

TP3 : CSS

Animations et positionnement

Objectif :

- Positionner des bocs en CSS avec les positionnements relatifs, absolus, flottants et « sticky »
- Utiliser les propriétés d'animations en CSS3 et les préfixes vendeurs

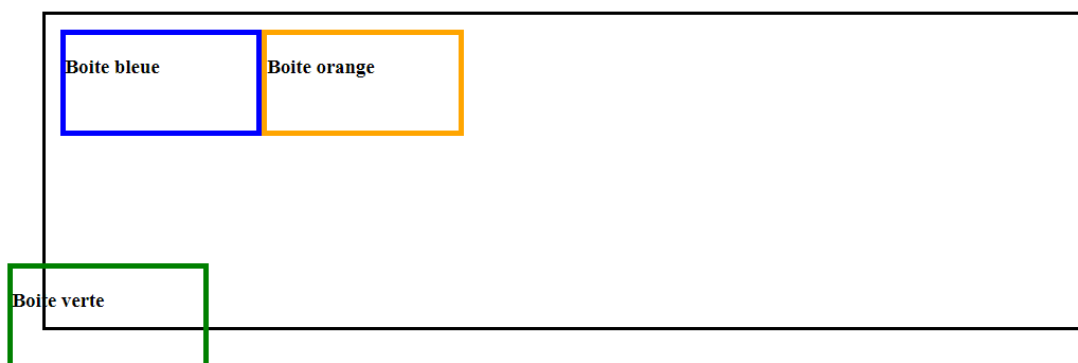
Consignes :

- ❖ Pour chaque exercice, il faut bien examiner le code HTML et CSS fourni avant de commencer à faire l'exercice
- ❖ Téléchargez le dossier TP3 et décompressez le

Exercice1 : Placement de blocs avec le positionnement : relative

Rappel : Position : relative : Décalage par rapport à la position originale dans le flux

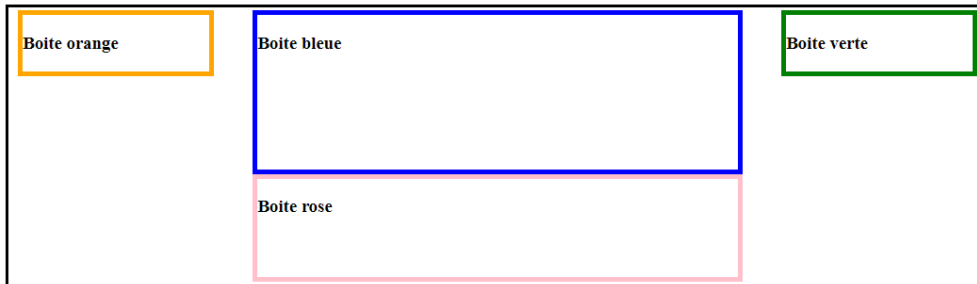
- Considérez le premier conteneur div avec l'id « exercice1 » dans TP3.html
- Complétez le style css du premier exercice comme indiqué sur la feuille de style TP3.css, afin d'obtenir le résultat illustré ci-dessous.



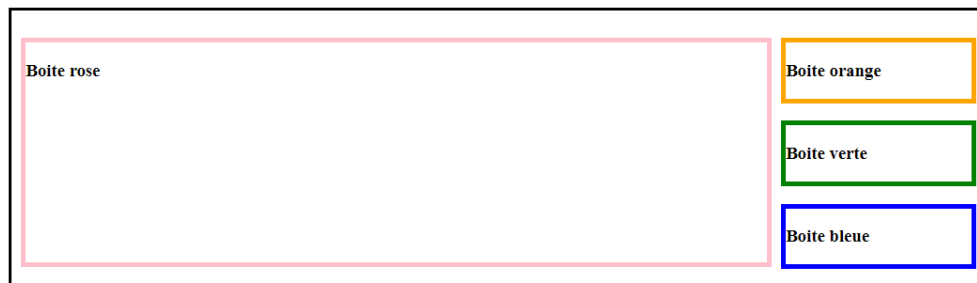
Exercice 2 : Placement de blocs avec le positionnement : absolute

Rappel : Position : absolute : Sort du flux; Se place par rapport au premier ancêtre positionné Ou par rapport à body si aucun ancêtre n'est positionné

- 1) Complétez le style css du deuxième exercice afin d'obtenir le résultat illustré ci-dessous.



