

Balises HTML

On crée une page pour la liste de courses. Le titre doit être centré, les éléments importants écrits en *gras*. Une partie de la liste se trouve en ligne sur votre compte *marmiton.org*.

```
1 <style>
2     ... {
3         text-align: center;
4     }
5     ... {
6         list-style-type: square;
7     }
8     .
9         {
10            text-decoration: bold;
11        }
12 </style>
13 <body>
14     <h1>Liste de courses</h1>
15     <img ... = "caddie.png">
16     <ul>
17         <li>Café</li>
18         <li>Thé</li>
19         <li class="plusimportant">Chocolat</li>
20     </ul>
21     <p>Le reste de la liste: <a ... = "marmiton.org/client=20290&
    liste=234">Clic</a></p>
22 </body>
```

1. Compléter le script CSS avec les selecteurs appropriés.
2. Compléter les attributs des balises <a> et

Positionnement

2.1 display

L'image suivante illustre des éléments dont la propriété display a été modifiée

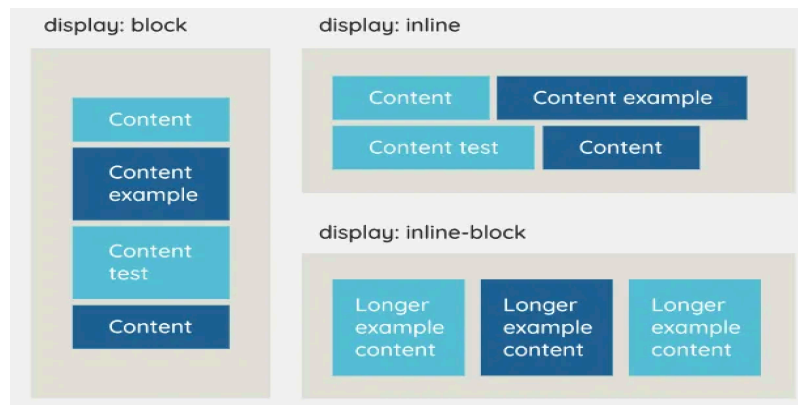


FIGURE 1 – élément p

1. Comment ceux-ci se disposent-ils les uns par rapports aux autres, selon leur display?
2. Citer 2 éléments html dont le display est block
3. Citer 2 éléments html dont le display est inline
4. Pour chaque affirmation, préciser q'il s'agit du display inline ou block :
 - l'élément occupe toute la largeur de son parent :
 - l'élément se met au dessous du précédent :
 - l'élément se met à côté du précédent :
 - la largeur ainsi que sa hauteur sont modifiables avec les paramètres css width et hight :
 - la largeur s'adapte au contenu :
 - on peut centrer les éléments contenus :

2.2 Reproduire une mise en forme

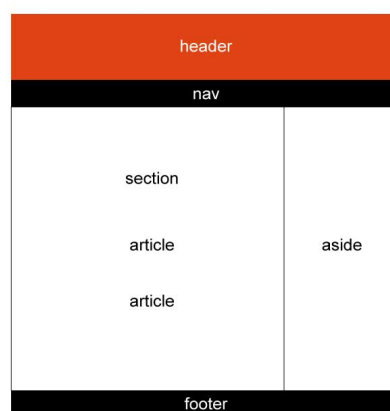


FIGURE 2 – page avec sections

Les éléments *containers* de type paragraphes, sections, header, nav, article, ... ont pour display la valeur block par défaut. Ils vont se placer naturellement l'un sous l'autre.

