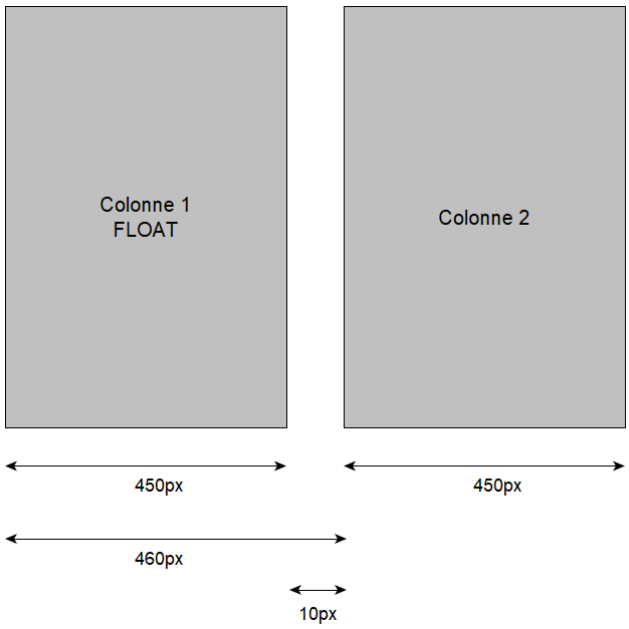
EP2IC3IV - ECUE Introduction au web

Création de colonnes en CSS

Les colonnes permettent de structurer le contenu du corps d'une page web. Créer des colonnes consiste à placer côte à côte des éléments HTML.

La **première étape** consiste à décider le nombre et la taille de ces colonnes. La **seconde étape** consiste à placer les éléments les uns à côté des autres horizontalement en utilisant notamment la propriété float.

Prenons l'**exemple** suivant : on souhaite créer une page web à 2 colonnes. On décide que chaque colonne prend 450 pixels, ce qui est raisonnable si on suppose une largeur d'écran de 1024 pixels. La première colonne aura la propriété "float : left", mais pas la 2ème colonne. En revanche sur cette dernière, on devra donner la propriété "margin-left : 460px;" ce qui donnera un espace de 10 px entre les 2 colonnes si on suppose une bordure de 0 px. Si la bordure est de 2 px pour la colonne 1 et qu'on veut tout de même 10px entre les 2 colonnes on utilisera la règle "margin-left : 464 px;" pour positionner la colonne 2.



```css
.col1 {
    border: 2px solid blue;
    float: left; width: 450px;
    margin-left: 0px;
    margin-right: 0px;
}

.col2 {
    border: 2px solid black;
    margin-left: 460px;
    width: 450px;
}
```

Ci-dessus nous avons abordé la notion de float. Un élément flottant est retiré du flux et placé soit à l'extrême gauche (float : left;) ou droite (float : right;) de son conteneur. Tout élement du flux qui suit le ou les éléments flottants l'enveloppe (on parle d'habillage). Si deux éléments flottent dans la même direction alors ils se rangeront côte à côte. Ainsi en reprenant l'exemple ci-dessus, la colonne 1 va se placer au niveau du bord gauche de la fenêtre car elle a suffisamment de place pour s'y insérer. La colonne 2 va habiller par la suite la colonne 1 en suivant le flot normal (haut bas, gauche puis droite). Notons que la colonne 2 pourrait se retrouver en dessous de la colonne 1 si par hasard la largeur de la fenêtre était trop étroite.

La **troisième étape** consiste à utiliser la propriété "clear", qui permet à un élément de ne plus subir le comportement d'habillage dicté par un objet flottant qui le précède directement et, par conséquent, de se caler en-dessous de ce dernier. On peut déclarer un nettoyage des flottants à gauche (clear : left;) ou à droite (clear : right;) ou les deux (clear : both;). Pour plus d'informations : **https://www.alsacreations.com/tuto/lire/608-initiation-positionnement-css.html#flottants**.

Quelques remarques : Dans cet exemple les largeurs de colonnes sont toutes fixes. Pour éviter d'entasser toutes les colonnes à gauche si la fenêtre du navigateur a une taille très supérieure à la largeur totale des colonnes, il faut centrer l'ensemble des colonnes.