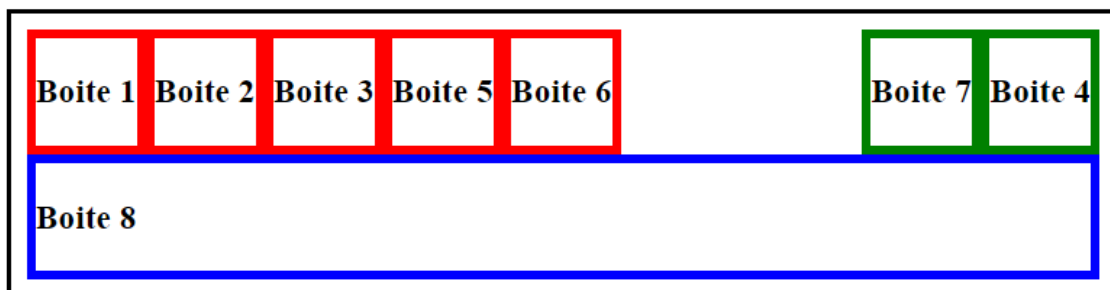
- 2) Toujours avec des positionnements absolus placez les boîtes de façon à obtenir le résultat illustré ci-dessous. **Ne modifiez pas les valeurs de la largeur et la hauteur.**



Exercice 3 : Placement de blocs avec le positionnement flottant

Rappel : Positionnement flottant : Sort du flux pour se placer à gauche ou à droite du **bloc qui le contient** (avec `float: left;` ou `float: right;`). La propriété `clear: both;` permet au bloc suivant de ne plus être soumis aux flottements précédents et de se placer **en dessous** des flottants précédents.)

- 1) Complétez le code css comme indiqué sur la feuille de style, afin d'obtenir le résultat ci-dessous



2) Complétez le code css de façon à obtenir le résultat illustré après.

Aide : Vous avez juste besoin d'ajouter des propriétés de positionnement flottant. C'est-à-dire **float:left | right;** ou **clear : left | right | both;**

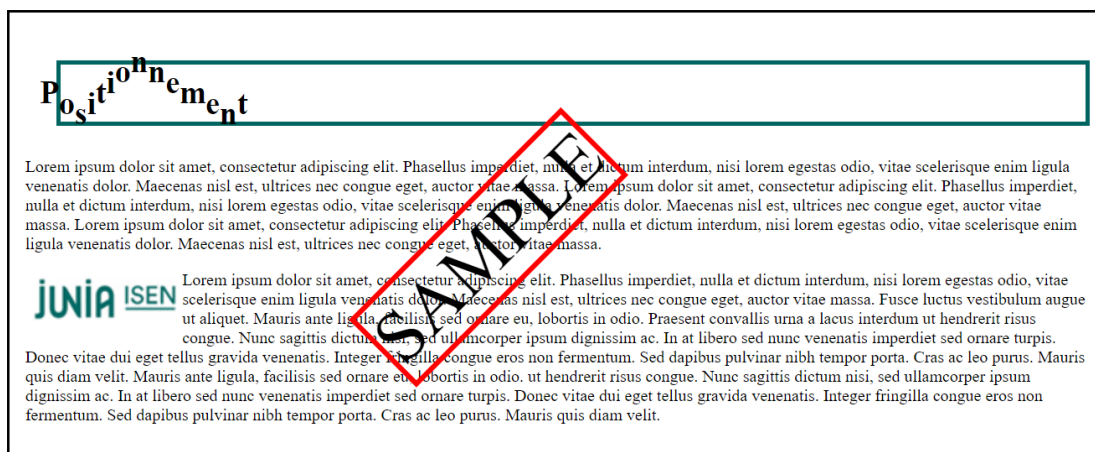


Exercice 4 : Positionnement relatif, absolu et flottant

Complétez le code CSS pour obtenir le résultat ci-dessous :

Aide :

