

Nous allons dans cet atelier découvrir, par la pratique, les bases des feuilles de styles en cascade CSS (Cascading Style Sheets).

Le langage HTML est assez simple et son implémentation comporte peu d'éléments.

Il n'en est pas de même du langage de présentation CSS, aujourd'hui dans sa version 3.

Le travail sur le rendu visuel d'une page peut nécessiter des phases de mise au point longues et parfois fastidieuses.

Nous devons donc être rigoureux et attentifs à notre manière de coder.

Pour chaque bloc d'exercices, vous trouverez une page HTML de base que vous devrez mettre en forme en respectant les consignes communiquées.

Dans un objectif de simplification, la feuille de style sera incorporée à la page.

Ce n'est pas l'option à privilégier dans la "vie réelle". Il faut mieux externaliser les définitions de styles dans un document externe lié à nos pages HTML.

En effet, nombre de styles sont mis en facteur et partagés par plusieurs pages afin d'éviter, d'une part, la redondance et donc faciliter la maintenance et, d'autre part, assurer un rendu visuel cohérent sur l'ensemble des pages du site.

Nous devons observer le principe DRY bien connu en programmation, Don't Repeat Yourself.

Les sélecteurs et les règles Css

Regardons la syntaxe à mettre en œuvre pour définir nos styles.



Les règles de styles définies s'appliquent aux éléments paragraphe P

Les déclarations de style sont toujours définies par des couples

attribut : valeur séparés par un point virgule et encadrés par des accolades.

On parle de bloc de styles.

L'expression des sélecteurs cibles peut prendre des formes diverses.

Quelques exemples

- **p** : tous les éléments **p**
- **div p** tous les éléments **p descendants** d'une **div**
- **.menu > ul** tous les éléments **ul enfants directs** d'un élément dont l'attribut **class** est **menu**
- **p + span** un élément **span adjacent** à un **paragraphe**
- **.article** élément dont l'attribut **class** est **article**
- **#logo** élément dont l'attribut **id** est **logo**
- **input[type=text]** tous les éléments de formulaire **input** dont l'attribut **type** est **text**
- **h1,h2,h3** définition **commune** pour les éléments **h1,h2 et h3**

Retrouvez la documentation sur les sélecteurs Css de MDN

[Sélecteurs Css](#)

Téléchargez le document HTML à partir duquel réaliser la première série d'exercices.

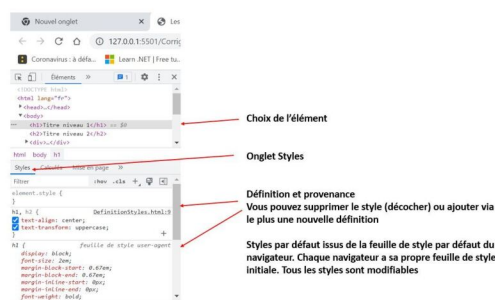
[Archive avec code initial \(cf. *Selecteurs.zip*\)](#)

! Conseil

Quelques conseils pour faciliter la mise au point des css et gagner en efficacité :

1. Testez individuellement chacune de vos modifications pour en apprécier l'impact.
2. Utilisez le mode développeur F12 (devtools) pour mettre au point plus facilement vos feuilles de styles.

Consultez l'illustration ci-dessous pour comprendre comment utiliser le mode développeur pour la mise au point des styles.



Vous trouverez de nombreux supports relatifs aux outils du développeur. Il s'agit d'un ensemble de fonctionnalités incontournables.

[Lien vers vidéo présentant rapidement les fonctionnalités de devtools.](#)

Certaines explications données resteront abscondes pour le moment. Pas d'inquiétudes !

Concentrez vous sur la fonction inspecteur et les éléments HTML/CSS du DOM (Document Object Model).

Question 1

Définir les règles de style suivantes :

1. Le texte des titres de niveau 1 et 2 doit être en majuscules et centré
2. Le texte des titres de niveau 2 doit être en italique

Redéfinir les styles des liens adjacent à un paragraphe :

1. Suppression de la décoration du lien par défaut
 - Couleur de fond rouge
 - Couleur de texte jaune
2. Sur survol du lien
 - Couleur de fond jaune
 - Couleur de texte rouge

Redéfinir la présentation de la liste non ordonnée enfant de l'élément **nav** et ses éléments


1. Supprimer les symboles de la liste

2. Afficher les items de la liste en ligne
3. Affecter un fond vert à l'élément nav
4. Affecter la couleur blanche au texte
5. Modifier les liens : supprimer les styles par défaut et affecter la couleur blanche au texte des liens.

Indice

Je vous conseille d'introduire dans vos recherches quelques mots clés en préfixe pour obtenir rapidement des infos pertinentes.

Exemple : préfixe : mdn css + mots clés cibles (en français ou en anglais) : vous obtenez la documentation Mozilla sur cet effet et depuis ce point étendre votre lecture à Mettre en forme du texte

 [mdn css anchor hover](#)

Question 2

Pour bien comprendre l'héritage et l'application des styles en cascade, il convient de s'intéresser aux poids des différents sélecteurs.

Définissez les styles affectant les divisions d'espace et les paragraphes de la manière suivante :

1. La couleur de fond de l'élément div est vert
2. Le style de la classe Option1
 - La couleur de fond est en hexa #b10f2f
 - La couleur du texte est bleue
 - Le texte est en italique
3. Le style de la classe Option2
 - La couleur de fond est en RGB 183, 236, 25
 - La couleur du texte est bleu-violet
4. Le style associé à l'élément dont l'identifiant est parap
 - La couleur de texte est noire

Observez le résultat obtenu.

Vous assistez à des mécanismes de fusion et remplacement de styles qui obéissent à des règles définies en fonction du poids des sélecteurs Css

Vous trouverez des infos sur ce sujet dans le lien suivant :  [Cours css poids des sélecteurs](#)

Ainsi qu'un calculateur de poids des sélecteurs  [Specificity Calculator](#)

Le positionnement des éléments

Autre point très important à prendre en considération lors de la mise en forme de nos pages est les techniques de positionnement.

Il existe différents types de positionnement des éléments :

- Positionnement dans le flux (normal ou statique)
- Positionnement relatif
- Positionnement flottant
- Positionnement absolu
- Positionnement fixe
- Positionnement collant (sticky)

Ces différents positionnements permettent de gérer la répartition des éléments dans la page.