Création de menus

Pour créer un menu de navigation horizontal ou vertical, la **première étape** consiste à définir une liste non ordonnée avec la balise *<ul>*. Chaque item de la liste correspond à une entrée du menu et est définie par un hyperlien (*e.g.* vers une autre page web ou section dans la page). Une entrée du menu peut elle même basculer vers un sous-menu. Si vous souhaitez utiliser des balises sémantiques, la liste devra se situer à l'intérieur d'une balise *<nav>* (sinon la balise neutre * *) ce qui permet d'isoler le menu du reste du contenu de la page web.

```html
<nav class="mmenu">
    <ul>
        <li> <a href="page1.html">Entrée 1 du menu</a> </li>
        <li> <a href="page2.html">Entrée 2 du menu</a> </li>
        <li> <a href="page3.html">Entrée 3 du menu</a> </li>
    </ul>
</nav>
```

Menu horizontal : approche par l'exemple

On commence par déclarer la taille du menu. Si chaque entrée du menu a une largeur de 150px alors pour 3 entrées on doit prévoir une largeur de 450px. Si vous souhaitez ajouter des bordures d'une largeur de 1px alors 6px doivent être ajoutés à la largeur prévue. Le menu aura donc une largeur de **456px**.

```css
.mmenu > ul {
    width: 456px;
}
```

Notez la présence du combinateur **>** qui facilitera la création d'un sous-menu qui se déroule vers le bas. Il permet en effet de sélectionner les éléments *<ul>* imbriqués dans l'élément de la classe *.mmenu* sans risque de confusion avec d'autres éléments <ul> qui pourraient être déclarés à l'intérieur des items de cet élément <ul>.

Le menu correspondant est le suivant :

- Entrée 1 du menu
- Entrée 2 du menu
- Entrée 3 du menu

Le mode *block* par défaut des *<ul>* impose un affichage vertical des éléments de la liste. Utilisons la technique des flottants pour aligner horizontalement toutes les entrées du menu comme proposé dans la section précédente.

```
.mmenu > ul > li {
    float: left;
    border:1px solid #8AC007;
}
```

Le menu résultant :

- Entrée 1 du menu Entrée 2 du menu Entrée 3 du menu

Pour enlever les "*bullet*" des listes il suffit d'ajouter : *list-style-type: none;* aux déclarations CSS de la liste concernée. Afin de décorer chaque entrée, une couleur d'arrière plan est ajoutée. On souhaite que l'arrière plan de l'élément *< a>* s'étende sur la totalité du bouton d'entrée du menu, indépendamment de la taille du texte du lien. Pour cela la propriété *width* est appropriée mais ne s'applique qu'aux éléments *block* et *<a>* est un élément " *inline*". Nous transformons donc *<a>* en élément *block* et fixons la taille bouton à 150px comme spécifié au début du paragraphe.

```
.mmenu > ul > li > a {
color: white;
background-color: red;
display: block;
width: 150px;
}
```

Enfin visuellement afin d'aider le lecteur de la page web à savoir sur quel bouton il a placé (sans cliquer) la souris, nous utilisons la pseudo-classe *:hover*.

```
.mmenu > ul > li a:hover {
background-color: blue;
}
```

Le menu horizontal résultant est le suivant :

Entrée 1 du menu Entrée 2 du menu Entrée 3 du menu

Nous avons conçu un menu fonctionnel avec les effets visuels de base. Il pourrait être encore amélioré en ajoutant des effets de bordures qui simulent l'effet d'un bouton "cliqué" typiquement. Jusqu'ici nous avons travaillé avec les boîtes " *<li>*" en mode flottant. Nous devrons donc appliquer la propriété "*clear : both*" sur la prochaine boîte afin de rendre à notre page Web son flux normal.

Sous-menus : approche par l'exemple

Nous souhaitons désormais ajouter des sous entrées à chaque entrée du menu. Nous allons créer un sous-menu, visible uniquement lorsque la souris se trouve sur une entrée donnée (pour l'exemple nous choisissons l'entrée 2 du menu).