- Pour le titre h3 : marge externe à gauche de 30px, marge interne de 10px, taille de police de 30px.
- Positionnez chaque lettre du titre avec un positionnement relatif, et appliquez des décalages avec des pas de 10px.
- Pour l'image : utilisez un positionnement flottant
- Pour le cadre « sample » : bordure de 5px, taille de police de 50px, rotation de -45deg : **transform : rotate(-45deg)** et positionnement absolu. Attention : le positionnement absolu doit être par rapport au parent dont l'id est **Exercice4**
- Utilisez les préfixes nécessaires pour la propriété transform : consultez le site <https://caniuse.com/> pour vérifier le support des navigateurs, et ajouter les préfixes manuellement ou à l'aide des sites comme <https://autoprefixer.github.io/>



Exercice 5 : Placement avec le positionnement « **sticky** » et la propriété **overflow**

- Ajoutez le positionnement **sticky** au titre **h3** avec un décalage en haut (top) de -10px
- Ajoutez le positionnement **sticky** aux éléments **dt** en précisant la valeur adéquate du décalage pour avoir le résultat souhaité comme montré sur la page html.
- Appliquez une **translation** de -10px aux éléments **dd** lors du survol de la souris (fonction : **translateX()**)
- Donnez à la boîte de l'exercice5 une hauteur de 500px. Remarquez que le contenu déborde de la boîte.
- Afin de gérer ce problème de débordement, on doit modifier la valeur de la propriété **overflow**

➔ Documentation sur [Overflow](#)

Ajouter à la boîte de l'exercice5 la propriété **overflow** avec la valeur **scroll**, afin d'afficher la barre de défilement.

- On souhaite masquer la barre de défilement tout en gardant la possibilité de défiler la liste des noms. Consultez la documentation [suivante](#) et ajoutez le code adéquat, remarquez que cette fonctionnalité n'est supportée par les navigateurs qu'en utilisant des préfixes vendeurs et en utilisant différentes propriétés.

Exercice 6 : Animations CSS3 simple et la fonction **calc()**

- Utilisez la classe **cercle** afin de transformer les trois éléments en cercle : donnez la valeur 50% à la propriété border-radius.
- Mettez les cercles en positionnement absolu par rapport à leur conteneur
- Utilisez la classe **anim1** afin d'animer le cercle A :
 - Décalez de gauche de 80px
 - Déclarez une animation nommée : **animation-1** d'une durée de 2 secondes
 - Utilisez la propriété **animation-iteration-count** pour faire en sorte que l'animation tourne en continu (valeur **infinite**)
- Préparez les **keyframes** de l'animation **animation-1** avec deux étapes **from** et **to**.

- L'état de départ : appliquez un décalage du top de 0px
- L'état d'arrivée : appliquez un décalage du top de 160px (hauteur du conteneur **#Exercice6** 200px – hauteur du cercle 40px)
- Remarquez que le cercle déborde du bas ! pourquoi à votre avis ? Utilisez l'outil « inspecter » pour vous aider.
- **Réponse :**
Le cercle déborde car : vu la propriété **box-sizing :border-box** ; la hauteur 200px inclut la hauteur du contenu et aussi l'épaisseur des bordures inférieure (2px) et supérieure (2px) ; afin que le cercle ne déborde pas il faut mettre : $(200px - 2*2px) - 40px = 156px$ → changez la valeur de décalage du haut de 160px à 156px
Afin d'éviter de faire ces calculs à chaque fois, on va utiliser la fonction CSS **calc()** → changez la valeur du décalage du haut et mettre : **calc(100% - 40px)** ;
⇒ Documentation sur la fonction [calc\(\)](#)
- Utilisez la propriété **animation-direction** pour faire en sorte que l'animation se déroule dans un sens puis dans l'autre de manière alternative (valeur : **alternate**).
- Utilisez la classe **anim2** afin d'animer le cercle B :
 - Décalez du haut de : **calc(100% - 40px)**;
 - Déclarez une animation nommée : **animation-2** d'une durée de 2 secondes, qui tourne en continu de manière alternative
- Préparez les **keyframes** de l'animation **animation-1** avec deux étapes **from** et **to**.
 - L'état de départ : appliquez un décalage du top de **calc(100% - 40px)** et à gauche de 0px, et une rotation de 0deg.
 - L'état d'arrivée : appliquez un décalage du top de 0px à gauche de **calc(100% - 40px)** et une rotation de 360deg
- Utilisez la classe **anim3** afin d'animer le cercle C comme montré sur la page HTML
- Ajoutez les **préfixes vendeurs** nécessaires pour les propriétés d'animation, transformation et les keyframes : <https://autoprefixer.github.io/>

Exercice 7 : Diaporama des images

On souhaite créer un diaporama des images, comme montré sur la page HTML.