Je vous propose de découvrir quelques contextes de positionnement au travers des exercices suivants.

Vous trouverez des informations utiles sur les différents types de positionnement à partir des liens suivants :

[!\[\]\(ec9132f1d27c8919987d92907322654d_img.jpg\) mdn : Introduction au positionnement](#)

[!\[\]\(05be7c7a8995decd503647c99211f7c2_img.jpg\) mdn : positionnement flottant](#)

Nous verrons par la suite les boîtes flexibles (flexbox) qui feront l'objet d'ateliers spécifiques.

Question 3

Dans ce premier exercice, observons le positionnement par défaut dans le flux (static).

Télécharger l'exemple qui servira de source à cette partie de l'atelier. [!\[\]\(fe3aebe81acea8d45108cd2768939da7_img.jpg\) Télécharger \(cf. \[PositionStatic.zip\]\(#\)\)](#)

Il contient une page html, une feuille de style incorporée et des images.

Sauriez-vous expliquer pourquoi certains éléments occupent toute la place alors que d'autres sont collés les uns aux autres sur une ligne.

Vous pouvez aussi constater que des définitions de marges existent alors qu'aucune mise en forme les concernant n'a été définie.

Utilisez pour cela l'option inspecter de votre navigateur et passez en revue les définitions de style.

D'où proviennent ces définitions ?

Doit-on s'en préoccuper ?

Question 4

Continuons notre découverte du positionnement.

Reprenons la page exemple précédente, introduisons une liaison vers la feuille de style normalize.css téléchargée.

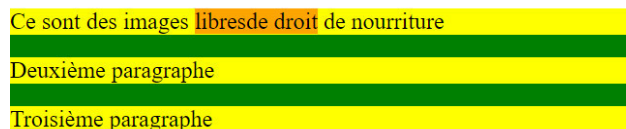
Définissons les marges de l'élément paragraphe, p.

Nous souhaitons que les paragraphes aient une marge extérieure verticale de 15 px en haut et en bas.

Que constatons-nous ?

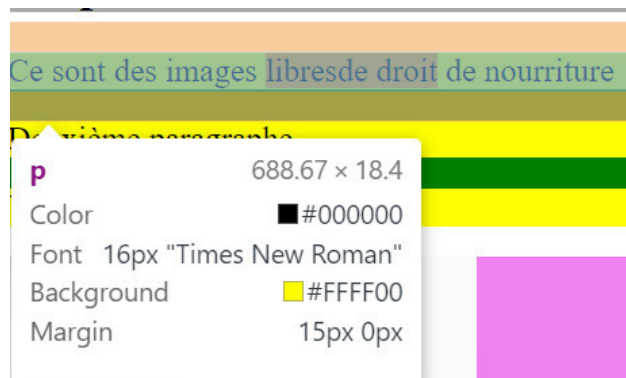
Les marges verticales fusionnent.

Ainsi, l'espace entre le 2ème et premier paragraphe est de 15 et non de 30 (top 15 + bottom 15)



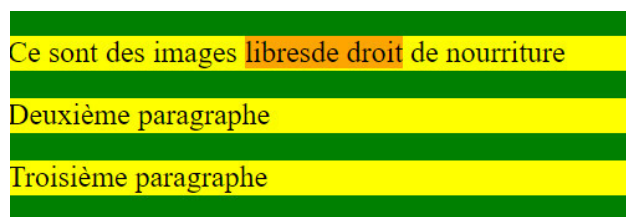
Les marges top et bottom ne semblent pas être appliquées au premier et dernier élément. En fait, elles se trouvent en dehors du contour de la boîte comme le montre l'illustration ci-contre.

Ce phénomène, assez contre-intuitif, peut être supprimé.



Vous pouvez :

- changer la propriété **overflow** du parent **div** de **visible**, valeur **par défaut**, à tout autre valeur valide. Je choisis ici **hidden**.
- Ou afficher une bordure autour du conteneur **div**.



Essayez les 2 solutions. Elles peuvent être aussi cumulées !

Question 5

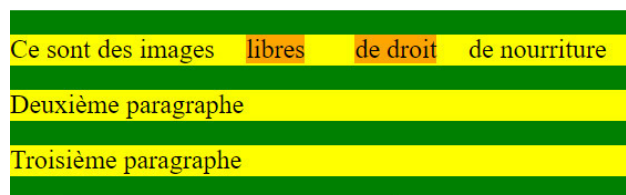
Poursuivons avec la définition des marges sur des éléments **inline**, comme l'élément **span**

Définissons des marges de 15 px sur les 4 directions et observons le résultat

Sélectionner les éléments **span** avec l'outil inspecteur.

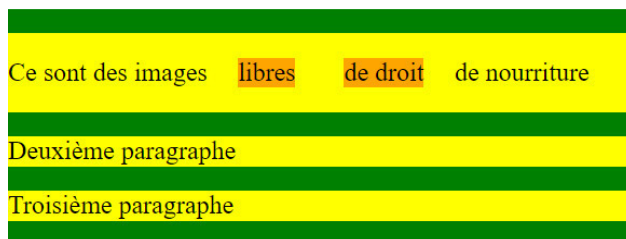
Vous constaterez que les marges horizontales ne fusionnent pas, contrairement aux marges verticales.

Il n'existe pas de marges verticales avec les éléments inline.



Pour obtenir une marge verticale, nous allons modifier le comportement des éléments span et leur disposition en définissant la propriété **display à inline-block**.

Nous obtenons alors les marges souhaitées.



Question 6

Regardons maintenant du côté des images.

L'élément img, défini de base comme inline, se comporte en fait comme un inline-block.

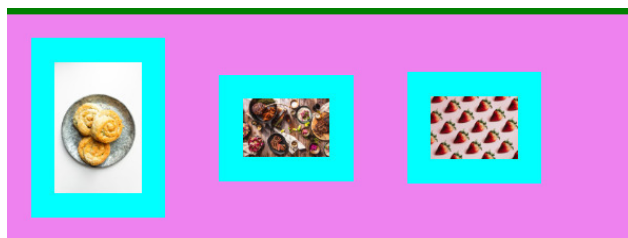
Nous pouvons le positionner verticalement et horizontalement, lui attribuer des marges verticales et réduire ou augmenter l'espace qu'il occupe.