Commençons par déclarer le contenu de ce sous-menu en supposant que la souris pointe l'entrée : "*Entrée 2 du menu*" (en fait, dans l'html ci dessous, c'est dans Entrée 1 et non 2 que se trouve le sous menu, mais vous comprendrez quand même les explications suivantes.)

```
<nav class="mmenu">
    <ul>
        <li><a href="page1.html">Entrée 1 du menu</a>
            <ul class="smenu">
                <li>entrée 1 sous-menu</li>
                <li>entrée 2 sous-menu</li>
            </ul>
        </li>

        <li><a href="page2.html">Entrée 2 du menu</a></li>
        <li><a href="page3.html">Entrée 3 du menu</a></li>
    </ul>
</nav>
```

Pour pouvoir changer les propriétés de l'élément *<ul>* contenant le sous-menu, nous devons déclarer la classe nommée pour l'occasion "*smenu*". Ensuite nous déclarons comme invisible le sous-menu avec "*display : none ;*". On

constate que même si un élément HTML est présent, la prise en compte de sa boîte CSS dans l'agencement des boîtes dans la fenêtre de navigation peut être rendue optionnelle.

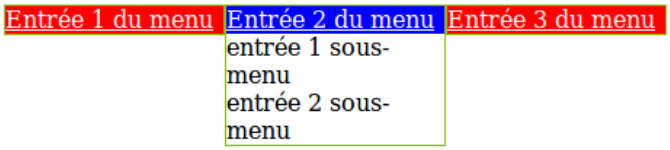
```
ul.smenu {
width: 150px;
padding-left: 0px;
display: none;
}
```

Remarque : La propriété *padding-left*: 0px; permet d'aligner chaque entrée du sous-menu avec l'entrée du menu principal.

La dernière étape consiste à rendre visible le sous-menu lorsque la souris survole l'entrée 2 de notre menu principal. (en fait, et plus généralement, lorsque survole toute entrée du menu qui contient un sous-menu)

```
.mmenu ul li:hover > ul {
display: block;
list-style-type: none;
}
```

La page web résultant est la suivante en supposant que la souris pointe l'entrée : "*Entrée 2 du menu*".



Notez l'utilisation du combinateur ">" après "*li :hover*" qui permet de s'assurer que l'élément <*ul*> qui s'affichera sera bien celui inclus dans l'élément <li>, et pas un éventuel <ul> qui se trouverait dans un sous-sous-sous...-menu. Pour afficher l'élément concerné on déclare la propriété "*display : block ;*" . Notez également que la pseudo classe *:hover* peut s'appliquer à autre chose qu'un hyperlien.

Menu vertical : approche par l'exemple

Un menu vertical suit le même principe que le menu horizontal à la différence que les éléments <ul> sont par défaut en mode *block* et n'a pas besoin d'être modifié par la propriété *display*. Savoir fabriquer un menu vertical est bien utile si l'on veut par exemple positionner le menu sur un des côtés de la page dans une colonne et laisser le reste de la page occuper l'espace libre.

La **première étape** consiste à découper en 2 colonnes la page web :
une colonne contenant le menu,
l'autre colonne contenant le reste de la page.
La deuxième étape consiste à fixer les marges de ces zones de manière appropriée. Ceci est d'autant plus intéressant si on a besoin de se déplacer verticalement dans la zone de contenu tout en gardant à tout moment le menu visible.

Pour obtenir le résultat désiré, donc pour continuer à voir le menu à gauche une fois descendu en bas dans la page, il faut fixer statiquement sa position par rapport à la fenêtre du navigateur. Il n'est donc pas nécessaire d'utiliser un "*float :left;*". Cependant il faut faire attention à ce que les éléments de contenu soient suffisamment décalés sur la droite grâce à un *margin-left* approprié. Le plus simple est de fixer cette marge pour le body en entier ce qui **par héritage** sera appliqué à tous ses éléments.

```
body {
margin-left: 180px;
}

.mmenu {
left: 10px;
width: 150px;
}
```

Exercice

Toutes les ressources dont vous aurez besoin pour les exercices suivants sont disponibles en fin de page