Par contre, s'il faut disposer côte à côte 2 éléments containers, pour faire 2 colonnes, il faudra les disposer à l'intérieur d'un autre élément parent. Puis modifier leur disposition naturelle (paramètre `display: flex;`)

1. Ecrire le script HTML de ces balises pour obtenir la mise en forme de l'image précédente. (sans le contenu)
2. Dessiner l'arbre du DOM

Exercice 3

HTML et CSS

3.1 Paragraphe et lien

Ecrire le script HTML qui permet d'afficher le paragraphe suivant :


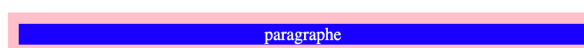
Prendre rendez vous:  01 79 48 23 60 ou bien sur le site [doctolib.fr](https://www.doctolib.fr)

FIGURE 3 – l'image phone.png se trouve dans le même dossier que la fichier html

3.2 règles CSS pour le texte et le container

Le premier paragraphe montre un texte centré, coloré en blanc, avec un fond coloré en bleu et une bordure rose de largeur 10px.

paragraphe avec les regles css n°1



paragraphe avec les regles css n°2



FIGURE 4 – Deux paragraphes aux propriétés personnalisées

1. Ecrire les scripts HTML et la déclaration CSS relative à ce premier paragraphe
2. Le deuxième paragraphe diffère du premier par sa bordure intérieure. Ecrire la règle CSS qui a été ajoutée.

Exercice 4

Propriétés CSS

Parmi les paramètres suivants, lesquels peuvent être employés pour des règles de style relatives à un élément textuel ?

```
1 color
2 background-color
3 font-family
4 font
5 text-align
6 font-weight
```

Parmi les paramètres suivants, lesquels peuvent être appliqués à un élément de type bloc ?

```
1 border
2 margin
3 padding
4 vertical-align
5 width
6 height
```

Parmi les paramètres suivants, lesquels peuvent être appliqués à un élément de type inline ?

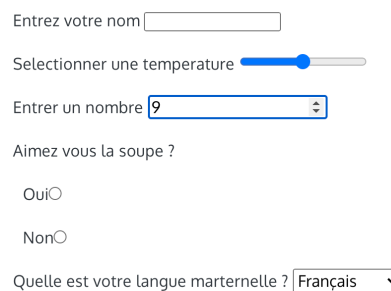
```
1 border
2 margin
3 padding
4 vertical-align
5 width
6 height
```

Exercice 5

Formulaires

Les formulaires permettent à l'utilisateur de saisir un **ensemble de données** et d'envoyer une **requête** à un serveur en cliquant sur un bouton. Le formulaire est délimité par l'élément HTML `<form>` qui précise :

- L'URL (attribut `action`)
- La méthode (attribut `method`)



Entrez votre nom

Sélectionner une température

Entrer un nombre

Aimez vous la soupe ?

Oui ☐

Non ☐

Quelle est votre langue maternelle ?

FIGURE 5 – Exemple de formulaire

Il contient un ensemble d'éléments interactifs permettant la saisie de données, spécifiés par l'élément `<input>`. Son attribut `type` permet de préciser le type de saisie :

- Text
- Number
- Range (valeur numerique)
- Checkbox
- Radio
- Button
- ...

Les type *reset* et *submit* sont des boutons speciaux pour reinitialiser ou envoyer la requete vers le serveur. L'attribut `value` spécifie la valeur initiale du champ de saisie.

Voici un exemple de script HTML pour un formulaire.

1. Quels sont les attributs obligatoires pour les balises des éléments *form*, *label* et *input* ? Quel est leur rôle ?
2. Représenter l'élément *input* de type *text* tel qu'il sera affiché dans la page.

```
1 <form method="post" action="action.php">
2   <p>
3     <label for="prenom">Entrez votre prénom :</label>
4     <input type="text" name="prenom" id="prenom" placeholder=
      "Ex: Jojo" size="30" maxlength="10" />
5   </p>
6   <p>
7     <input type="submit" value="Envoyer" />
8   </p>
9 </form>
```