Vérifions en lui appliquant la mise en forme suivante :

- Marge externe de 15 px
- Marge interne de 10 px (encadré)
- Positionnement vertical : milieu
- Largeur uniforme de 55 px (la hauteur sera calculée proportionnellement)

Le résultat est conforme à la définition.

Nos images sont centrées verticalement et des marges internes et externes présentes.



Les positionnements relatif, absolu, fixe et adhérent (sticky)

Ces styles de positionnement utilisent tous les propriétés de positionnement top, bottom, left, right pour placer les éléments.

Le positionnement relatif décale simplement l'élément du flux normal en tenant compte des positions définies.

Le positionnement absolu sort l'élément du flux normal et le positionne en fonction des valeurs définies à partir :

- de la position du premier ancêtre positionné (s'il existe....)
- à partir du conteneur global body sinon

Le positionnement fixe et sa variante, sticky, permet de figer la position d'un élément même en cas de défilement du contenu (scroll).

Vous pouvez consulter [cette vidéo](#) qui vous présente ces 4 types de positionnement

Téléchargez l'archive qui contient la source nécessaire aux exercices qui suivent. [Télécharger \(cf. PositionRelAbsFixSti.zip\)](#)

Question 7

Dans cet exercice, vous devez mettre en œuvre les positionnements relatifs et absolus pour obtenir un résultat analogue à celui présenté ci-dessous.

La boîte 3 de la première série est décalée et vous constatez que son emplacement dans le flux initial est préservé. Ainsi, la boîte suivante se trouve dans le flux à sa position normal (static).

L'un des usages du positionnement relatif est le décalage de portions de texte pour créer des effets.

Le positionnement absolu fonctionne différemment. La boîte 7 ne se trouve plus dans le flux mais en dehors, positionnée par rapport à son conteneur.

Le conteneur est positionné en absolu.

Complétez la définition des styles de la page PositionRelAbs.html.

Positionnement relatif

Positionnement absolu

Positionnements relatif et absolu

Relatif



La position relative peut être utilisée pour décaler du texte mis ainsi en exergue

Absolu



Vous pouvez jouer sur la propriété z-index pour mettre en arrière plan les éléments positionnés.

Question 8

Dans l'exercice suivant, mise en œuvre des positionnements fixed et sticky.

Ils ont un objectif similaire : conserver un élément en position fixe même en cas de débordement de contenu.

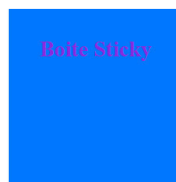
Toutefois, l'usage du positionnement sticky est dans la plupart des cas plus simple et préférable au positionnement fixed.

Usage très courant : positionnement des barres de navigation en haut ou en bas de la page.

Complétez la définition des styles des pages PositionFixed.html et PositionSticky.html afin d'obtenir le résultat tel qu'illustré ci-dessous.

Positionnement Fixed

Positionnement Sticky



Positionnement adhérent Sticky

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Quia sapiente voluptatem blanditiis praesentium magnam libero obcaecati dolores nemo? Voluptatem repudia cum eum ducimus harum rem fugiat possimus consequuntur sunt ea voluptatum, ad eligendi. Voluptatem reprehenderit, omnis, vitae provident consequatur ut quisquam

Vous noterez que le positionnement **sticky** conserve l'élément **dans le flux** alors que l'élément **fixed** est **en dehors du flux**. Pour que le titre soit correctement affiché et non masqué dans ce dernier cas, il faut jouer sur les marges du titre : cette option n'est pas judicieuse et peut rendre la mise au point du positionnement des autres éléments plus complexe.

Le positionnement flottant Float

Il s'agit assurément d'un des mécanismes de positionnement des plus courants pour faire flotter un élément au sein de contenus.

Ce mécanisme est bien connu en traitement de texte pour écrire du texte autour d'une image ou d'une lettrine.

Il peut être aussi utilisé pour réaliser des présentations multi-colonnes. Toutefois, il est aujourd'hui concurrencé, pour la présentation en colonnes, par les boîtes flexibles (FlexBox) et grilles (Grid) que vous verrez par la suite.

La position float retire une boîte du flux normal pour la placer le plus à droite (float :right) ou le plus gauche (float: left) possible dans son conteneur.

Le contenu suivant cette boîte flottante s'écoule le long de celle-ci, dans l'espace laissé libre.

Il faut prendre garde à la position du conteneur et à la position des éléments dans celui-ci.

Voir la documentation mdn [ici](#)

Télécharger le code source [ici](#) (cf. *Flottants.zip*)

Question 9

Le plus courant des usages : un texte entourant une image ou une lettrine.

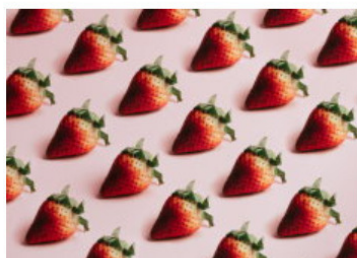
Coder la feuille de style incorporée de la page Flottant1.html afin d'obtenir le résultat présenté ci-dessous.

Positionnement Flottant



Lorem ipsum dolor, sit amet consectetur adipisicing elit. Vero quisquam eum dolorum. Nesciunt mollitia officia perferendis, fugit expedita nulla non architecto impedit, at aliquid excepturi rem placeat animi magni officiis assumenda nam! Cupiditate quae placeat nobis porro mollitia blanditiis, voluptates quisquam distinctio sunt, id odii itaque delectus odio minus dolores sit quidem modi officiis labore beatae voluptas, magni sequi? Impedit fuga quod tempore vero illum porro nam accusantium, voluptate Quo beatae enim, in commodi blanditiis eveniet nesciunt numquam, ipsum ex iure n minus? Vitae aperiam magni ipsa consequuntur, nihil at molestias quia nostrum maxime, quam optio distinctio ducimus qui! Odit possimus consectetur, recusandae,

distinctio. Consequatur pariatur at perferendis, molestiae sunt tenetur? Explicabo ducimus nemo reprehenderit vel ab rei laborum neque nesciunt itaque provident harum? Officiis fugiat natus impedit, iure eum veniam eaque provident. Veritatem voluptatum vitae exercitationem deserunt, dolores adipisci harum, eius fuga? Nam, mollitia voluptate! Illum, expedita ut quae mollitia quidem laborum iusto, possimus hic facilis blanditiis optio quis deserunt ab nobis esse? Dolorem harum officiis ipsam inventore minus. Corporis, necessitatibus? Eum asperiores voluptatibus ad at repellat perspiciatis laboriosam



Lorem ipsum dolor, sit amet consectetur adipisicing elit. Vero quisquam eum dolorum. Nesciunt mollitia officia perferendis, fugit expedita nulla non architecto impedit, aut

Question 10

Que se passe-t-il si nous inversons les groupes image/texte ?