- Afin de bien visualiser le fonctionnement, on a ajouté des bordures rouge et bleue aux deux conteneurs **#Exercice7** et **.diapo**. Vous pourrez les enlever lorsque l'exercice sera terminé.
- Donnez aux deux conteneurs et aux images une hauteur de 400px.
- Donnez aux (quatre) **images** une largeur de 25% de leurs conteneur (**.diapo**)
- Donnez au conteneur secondaire (**.diapo**) une largeur 4 fois plus grande que son conteneur : 400%
- Associez à **.diapo** l'animation nommée **diaporama** pour une durée de 10s.
- Préparez les **keyframes** de l'animation **diaporama** avec deux étapes **from** et **to**.

- Nous allons effectuer le mouvement en utilisant un simple positionnement relatif. Mettez l'élément **.diapo** en positionnement relatif et en utilisant les keyframes, réglez la position de **.diapo** à 0 pixel de la gauche au temps 0%(**from**) et à -(la largeur de trois images) de la gauche au temps 100%(**to**). PS : largeur de 3 images = 300% la largeur du conteneur principal.
- Ajoutez les propriétés pour que l'animation tourne en continu de manière alternative
- Utilisez la propriété **animation-timing-function** pour avoir un mouvement linéaire sans accélération. (valeur : **linear**)
- Masquez la partie qui déborde du conteneur **#Exercice7** en donnant la valeur **hidden** à la propriété **overflow**. Supprimez les deux bordures

Nous souhaiterions ajouter des états intermédiaires afin que le diaporama défile tout en s'arrêtant un instant sur chaque image.

- Supposant qu'on va s'arrêter 10% du temps sur chaque image et faire la transition vers l'image suivante en 20% du temps :
 - ⇒ Donc l'état « **from** » et le premier état intermédiaire « **10%** » auront la même valeur de décalage 0px afin de pauser sur la première image
Ensuite dans l'état intermédiaire suivant (après 20% du temps et donc c'est 10+20 : 30%) on fera le décalage vers la deuxième image et donc un décalage à gauche égale à : -(la largeur d'une image : -100%) ... etc
 - ⇒ Complétez le code CSS pour avoir le résultat souhaité comme montré sur la page HTML
- Ajoutez les préfixes vendeurs nécessaires pour les propriétés d'animation, les keyframes : <https://autoprefixer.github.io/>

Exercice 8 :

1) Les cartes

- Mettez les images en positionnement absolue par rapport à leur conteneur et appliquez un décalage à gauche de 0px afin de superposer toutes les images.
- Remarquez comment le conteneur s'effondre ! Pourquoi à votre avis ?
- **Réponse** : car il n'y a plus aucun élément/contenu à l'intérieur et le conteneur n'a pas de dimension (remarquez que dans les exercices précédents on n'a pas eu ce problème car on précisait toujours la largeur et la hauteur du conteneur). Donnez alors au conteneur les mêmes dimensions que les images (largeur de 224px et hauteurs de 313px).
- Lors du survol du conteneur, appliquez des rotations aux cinq images de -60deg, -40deg, -20deg, 0deg et 20deg respectivement
- Remarquez que la rotation se fait par rapport au centre des images, déplaçant l'origine du repère du centre des images vers le coin bas gauche : appliquez aux images la propriété **transform-origin : left bottom ;**

- On souhaite agrandir chaque image lors du survol de la souris, ajoutez, en plus de la rotation, un agrandissement de 110% ç-à-d : **scale(1.1)**
- Supprimez la bordure du conteneur.

2) Le message caché

Complétez le code CSS afin d'obtenir le résultat souhaité montré sur la page HTML

Aide :

- Précisez la largeur, la hauteur et la couleur des 4 blocs et utilisez le positionnement absolu afin de les positionner
- Pensez à bien placer le repère des transformations pour chaque bloc
- Pensez à appliquer la transformation en 3 étapes au lieu de 2 à l'aide des keyframes
- Pensez à donner la valeur « **forwards** » à la propriété **animation-fill-mode** afin de conserver l'état final de l'animation après la fin de cette dernière.