Exercice 1 : Calcul de distance pour 3 colonnes

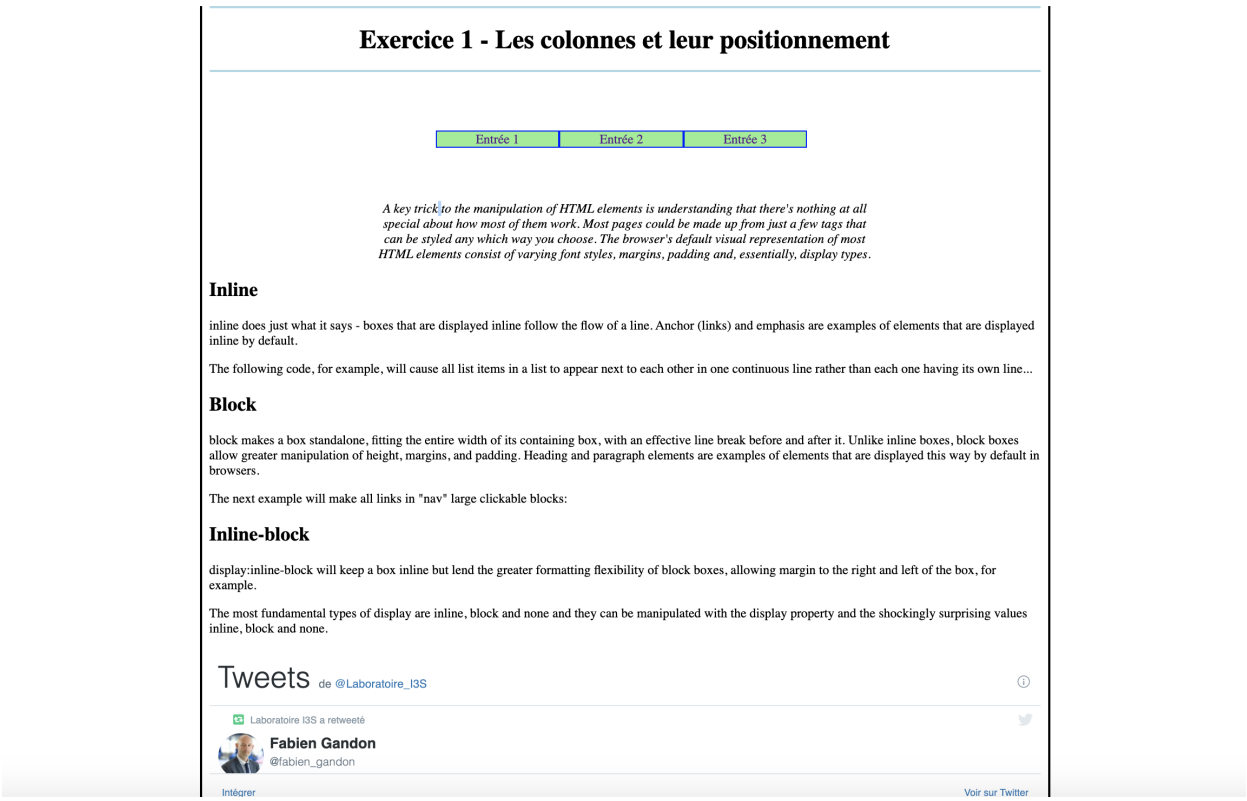
Observez le code CSS ci-dessous :

```css
.col1 {
    border: 2px solid blue;
    float: left;
    width: 200px;
}


.col2 {
    border: 2px solid black;
    float: left;
    margin-left: 10px;
    width: 200px;
}


.col3 {
    border: 2px solid black;
    margin-left: 428px;
}
```

Pourquoi un *margin-left* de 428px donne une distance de 10px entre la colonne 2 et 3?

Exercice 2 : Création d'un menu horizontal

Sans modifier le code HTML du fichier `2_exo.html` et en ajoutant des règles CSS à la feuille de style que vous avez écrite pour l'exercice 5 du TP précédent, créez un menu horizontal comme décrit dans la figure suivante :