Nous obtenons alors le résultat présenté ci-dessous qui ne correspond pas à notre objectif.

Ceci est dû au fait que le texte associé à l'image ne déborde pas du bloc image.

Modifiez la feuille Flottant2.html et sa feuille de style incorporée pour obtenir le résultat présenté ci-après.

Positionnement Flottant



Lorem ipsum dolor, sit amet consectetur adipisicing elit. Vero quisquam eum dolor Nesciunt mollitia officia perferendis, fugit expedita nulla non architecto impedit, aut



Lorem ipsum dolor, sit amet consectetur adi quia quos aliquid nisi a, nostrum saepe! N impedit, autem magnam optio, id quasi maic magni officiis assumenda nam! Cupiditate q Placeat nobis porro mollitia blanditiis, volup non! Obcaecati fugit temporibus itaque delect unde, eveniet tempora voluptatum expedita h porro nam accusantium, voluptatum rem dele in commodi blanditiis eveniet nesciunt numq

recusandae ratione minus? Vitae aperiam magni ipsa consequuntur, nihil at molestias quia nostrum aliquam sit soluta distinctio ducimus qui! Odit possimus consectetur, recusandae, aperiam nulla totam ullam debitis suscipit vero architecto c tenetur? Explicabo ducimus nemo reprehenderit vel ab reiciendis iure, distinctio doloremque aliquid, esse corrupti minin natus impedit, iure eum veniam eaque provident. Veritatis a ipsa provident eaque quis, accusantium suscipit ut possimus eius fuga? Nam, mollitia voluptate! Illum, expedita voluptatibus aut quisquam nam velit, obcaecati temporibus id perspi optio quis deserunt ab nobis esse? Dolorem harum omnis maiores sequi, error pariatur, tenetur autem porro molestiae, voluptatibus ad at repellat perspiciatis laboriosam non itaque.

Après correction, en appliquant les mécanismes de dégagement, nous obtenons le résultat souhaité.

Positionnement Flottant



Lorem ipsum dolor, sit amet consectetur adipisicing elit. Vero quisquam eu Nesciunt mollitia officia perferendis, fugit expedita nulla non architecto imp



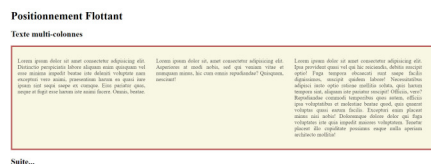
Lorem ipsum dolor, sit amet consectetur adipisicing elit. Vero quisquam eu Nesciunt mollitia officia perferendis, fugit expedita nulla non architecto in aliquid excepturi rem placeat animi magni officiis assumenda nam! Cupic Placeat nobis porro mollitia blanditiis, voluptates quisquam distinctio sunt, i delectus odio minus dolores sit quidem modi officiis labore beatae volupta sequi? Impedit fuga quod tempore vero illum porro nam accusantium, volup enim, in commodi blanditiis eveniet nesciunt numquam, ipsum ex iure moll aperiam magni ipsa consequuntur, nihil at molestias quia nostrum aliquam distinctio ducimus qui! Odit possimus consectetur, recusandae, aperiam null

pariatur at perferendis, molestiae sunt tenetur? Explicabo ducimus nemo reprehenderit vel ab reiciendis iure, dis itaque provident harum? Officiis fugiat natus impedit, iure eum veniam eaque provident. Veritatis a ipsa pr exercitationem deserunt, dolores adipisci harum, eius fuga? Nam, mollitia voluptate! Illum, expedita voluptatibus : laborum iusto, possimus hic facilis blanditiis optio quis deserunt ab nobis esse? Dolorem harum omnis maiores se Corporis, necessitatibus? Eum asperiores voluptatibus ad at repellat perspiciatis laboriosam non itaque.

Question 11

Une autre forme d'usage des flottants, l'affichage multi-colonnes. Il existe d'autres approches plus intuitives et plus simples pour obtenir un résultat similaire.

Pas d'inquiétude donc si vous rencontrez quelques difficultés pour réaliser cet exercice.
Modifiez la page Flottant3.html afin d'obtenir un résultat similaire à celui présenté ci-dessous.



Les unités de mesure

Vous devez être vigilant sur le choix des unités de mesure. Il en existe de très nombreuses et je n'évoque ici que les plus usitées.

Tout d'abord le pixel, px, unité fixe de base, dont l'inconvénient réside dans l'absence d'adaptation aux choix de l'utilisateur (taille de la police du navigateur) tout comme aux différentes résolutions des unités d'affichage.

Vous devrez donc privilégier des unités de mesure proportionnelles afin de faciliter l'adaptabilité de l'affichage de vos pages aux différentes résolutions (caractère responsiv) ou aux choix liés à l'accessibilité de l'internaute.

Pour les tailles de police, vous pourrez utiliser 2 unités, em ou rem.

A la différence du rem qui rend proportionnelle une taille par rapport à la taille du texte à la racine du document, le em la rend proportionnelle par rapport à la taille de l'élément parent dans lequel il est contenu. Par exemple, vous pouvez définir que la police de l'élément sera de 0.7em soit 70% de la taille de police du parent. La taille de police est toujours héritée.

Par défaut, la taille de base de la police des navigateurs est de 16px = 1 rem.

Les valeurs exprimées en rem ou em sont donc transformées par calcul en pixels. Ainsi, 0.65 rem correspond à 10,4 px.

Soyez cohérent et exprimez de préférence vos marges internes et externes dans les mêmes unités que la police.

Pour l'expression de la taille de vos boîtes, les unités de mesure proportionnelles vh et vw sont extrêmement intéressantes.

Elles représentent pour vh, 1% de la hauteur de la zone d'affichage et vw, 1% de la largeur de la zone d'affichage.

Ainsi, si votre conteneur doit occuper tout l'espace d'affichage disponible, sa hauteur sera de 100 vh et sa largeur de 100 vw.