Exercice 6

Position

6.1 Positionner des carrés l'un par rapport à l'autre

Deux carrés sont positionnés de la manière suivante



FIGURE 6 – position relative de 2 carrés

Le script correspondant est le suivant :

```
1 <div class="parent-div">
2   <div class="child-div"></div>
3 </div>
4
5 <style>
6 .parent-div{
7   position: relative;
8   background-color: #CCC;
9   width: 200px;
10  height: 200px;
11 }
12
13 .child-div{
14   display: inline-block;
15   width: 50px;
16   height: 50px;
17   position: absolute;
18   right: 0px;
19   bottom: 0px;
20   background-color: blue;
21 }
22 </style>
```

Pour positionner un élément fils par rapport à son parent, les 2 éléments doivent avoir la propriété CSS suivante :

- pour le parent : `position: relative`
- pour l'enfant : `position: absolute`

Il y a alors 2 options pour le positionnement :

- Option 1 : On définit alors la position de l'enfant par rapport aux rebords du parent, avec les paramètres **right** (bord droit), **left**, **top** (bord supérieur), **bottom**.
- Option 2 : utiliser des marges pour les éléments : **margin-left**, **margin-right**, ...

1. Quel est la classe de l'élément parent ?
2. Quel est la classe de l'élément fils ?
3. Peut-on interpréter cette relation parent-fils à partir du script HTML ?
4. Quelles sont les règles CSS qui permettent de positionner l'élément fils ?
5. Comment faut-il modifier ces règles pour centrer le petit carré dans le grand ?

6.2 Design d'une application de SMS



FIGURE 7 – blague

En vous inspirant de l'exercice précédent, créer un mini dialogue entre 2 personnes (une question, une réponse, une autre question). Ajouter les règles de style pour que la présentation de la conversation ressemble à celle d'une application de messagerie instantanée.

Exercice 7

Règles CSS orientée objet

Souvent, les objets semblables partagent les mêmes règles, avec certaines différences. Comme par exemple, ici, pour 2 boutons rouge et vert :

```

1  .boutonRouge {
2      display: block;
3      font-size: 12px;
4      color: red;
5      border: 1px solid red;
6  }
7
8  .boutonVert {
9      display: block;
10     font-size: 12px;
11     color: red;
12     border: 1px solid red;

```

```
13 }
```

On souhaite rassembler dans une même classe les règles communes à tous les boutons. Et créer d'autres classes selon les différences entre ces boutons.

On utilisera alors la balise HTML suivante pour appeler le bouton rouge ; l'élément div a alors 2 classes :

```
1 <div class="bouton bouton-rouge">Bouton rouge</div>
```

Modifier les règles CSS en partageant celles-ci entre les classes :

- bouton
- bouton-rouge
- bouton-vert

Exercice 8

Correction

8.1 Exercice positionnement des carrés

```
1 <div class="parent-div">
2   <div class="child-div"></div>
3 </div>
```

```
1 .parent-div{
2   position: relative;
3   background-color: #CCC;
4   height: 200px;
5 }
6
7 .child-div{
8   display: inline-block;
9   width: 50px;
10  height: 50px;
11  position: absolute;
12  right: 0px;
13  bottom: 0px;
14  background-color: blue;
15 }
```

8.2 Exercice SMS

```
1 <div class="container">
2   <div class = "gauche">
```

```
3     Hello
4     </div>
5 </div>
6
7 <div class="container">
8     <div class = "droite">
9         Hey! ça va?
10    </div>
11 </div>
12
13 <div class="container">
14     <div class = "gauche message2">
15         Yep!
16     </div>
17 </div>
```

```
1 .container {
2     width: 200px;
3     position: relative;
4     background-color: red;
5 }
6
7 .gauche, .droite{
8     position: absolute;
9     height: 50px;
10    display: inline-block;
11    width: 100px;
12    border-radius: 10px;
13 }
14
15 .milieu {
16     height: 10px;
17     background-color: red;
18 }
19
20 .gauche {
21     background-color: yellow;
22 }
23
24
25 .droite {
26     top:60px;
27     //right:0;
28     margin-left:50px;
29     background-color: blue;
30     text-align: right;
31     color: white;
32 }
33
34 .message2 {
```

```
35     top: 120px;  
36 }
```