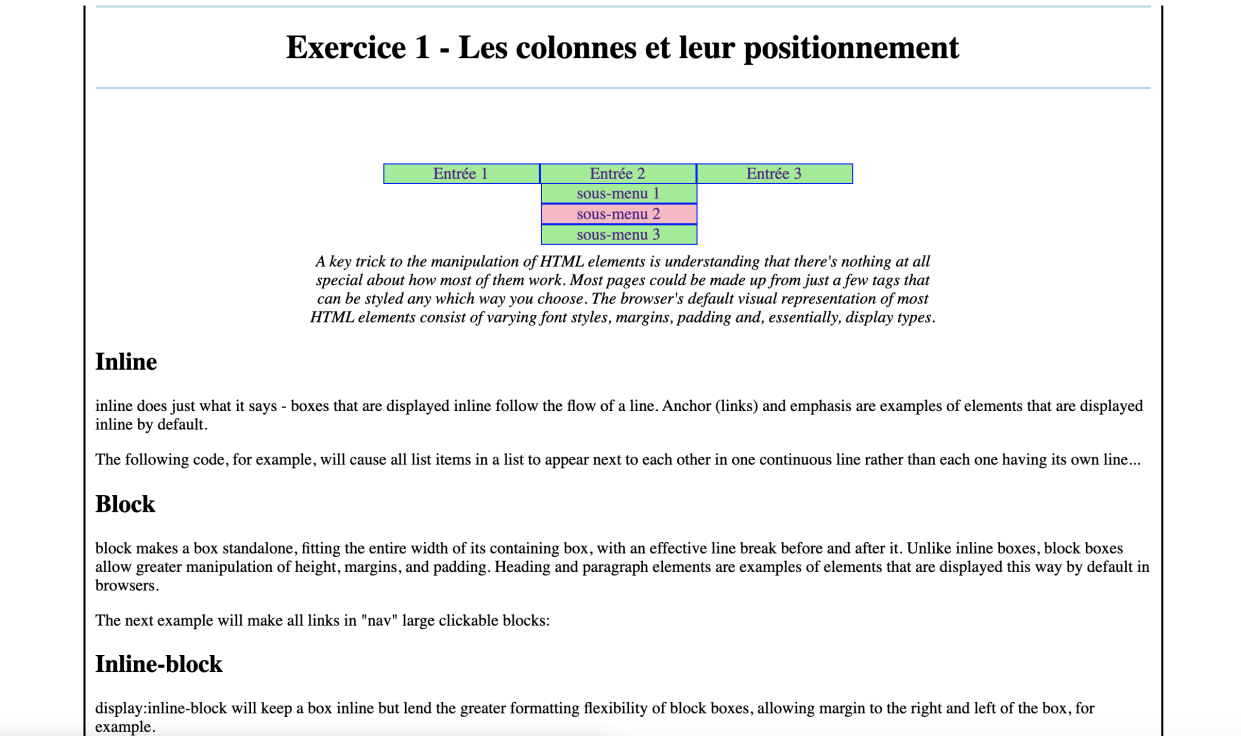
Précisions :

Commencez par ajouter un commentaire CSS (e.g. "/* ======= règles exercice 2 ======= */"), qui marquera le début de vos réponses pour cet exercice.
le bord des entrées du menu est de couleur bleu et 2 pixels d'épaisseur
il y a 50 pixels d'espace entre la balise "ul" du menu et l'élément précédent
créez un menu horizontal en vous inspirant de la section "Création de menus -> Menu horizontal"

chaque entrée du menu (chaque « bouton ») a une largeur de 150 pixels, fond "lightgreen" et le fond devient "lightpink" lorsque la souris le survole.
la div avec classe "separator" introduit un espace de 50 pixels avec l'élément suivant afin de bien délimiter l'espace entre le menu et le contenu principal de la page
la balise "ul" faisant office de sous-menu est pour l'instant cachée (il n'y a pas besoin de gérer l'affichage dynamique pour cet exercice)
le menu horizontal est centré sur la page (i.e., entre les bords droit et gauche)
pour supprimer le soulignement d'un lien hypertexte, utilisez la propriété "text-decoration: none;".

Exercice 3 : Création d'un sous-menu vertical dynamique

Sans modifier le code HTML de 2_exo.html, ajoutez à votre feuille de style de l'exercice 2, les règles nécessaires pour créer un sous-menu vertical dynamique. Séparez les règles de l'exercice précédent de l'exercice en cours avec un commentaire CSS (e.g. "/* ======= règles exercice 3 ======= */").