Autre unité proportionnelle pour vos boîtes est le %, exprimé par exemple aux propriétés height et width. Il s'applique de même aux tailles de police.

L'unité % s'applique sur les valeurs héritées du parent.

Question 12

Un exercice pour découvrir les unités relatives.

Le code source de la page est disponible [ici](#) (cf. [UnitesAffichage.zip](#)).

Soit le code html suivant :

```
<body>
  <main>
    <div class="divRep">
      <p>Caractères rem Niveau 1</p>
      <div class="divRep-N2">
        <p>Caractères rem Niveau 2</p>
      </div>
    </div>
    <div class="divPourc">
      <p>Caractères em Niveau 1</p>
      <div class="divPourc-N2">
        <p>Caractères em Niveau 2</p>
      </div>
    </div>
  </main>
</body>
```

Définissez les styles des différentes zones de la manière suivante :

Les marges internes et externes sont à 0 pour tous les éléments.

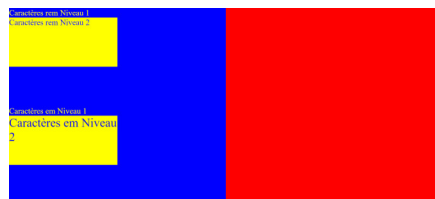
L'élément main doit occuper toute la surface d'affichage disponible. Couleur de fond rouge

La zone divRep doit occuper une place égale à la moitié de la surface dispo et la zone divRep-N2 un quart. Utilisez les unités vh et vw. La taille de police sera de 1.25 rem

Les zones divPourc et divPourc-N2 doivent occuper les mêmes surfaces mais l'unité retenue ici doit être le % et la taille de police de 1.25 em

Définissez les couleurs de fond et de premier plan de manière à visualiser les emplacements de vos boîtes et le texte présent.

Vous devriez obtenir un résultat similaire à l'image suivante :



Vous pouvez modifier la taille de la police de base et observer le résultat produit.

Utilisez les fonctions du menu développeur pour tester l'affichage dans différentes résolutions.

✓ Corrigés

Question 1

Définir les règles de style suivantes :

1. Le texte des titres de niveau 1 et 2 doit être en majuscules et centré
2. Le texte des titres de niveau 2 doit être en italique

Redéfinir les styles des liens adjacent à un paragraphe :

1. Suppression de la décoration du lien par défaut

Couleur de fond rouge

Couleur de texte jaune

2. Sur survol du lien

Couleur de fond jaune

Couleur de texte rouge

Redéfinir la présentation de la liste non ordonnée enfant de l'élément **nav** et ses éléments

1. Supprimer les symboles de la liste
2. Afficher les items de la liste en ligne
3. Affecter un fond vert à l'élément nav
4. Affecter la couleur blanche au texte
5. Modifier les liens : supprimer les styles par défaut et affecter la couleur blanche au texte des liens.

? Indice

Je vous conseille d'introduire dans vos recherches quelques mots clés en préfixe pour obtenir rapidement des infos pertinentes.

Exemple : préfixe : mdn css + mots clés cibles (en français ou en anglais) : vous obtenez la documentation Mozilla sur cet effet et depuis ce point étendre votre lecture à Mettre en forme du texte

[mdn css anchor hover](#)

✓ Corrigé

Quelques éléments de correction

Factorisation des styles des titres

Complément pour les styles de niveau 2

Sélecteur éléments adjacents

Définition de chemin permettant

de modifier les enfants

```
h1,
h2 {
  text-align: center;
  text-transform: uppercase;
}
h2 {
  font-style: italic;
}
p + a {
  background-color: yellow;
  text-decoration: none;
  color: red;
}
p + a:hover {
  color: yellow;
  background-color: red;
}
```

```
nav {
  background-color: green;
  color: white;
}
nav > ul {
  list-style-type: none;
}
nav > ul > li {
  display: inline;
}
nav > ul > li > a {
  text-decoration: none;
  color: white;
}
```

Question 2

Pour bien comprendre l'héritage et l'application des styles en cascade, il convient de s'intéresser aux poids des différents sélecteurs.

Définissez les styles affectant les divisions d'espace et les paragraphes de la manière suivante :

1. La couleur de fond de l'élément div est vert
2. Le style de la classe Option1
 - La couleur de fond est en hexa #b10f2f
 - La couleur du texte est bleue
 - Le texte est en italique
3. Le style de la classe Option2
 - La couleur de fond est en RGB 183, 236, 25
 - La couleur du texte est bleu-violet
4. Le style associé à l'élément dont l'identifiant est parap
 - La couleur de texte est noire

Observez le résultat obtenu.

Vous assistez à des mécanismes de fusion et remplacement de styles qui obéissent à des règles définies en fonction du poids des sélecteurs Css

Vous trouverez des infos sur ce sujet dans le lien suivant : [Cours css poids des sélecteurs](#)

Ainsi qu'un calculateur de poids des sélecteurs [Specificity Calculator](#)

✓ Corrigé

La définition des styles et un extrait du résultat obtenu

```
div{
  background-color: green;
}
.option1 {
  background-color: #b10f2f;
  color: blue;
  font-style: italic;
}
.option2 {
  background-color: rgb(183, 236, 25);
  color: blueviolet;
}
#parap {
  color: black;
}
```



[Téléchargez ici l'archive avec la solution \(cf. *SelecteursCor.zip*\)](#)

Question 3

Dans ce premier exercice, observons le positionnement par défaut dans le flux (static).

Télécharger l'exemple qui servira de source à cette partie de l'atelier. [Télécharger](#) (cf. *PositionStatic.zip*)

Il contient une page html, une feuille de style incorporée et des images.

Sauriez-vous expliquer pourquoi certains éléments occupent toute la place alors que d'autres sont collés les uns aux autres sur une ligne.

Vous pouvez aussi constater que des définitions de marges existent alors qu'aucune mise en forme les concernant n'a été définie.

Utilisez pour cela l'option inspecter de votre navigateur et passez en revue les définitions de style.

D'où proviennent ces définitions ?

Doit-on s'en préoccuper ?

✓ Corrigé

Vous trouvez ici des éléments qui sont de type **block** et occupent la totalité de l'espace disponible en largeur (h1,p, div)

Vous avez des éléments comme les span qui sont de type **inline**.

