulla

Il y a 2 jours

In Hac

In hac habitasse platea dictumst. Praesent et lacus eu quam aliquam vestibulum non sed velit. In interdum nisl leo, sagittis tincidunt ex scelerisque et. Nunc a neque quis velit fermentum imperdiet. Nullam sed porttitor nisi. Praesent pharetra quis

Il y a 3 jours

Mauris ac

Mauris ac commodo ligula, ac convallis eros.  
Pellentesque ut tortor porta, dictum eros a,  
vulputate mauris. In consequat nibh velit, at tempor

## Etape finale : Et maintenant ?

Vous venez de voir une bonne panoplie d'éléments CSS qui vous permettent de faire des sites au design sympathique ! Maintenant, il s'agit de redoubler d'astuce pour donner vie à votre vision !

Ce TP vous a aussi montré l'importance d'exploiter la documentation lorsque vous découvrez un langage. Et vous verrez que de simple recherches internet vous permettent de trouver rapidement votre bonheur :

[Comment mettre une couleur de fond css](#)

[Comment changer la taille du texte en css](#)

[Comment espacer les éléments en css](#)

Et ça marche encore mieux avec les recherches en anglais !

En conclusion, un petit ajout avec javascript !

En javascript, il est possible [d'activer/de désactiver une classe CSS lors d'une action de l'utilisateur](#) !

Ce n'est pas tout, mais cela permet déjà beaucoup de choses ! Comme la création d'un menu plus adapté lors de l'utilisation de petits écrans (smartphones), par exemple !