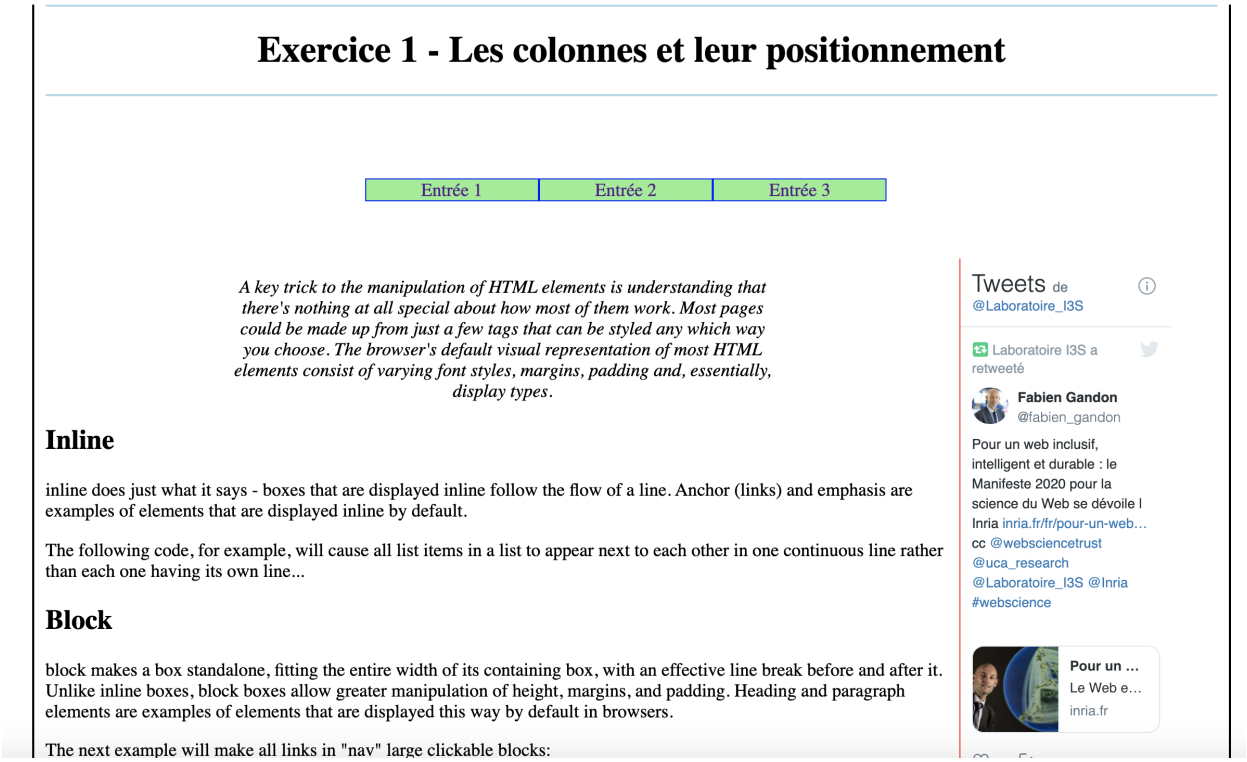
Précisions :

le sous-menu devient visible lorsque la souris survole "Entrée 2"
l'entrée "sous-menu 2" possède un fond "lightpink" car la souris la survole (souris non visible dans le screenshot)
le sous-menu reste invisible si la souris quitte le sous-menu ou "Entrée 2" du menu principal
vous pouvez affecter la propriété "position:absolute;" sur le sous-menu afin d'éviter que celui-ci ne déplace vers le bas le contenu principal de la page
une marge négative de -2px vous aide à mieux centrer le sous-menu sur l'entrée du menu principal. Pourquoi -2px d'ailleurs ?

Exercice 4 : Colonnes I

Sans modifier le code HTML de 2_exo.html, ajoutez à votre feuille de style de l'exercice précédent les règles nécessaires pour créer une page avec une colonne de contenu principal et une colonne possédant un flux twitter. Séparez les règles de l'exercice précédent de l'exercice en cours avec un commentaire CSS (e.g. "/* ======= règles exercice 4 ======= */")

La mise en page attendue est la suivante :



Précision :