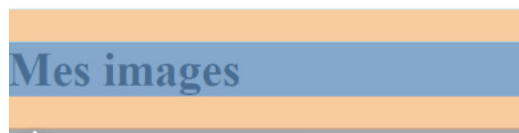
Et les images img qui se comportent un peu comme des inline-block ! Elles se mettent les unes à côté des autres, inline, mais peuvent se voir définir des marges verticales et des tailles...

Les définitions de marge proviennent de la feuille de style du navigateur. Chaque navigateur dispose d'une feuille de style par défaut.

Il est utile d'initialiser les styles afin d'obtenir un résultat homogène d'un navigateur à un autre.

Feuille de réinitialisation des styles à définir en premier dans la liste des feuilles de styles liées à votre page.

[Télécharger feuille de style normalize.css](#)



Par exemple ici, la marge en haut et en bas pour les titres H1 est de 0.67 em

```
h1, h2 {
  background-color: darkgray;
}
h1 {
  display: block;
  font-size: 2em;
  margin-block-start: 0.67em;
  margin-block-end: 0.67em;
  margin-inline-start: 0px;
  margin-inline-end: 0px;
}
```

Question 4

Continuons notre découverte du positionnement.

Reprenons la page exemple précédente, introduisons une liaison vers la feuille de style normalize.css téléchargée.

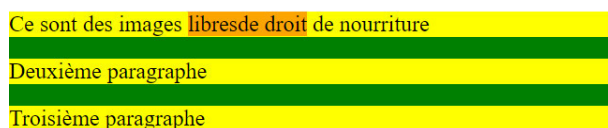
Définissons les marges de l'élément paragraphe, p.

Nous souhaitons que les paragraphes aient une marge extérieure verticale de 15 px en haut et en bas.

Que constatons-nous ?

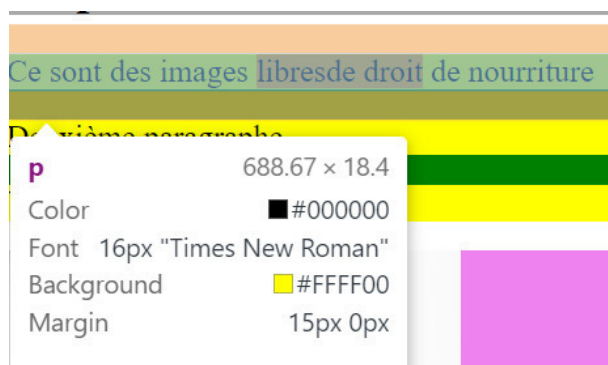
Les marges verticales fusionnent.

Ainsi, l'espace entre le 2ème et premier paragraphe est de 15 et non de 30 (top 15 + bottom 15)



Les marges top et bottom ne semblent pas être appliquées au premier et dernier élément. En fait, elles se trouvent en dehors du contour de la boxe comme le montre l'illustration ci-contre.

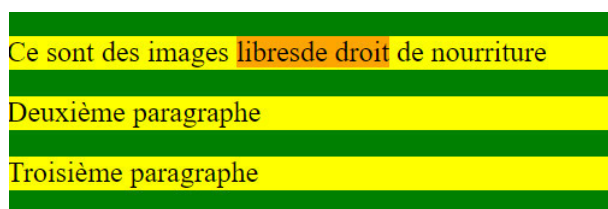
Ce phénomène, assez contre-intuitif, peut être supprimé.



Vous pouvez :

- changer la propriété **overflow** du parent **div** de **visible**, valeur **par défaut**, à tout autre valeur valide. Je choisis ici **hidden**.
- Ou afficher une bordure autour du conteneur **div**.

Essayez les 2 solutions. Elles peuvent être aussi cumulées !



✓ Corrigé

La définition de la feuille de style

```
#zoneTexte {
    background-color: green;
    overflow: hidden;
}

p {
    background-color: yellow;
    margin: 15px 0;
}
```

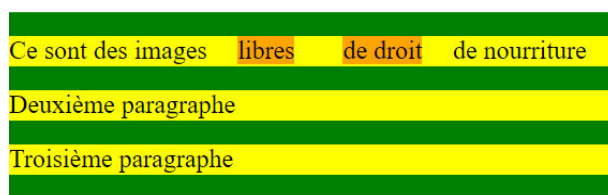
Question 5

Poursuivons avec la définition des marges sur des éléments **inline**, comme l'élément **span**
Définissons des marges de 15 px sur les 4 directions et observons le résultat

Sélectionner les éléments **span** avec l'outil inspecteur.

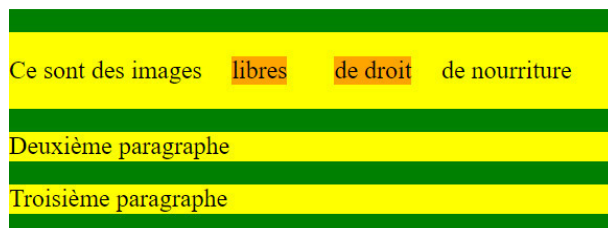
Vous constaterez que les marges horizontales ne fusionnent pas, contrairement aux marges verticales.

Il n'existe pas de marges verticales avec les éléments inline.



Pour obtenir une marge verticale, nous allons modifier le comportement des éléments span et leur disposition en définissant la propriété **display** à **inline-block**.

Nous obtenons alors les marges souhaitées.



✓ Corrigé

La feuille de style complétée. Points importants, les propriétés **display** et **overflow**.

```
#zoneTexte {
    background-color: green;
    overflow: hidden;
}

p {
    background-color: yellow;
    margin: 15px 0;
}

span {
    background-color: orange;
    margin: 15px;
    display: inline-block;
}
```

Question 6

Regardons maintenant du côté des images.

L'élément `img`, défini de base comme `inline`, se comporte en fait comme un `inline-block`.

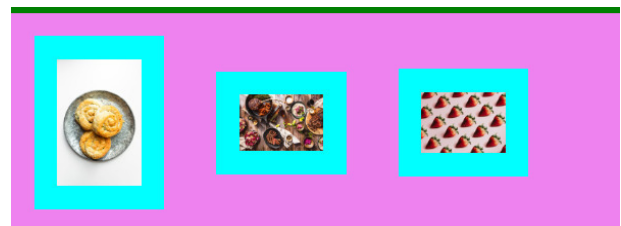
Nous pouvons le positionner verticalement et horizontalement, lui attribuer des marges verticales et réduire ou augmenter l'espace qu'il occupe.

