1. Objectifs du TP

- Utiliser le positionnement "flex" pour la réalisation d'un menu horizontal ;
- Utiliser le positionnement "flex" pour la réalisation d'un site simple ;
- Comprendre les positionnements "relative" et "absolute" ;
- Comprendre le positionnement "float".

Ce sujet est consultable en version PDF sur Chamilo.

---

2. Préparation et introduction

> ✎ À faire
>
> 1. Créer un répertoire TP4 dans le répertoire r1_02 ;
> 2. Se positionner dans ce répertoire ;
> 3. Recopier dans votre répertoire TP4 le contenu de l'archive.

Nous avons jusqu'à présent beaucoup parlé du positionnement Flexbox. Il en existe d'autres plus anciens, qui ont chacun un cas d'usage dédié. Dans ce TP nous allons approfondir le positionnement Flex et allons en découvrir d'autres.

3. Application du positionnement "flex" pour la réalisation d'un menu horizontal simple

Dans ce premier exercice, nous allons pratiquer le positionnement Flex pour la création d'un menu d'un moteur de recherche. La figure ci-dessous représente le rendu attendu. La case orange permet de voir l'effet désiré quand on passe sur un lien,

légende : rendu attendu

> ✎ À faire
>
> - Ouvrir le fichier moteur/index.html et examiner son contenu ;
> - Créer un fichier moteur/moteur.css ;
> - Appliquer les consignes suivantes.

Voici quelques indications:

- Vous utiliserez le positionnement "flex"
- Les couleurs sont: "white", "gray" et "tomato"
- La marge intérieure gauche qui est par défaut dans le ul doit être supprimée
- Le "petit point" des items li doit être supprimé
- Vous ajouterez une marge intérieure de 1em 2em 1em 2em aux éléments a
  Attention: vous remarquerez que les marges en haut et en bas ne sont pas prises en compte. Ceci est dû au fait que les éléments <a> sont des éléments en ligne. Pour prendre en compte ces marges, il faudra changer leur style d'affichage en modifiant l'attribut "display" et en affectant la valeur "block" ou "inline-block".
- Ajouter les règles de style qui conviennent pour a, a:hover (voir la figure ci-dessus)
- Finalisez!

> ⚠ Attention
>
> Vous remarquerez qu'on ne s'intéresse qu'au menu qui est dans la division nav.

3.1. Variante du design

Nous allons réaliser une variante du design dans laquelle nous allons appliquer une bordure arrondie avec une ombre portée. La figure ci-dessous représente le résultat attendu.

légende : Application de la bordure et de l'ombre portée.

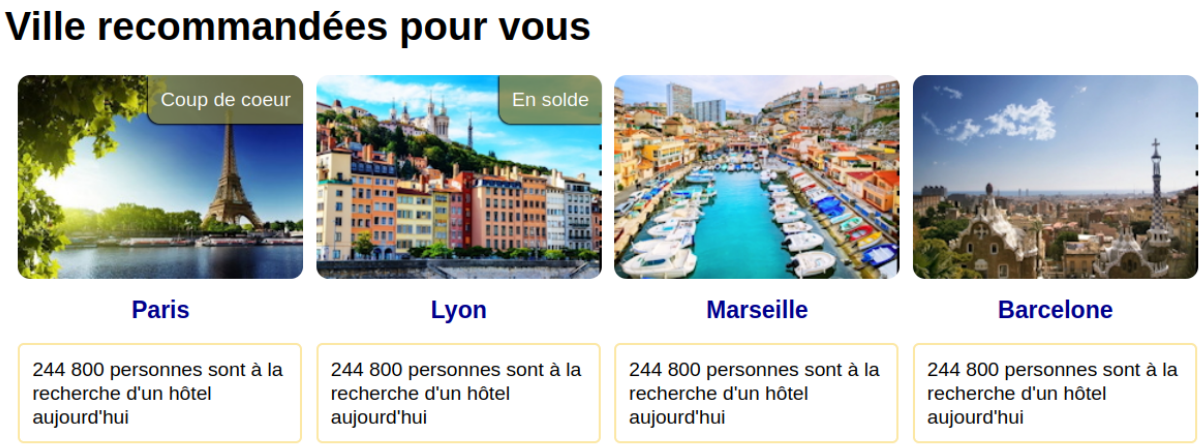
> ✎ À faire
>
> - Créer une copie du fichier index.html et moteur.css et travailler sur cette copie ;
> - Appliquer les consignes suivantes.