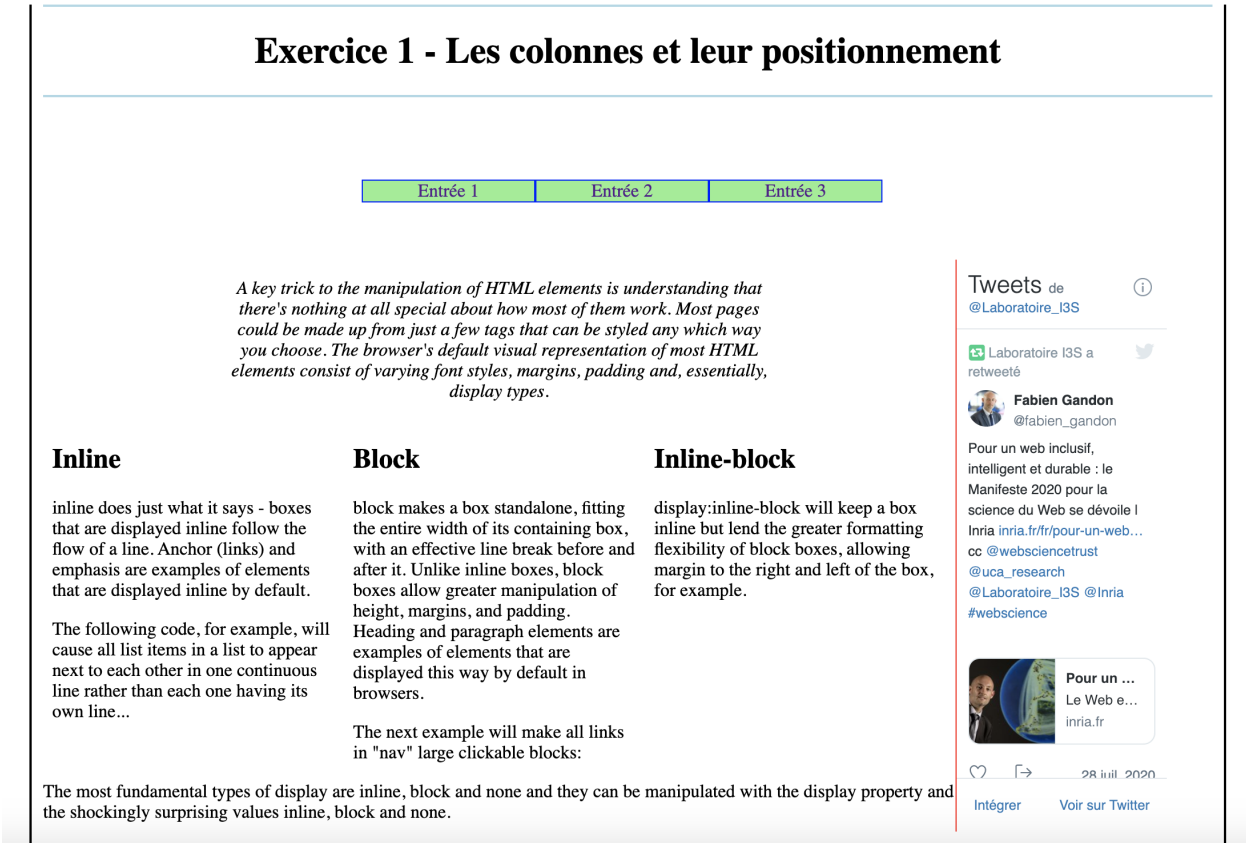
La colonne avec le contenu principal possède une largeur de 75% de la page.
La colonne avec le flux twitter possède une hauteur de 500px.
Dans cet exercice, les 2 colonnes ont été créées en faisant flotter uniquement l'une d'entre elles.
Pour mieux séparer visuellement le flux twitter du contenu principal, un bord rouge d'1px d'épaisseur est dessiné sur la div contenant le flux twitter

Exercice 5 : Colonnes II

Sans modifier le code HTML de `2_exo.html`, ajoutez à votre feuille de style de l'exercice précédent les règles nécessaires pour créer une page dont le contenu principal possède un texte d'introduction (paragraphe en italique) sur une seule colonne, mais est suivi de trois colonnes. Après les 3 colonnes, il y a un paragraphe final placé sur une seule colonne.

Le flux twitter (complètement indépendant du contenu principal) reste inchangé. Séparez les règles de l'exercice précédent de l'exercice en cours avec un commentaire CSS (e.g. "/* ======= règles exercice 5 ======= */").

La mise en page attendue est la suivante :



Précisions :

Cette fois-ci, les trois colonnes sont des éléments flottants.
la largeur de chaque colonne du contenu principal est de 31%, avec des marges à droite et à gauche d' 1%

Votre progression

Ressources TD2

Rendus TD2

◄Concepts et notions de CSS

Responsive Web Design►

Tableau de bord
Cours
Tous les cours
Recherche de cours
Accompagnement
Pédagothèque
Documentation
Français (fr)
English (en)
Français (fr)

Résumé de conservation de données