Vérifions en lui appliquant la mise en forme suivante :

- Marge externe de 15 px
- Marge interne de 10 px (encadré)
- Positionnement vertical : milieu
- Largeur uniforme de 55 px (la hauteur sera calculée proportionnellement)

Le résultat est conforme à la définition.

Nos images sont centrées verticalement et des marges internes et externes présentes.



✓ Corrigé

Définition des styles appliqués aux images.

```
#zoneImages {
    background-color: violet
}
img {
    background-color: aqua;
    vertical-align: middle;
    padding: 15px;
    margin: 15px ;
    width: 55px;
}
```

Le téléchargement des pages avec feuilles de styles incorporées.

[Téléchargement \(cf. Parties1_2_Res.zip\)](#)

Question 7

Dans cet exercice, vous devez mettre en œuvre les positionnements relatifs et absolus pour obtenir un résultat analogue à celui présenté ci-dessous.

La boîte 3 de la première série est décalée et vous constatez que son emplacement dans le flux initial est préservé. Ainsi, la boîte suivante se trouve dans le flux à sa position normale (static).

L'un des usages du positionnement relatif est le décalage de portions de texte pour créer des effets.

Le positionnement absolu fonctionne différemment. La boîte 7 ne se trouve plus dans le flux mais en dehors, positionnée par rapport à son conteneur.

Le conteneur est positionné en absolu.

Complétez la définition des styles de la page PositionRelAbs.html.

Positionnement relatif

Positionnement absolu

Positionnements relatif et absolu

Relatif



La position relative peut être utilisée pour

décaler
du texte mis ainsi

en exergue

Absolu



Vous pouvez jouer sur la propriété z-index pour mettre en arrière plan les éléments positionnés.

✓ Corrigé

Quelques éléments de correction

```
div {
  display: inline-block;
  margin-left: 5px;
  width: 150px;
  height: 150px;
  background-color: greenyellow;
}

#boite3 {
  position: relative;
  color: white;
  top: -35px;
  left: 35px;
  background-color: rgb(0, 119, 255);
}

span {
  display: inline-block;
  position: relative;
  top: -1em;
  left: 1em
}
```

```
#conteneur {
  position: absolute;
  width: 100%;
  background-color: white;
  top: 60%
}

#boite7 {
  position: absolute;
  top: 35px;
  left: 75px;
  background-color: rgb(0, 119, 255);
}
```

Question 8

Dans l'exercice suivant, mise en œuvre des positionnements fixed et sticky.

Ils ont un objectif similaire : conserver un élément en position fixe même en cas de débordement de contenu.

Toutefois, l'usage du positionnement sticky est dans la plupart des cas plus simple et préférable au positionnement fixed.

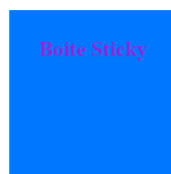
Usage très courant : positionnement des barres de navigation en haut ou en bas de la page.

Complétez la définition des styles des pages PositionFixed.html et PositionSticky.html afin d'obtenir le résultat tel qu'illustré ci-dessous.

Positionnement Fixed



Positionnement Sticky



Positionnement adhérent Sticky

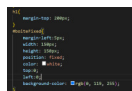
Lorem ipsum dolor, sit amet consectetur adipisicing elit. Quia sapiente voluptatem blanditiis praesentium magnam libero obcaecati dolores nemo? Voluptatem repudia cum eum ducimus harum rem fugiat possimus consequuntur sunt ea voluptatum, ac eligendi. Voluptatem reprehenderit, omnis, vitae provident consequatur ut quisquam

Vous noterez que le positionnement **sticky** conserve l'élément **dans le flux** alors que l'élément **fixed** est **en dehors du flux**. Pour que le titre soit correctement affiché et non masqué dans ce dernier cas, il faut jouer sur les marges du titre : cette option n'est pas judicieuse et peut rendre la mise au point du positionnement des autres éléments plus complexe.

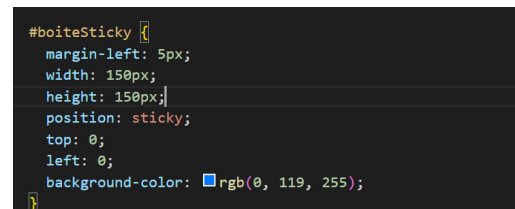
✓ Corrigé

Quelques éléments de correction.

Positionnement Fixed



Positionnement Sticky



Question 9

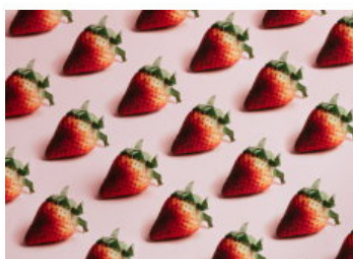
Le plus courant des usages : un texte entourant une image ou une lettrine.

Coder la feuille de style incorporée de la page Flottant1.html afin d'obtenir le résultat présenté ci-dessous.

Positionnement Flottant



distinctio. Consequatur pariatur at perferendis, molestiae sunt tenetur? Explicabo ducimus nemo reprehenderit vel ab rei laborum neque nesciunt itaque provident harum? Officiis fugiat natus impedit, iure eum veniam eaque provident. Veritas voluptatum vitae exercitationem deserunt, dolores adipisci harum, eius fuga? Nam, mollitia voluptate! Illum, expedita voluptatem quae mollitia quidem laborum iusto, possimus hic facilis blanditiis optio quis deserunt ab nobis esse? Dolorem harum officiis ipsam inventore minus. Corporis, necessitatibus? Eum asperiores voluptatibus ad at repellat perspiciatis laboriosae



Lorem ipsum dolor, sit amet consectetur adipisicing elit. Vero quisquam eum dolorum Nesciunt mollitia officia perferendis, fugit expedita nulla non architecto impedit, aut

Question 10

Que se passe-t-il si nous inversons les groupes image/texte ?

Nous obtenons alors le résultat présenté ci-dessous qui ne correspond pas à notre objectif.

Ceci est dû au fait que le texte associé à l'image ne débord pas du bloc image.

Modifiez la feuille Flottant2.html et sa feuille de style incorporée pour obtenir le résultat présenté ci-après.