Voici quelques indications:

- Il faudra utiliser la propriété border-radius (et ses variantes - chercher de la documentation)
- Et bien sûr, quand la souris passera sur "Google" les coins seront arrondis également!
- Le sélecteur ":first-child" pourra vous être utile (documentation de w3schools)
- Pour réaliser une ombre portée, vous utiliserez la propriété box-shadow

4. Positionnement Flex, relatif, et absolu

Dans la suite, nous allons melanger le positionnement Flex avec le positionnement relatif et absolu. La figure ci-dessous illustre le résultat attendu. **On ignore dans un premier temps les badges s'affichant dans les coins superieurs de la première et de la seconde image.**



légende : Résultat attendu pour le site hôtel.

Commençons par quelques styles simples :

> ✎ À faire
>
> - Ouvrir les fichiers hotel/index.html et hotel/hotel.css et comprendre la structure du fichier html ;
> - Appliquer une police de caractères Arial, Helvetica ou à défaut sans-serif ;
> - Appliquer une couleur bleue aux titres des images et aligner le texte.

4.2. Application du positionnement Flex

Dans la suite, nous allons appliquer une positionnement Flex pour positionner les villes et leur description l'une à la suite de l'autre

> ✎ À faire
>
> - Appliquer le positionnement Flex pour obtenir le résultat de l'image ci-dessus ;
> - Utiliser les bonnes propriétés CSS pour définir que sur l'axe principal, l'espacement entre les items Flex est le même afin que les items occupent l'intégralité de la page ;
> - Prévoir la possibilité d'un retour automatique à la ligne si le navigateur ne laisse pas suffisament d'espace.

Nous nous intéressons aux paragraphes sous les images.

> ✎ À faire
>
> - Appliquer une marge intérieure de 10px sur tous les côtés ;
> - Ajouter une bordure de 2px de couleur #fce8a6 ;
>
> Quel problème graphique intervient ?

Nous avons vu lors du TP précédent que la largeur totale d'un bloc HTML est égal à la largeur de son contenu, à laquelle s'ajoutent la bordure et la marge intérieure. Une façon de remédier au problème d'alignement que vous pouvez appercevoir est de modifier la taille des paragraphes (vu au TP précédent), mais cela est fastidieux. En CSS, il existe néanmoins une propriété CSS intéressante, la propriété box-sizing. Cette dernière permet de changer la façon dont la largeur et la hauteur totales d'un élément sont calculées.

✎ À faire

- Appliquer la propriété CSS `box-sizing: border-box;` et visualiser le rendu ;
- Utiliser l'inspecteur, onglet *Layout* pour comprendre ce qu'il s'est passé.

4.3. Utilisation de la position "relative" combinée à la position "absolute"

Nous avons vu précédement la propriété display. Nous allons découvrir la propriété position. Cette propriété permet de gérer le type de positionnement d'une boîte "dans le flux" ou "hors du flux". Cet attribut a cinq valeurs possibles :

static (valeur par défaut) :
Le comportement naturel d'affichage d'un élément dans le flux

relative :
Ce positionnement réserve l'emplacement de l'élément dans le flux et le déplace en superposition de l'affichage (l'élément ne sort pas du flux; il reste dans le flux). Le déplacement se calcule relativement à la position basse (bottom), haute (top), droite ou gauche (right, left).

absolute :
Ce positionnement sort l'élément du flux, il n'occupe donc aucune place. Ainsi, ce positionnement se fait toujours en superposition des autres éléments qui eux restent dans le flux.

fixed et sticky :
Ces positionnements très particuliers permettent de conserver la même position (définie à l'aide des propriétés bottom, right, top, left) même dans le cas du défilement de la page. La différence entre les deux réside du fait que sticky occupe une place dans le flux et sa position est relative par rapport à sa position nominale dans le flux, tandis que fixed sort du flux et le calcul de sa position s'apparente à celle de la position absolute.

✎ À faire

- Ouvrir cette page afin de tester les différents types de positionnement ;
- Remplacer dans les exemples fournis `positon: fixed;` par `positon: sticky;` pour tester le dernier cas de positionnement.

Si les positions fixed et sticky sont utiles dans certains cas, ils restent très anecdotiques. La position relative ne présente que très peu d'intérêt, puisque elle occupe un espace dans le flux tout en se décalant par rapport à cette position tout en se superposant à d'autres éléments. Quant à la position absolute, celle-ci présentait un intérêt pour concevoir des styles occupant tout l'écran (sans avoir à défiler), mais cet avantage a disparu avec l'apparition du positionnement Flex permettant le même résultat.

En revanche, il est un cas où les positions relative et absolute présentent un intérêt majeur, et c'est dans leur combinaison que nous allons le trouver.

✎ À faire

- Ouvrir le fichier hotel/index2.html et observer l'ajout qui a été fait par rapport au fichier hotel/index.html
- Rajouter une règle dans le fichier hotel/hotel.css permettant de positionner les *badges* (éléments HTML `<span class="badge">`) en position absolue.

Que se passe-t'il ?

- Compléter la règle pour que le badge se retrouve en haut à droite de l'image en utilisant les propriétés :

```
top: 0px;
right: 0px;
```

Obtient-on le résultat attendu ?

- Définir les éléments `<div>` (conteneur des images et des labels) une position relative sans décalage.

Que constatez-vous ?

- Terminer le style des badges pour qu'ils aient :
  - Une couleur de fond semi-transparente `rgba(128, 128, 64, 0.8)` . Noter l'usage de la fonction *rgba* possédant un quatrième paramètre *alpha* ;
  - Une couleur de texte blanche ;
  - Une ombre portée **vers la gauche** ;
  - Deux angles sur quatre arrondis.

⚠ Trois règles fondamentales du positionnement absolu à retenir

1. Un élément positionné en "absolute" est positionné par rapport au parent ou ancêtre qui n'est pas en placement en flux (static) (c'est-à-dire au parent ou ancêtre positionné lui-même "absolute" ou en "relative"), ou alors par rapport aux limites de la zone de visualisation (la fenêtre du navigateur) s'il n'y a pas de parent ou d'ancêtre positionné. **D'où l'impotance d'avoir positionné les éléments `<div>` en relatif.**
2. Si un bloc passe de la position static (sa position par défaut) à la position absolute, il sort donc du flux. Si l'on ne précise rien d'autre, il vient se placer à l'endroit où il aurait dû être dans le flux. Toutefois, sa taille change : il perd une caractéristique majeure relative aux éléments dans un flux, qui est celle de recouvrir la totalité de la largeur disponible de l'élément parent.
3. Un élément ayant une position "absolute" ne bougera pas de sa position initiale tant que l'une des propriétés top, bottom, left ou right n'aura pas été précisée. Ce comportement est valable pour toutes les positions (sauf pour static, ces propriétés étant ignorées).

5. Le positionnement "float" (s'il vous reste du temps, sinon, passez au portfolio)

Le positionnement float permet de faire "flotter" un contenu à côté d'un autre. C'est un positionnement hors flux délicat à manipuler et à réserver pour certains usages. Il y a plusieurs années, on utilisait cette propriété pour faire de la mise en page, mais c'est obsolète maintenant avec l'apparition de méthodes plus efficaces (dont flex fait partie). Le positionnement float conserve aujourd'hui un intérêt qu'il est difficile de répliquer par d'autres mécanismes plus récents : celui "d'entourer" une image de texte. Nous allons voir ce cas juste après.

✎ À faire

- Ouvrir le fichier float/index.html: il contient une boîte aside (qui contient une image et un mot) et deux paragraphes p.
- Visualiser avec votre navigateur la page Web.
- Créer un fichier float/float.css qui donne une couleur sombre au fond des paragraphes et une couleur claire au texte.

L'image fait 150 pixels de haut (height). Pourtant la boîte aside a une hauteur de quelques pixels de plus. De combien de pixels exactement ?

Explication: ces quelques pixels supplémentaires correspondent au fait qu'une image est un objet "en ligne" comme les mots d'un texte, et l'alignement se fait sur la ligne de base des caractères. Un espace est donc automatiquement ajouté aux lettres des mots pour placer les jambages comme le bas du y et du g du mot Voyage.

✎ À faire

- Ajouter pour l'image l'attribut vertical-align: bottom; cela force l'image à s'aligner sur l'élément le plus bas de la ligne.
- Constater ainsi que la boîte aside fait bien 150px et que l'image n'est plus alignée sur la ligne de base du mot (Voir Wikipedia pour plus d'informations).

Examinons maintenant le placement flottant.

✎ À faire

- Rajouter la propriété de flottant à droite à la balise aside.
- Constater que l'image a changé de place, et faire varier la taille de la fenêtre pour visualiser le placement du texte autour de l'image.
- Constater également que l'image s'est éloignée du haut de la page. L'explication est que la boîte flottante est "bloquée" par les marges des boîtes dans laquelle elle flotte.

6. Portfolio (à rendre)

Dans la suite, nous nous intéressons à appliquer les positionnements Flex, relatif, et absolu pour la création d'une page *projets* dans le portfolio.

✎ À faire

- Reprendre le portfolio et ajouter une page *projets* dans laquelle des miniatures des projets réalisés sont placées avec un positionnement Flex. Se baser sur les projets réalisés dans le cadre des SAÉ ;
- Pour chaque miniature, appliquer le positionnement combiné relatif / absolu pour mettre ne valeur le projet en question. Par exemple :
  - Lors du survol par la souris, appliquer un filtre `filter: brightess(50%);`
  - Toujours lors du survol, afficher un lien "en savoir plus" qui permet d'aller sur une nouvelle page décrivant plus en avant le projet ;
  - Pour un effet visuel intéressant, il est possible de jouer avec les transitions CSS. Par exemple :

```
.miniature {
  transition: all 1s;
}
```

- Déposer le portfolio sur Chamilo