Positionnement Flottant



Lorem ipsum dolor, sit amet consectetur adipisicing elit. Vero quisquam eum dolor Nesciunt mollitia officia perferendis, fugit expedita nulla non architecto impedit, aut



Lorem ipsum dolor, sit amet consectetur adi quia quos aliquid nisi a, nostrum saepe! N impedit, autem magnam optio, id quasi maic magni officiis assumenda nam! Cupiditate q Placeat nobis porro mollitia blanditiis, volup non! Obcaecati fugit temporibus itaque delect unde, eveniet tempora voluptatum expedita h porro nam accusantium, voluptatum rem dele in commodi blanditiis eveniet nesciunt numq

recusandae ratione minus? Vitae aperiam magni ipsa consequuntur, nihil at molestias quia nostrum aliquam sit soluta distinctio ducimus qui! Odit possimus consectetur, recusandae, aperiam nulla totam ullam debitis suscipit vero architecto c tenetur? Explicabo ducimus nemo reprehenderit vel ab reiciendis iure, distinctio doloremque aliquid, esse corrupti minin natus impedit, iure eum veniam eaque provident. Veritatis a ipsa provident eaque quis, accusantium suscipit ut possimus eius fuga? Nam, mollitia voluptate! Illum, expedita voluptatibus aut quisquam nam velit, obcaecati temporibus id perspi optio quis deserunt ab nobis esse? Dolorem harum omnis maiores sequi, error pariatur, tenetur autem porro molestiae, voluptatibus ad at repellat perspiciatis laboriosam non itaque.

Après correction, en appliquant les mécanismes de dégage ment, nous obtenons le résultat souhaité.

Positionnement Flottant



Lorem ipsum dolor, sit amet consectetur adipisicing elit. Vero quisquam eu Nesciunt mollitia officia perferendis, fugit expedita nulla non architecto imp



Lorem ipsum dolor, sit amet consectetur adipisicing elit. Vero quisquam eu Nesciunt mollitia officia perferendis, fugit expedita nulla non architecto in aliquid excepturi rem placeat animi magni officiis assumenda nam! Cupic Placeat nobis porro mollitia blanditiis, voluptates quisquam distinctio sunt, i delectus odio minus dolores sit quidem modi officiis labore beatae volupta sequi? Impedit fuga quod tempore vero illum porro nam accusantium, volup enim, in commodi blanditiis eveniet nesciunt numquam, ipsum ex iure moll aperiam magni ipsa consequuntur, nihil at molestias quia nostrum aliquam distinctio ducimus qui! Odit possimus consectetur, recusandae, aperiam null

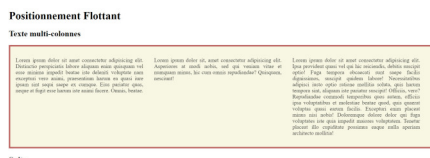
pariatur at perferendis, molestiae sunt tenetur? Explicabo ducimus nemo reprehenderit vel ab reiciendis iure, dis itaque provident harum? Officiis fugiat natus impedit, iure eum veniam eaque provident. Veritatis a ipsa pr exercitationem deserunt, dolores adipisci harum, eius fuga? Nam, mollitia voluptate! Illum, expedita voluptatibus e laborum iusto, possimus hic facilis blanditiis optio quis deserunt ab nobis esse? Dolorem harum omnis maiores se Corporis, necessitatibus? Eum asperiores voluptatibus ad at repellat perspiciatis laboriosam non itaque.

Question 11

Une autre forme d'usage des flottants, l'affichage multi-colonnes. Il existe d'autres approches plus intuitives et plus simples pour obtenir un résultat similaire.

Pas d'inquiétude donc si vous rencontrez quelques difficultés pour réaliser cet exercice.

Modifiez la page Flottant3.html afin d'obtenir un résultat similaire à celui présenté ci-dessous.



✓ Corrigé

Le code source du corrigé sur le positionnement flottant téléchargeable [ici](#) (cf. [FlottantsCor.zip](#)) (cf. [FlottantsCor.zip](#)).

Question 12

Un exercice pour découvrir les unités relatives.

Le code source de la page est disponible [ici](#) (cf. [UnitesAffichage.zip](#)).

Soit le code html suivant :

```
<body>
  <main>
    <div class="divRep">
      <p>Caractères rem Niveau 1</p>
      <div class="divRep-N2">
        <p>Caractères rem Niveau 2</p>
      </div>
    </div>
    <div class="divPourc">
      <p>Caractères em Niveau 1</p>
      <div class="divPourc-N2">
        <p>Caractères em Niveau 2</p>
      </div>
    </div>
  </main>
</body>
```

Définissez les styles des différentes zones de la manière suivante :

Les marges internes et externes sont à 0 pour tous les éléments.

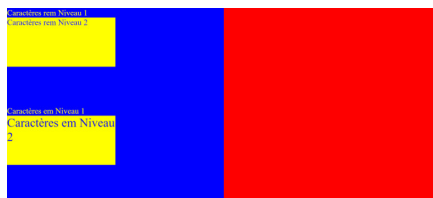
L'élément main doit occuper toute la surface d'affichage disponible. Couleur de fond rouge

La zone divRep doit occuper une place égale à la moitié de la surface dispo et la zone divRep-N2 un quart. Utilisez les unités vh et vw. La taille de police sera de 1.25 rem

Les zones divPourc et divPourc-N2 doivent occuper les mêmes surfaces mais l'unité retenue ici doit être le % et la taille de police de 1.25 em

Définissez les couleurs de fond et de premier plan de manière à visualiser les emplacements de vos boîtes et le texte présent.

Vous devriez obtenir un résultat similaire à l'image suivante :



Vous pouvez modifier la taille de la police de base et observer le résultat produit.

Utilisez les fonctions du menu développeur pour tester l'affichage dans différentes résolutions.



Corrigé

Éléments de correction téléchargeables [ici](#) (cf. *UnitesAffichageCor.zip*)



Œuvre collective de l'AFPA

Sous le pilotage de la Direction de l'Ingénierie.

© AFPA

Reproduction interdite

Article L 122-4 du code de la propriété intellectuelle.

« Toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite. Il en est de même pour la traduction, l'adaptation ou la transformation, l'arrangement ou la reproduction par un art ou un procédé quelconque. »

Date de mise à jour 15/06/2022