**Exercice 3**

**Remise avant la fin de la journée du jeudi 10 février**

**Rubriques du w3schools à utiliser :**

[Css Combinators](https://www.w3schools.com/css/css_combinators.asp)

[Css Specificity](https://www.w3schools.com/css/css_specificity.asp)

[Css Pseudo-class](https://www.w3schools.com/css/css_pseudo_classes.asp)

[Css Pseudo-Element](https://www.w3schools.com/css/css_pseudo_elements.asp)

[Css Align](https://www.w3schools.com/css/css_align.asp)

[Css Link](https://www.w3schools.com/css/css_link.asp)

[Css List](https://www.w3schools.com/css/css_list.asp)

[Css Text](https://www.w3schools.com/css/css_text.asp)

[Css Font](https://www.w3schools.com/css/css_font.asp)

[Css Font Size](https://www.w3schools.com/css/css_font_size.asp)

[CSS Display](https://www.w3schools.com/css/css_display_visibility.asp)

[Css Float](https://www.w3schools.com/css/css_float.asp)

[Css Overflow](https://www.w3schools.com/css/css_overflow.asp)

* Pratiquer les sélecteurs qui sont très importants pour appliquer une règle CSS aux bons éléments d’une page.
* Découvrir d’autres propriétés s’appliquant au texte, aux liens et aux listes.
* Utiliser **index.html** et **disney.css**
* NE PAS ajouter de noms de class ou de id. (Ne pas modifier le fichier index.html)

Les règles css existantes permettent :

* De donner une largeur maximale à la page et de centrer son contenu.
* De fixer la taille de la police à 100%. Cela permettra, par la suite, de donner une taille relative à cette taille de 100%.
* D’ajouter une bordure autour de chaque article et autour de la liste à puces à la fin.

1. En utilisant des tailles en **em** ou en **%**, grossir les titres de type h1 et ceux de type h2 tout en s’assurant que h1 est plus gros que h2.
2. En utilisant un sélecteur pouvant s’appliquer à plusieurs balises, centrer tous les titres.
3. Faire en sorte que tous les titres débutant les sections soient plus gros en hauteur.
4. Faire en sorte que tous les titres débutant les articles soient plus gros en hauteur.
5. Une image est de type **inline**, elle ne peut pas être centrée directement à l’aide de text‑align.Pour cette raison, les images sont à l’intérieur d’une balise <figure>.

Centrer les images.

1. Une façon souvent utilisée pour centrer un paragraphe de texte est d’utiliser les marges de type **auto.**

Pour tous les articles contenant une image :

* Donner une largeur de 75%
* Déclarer les marges comme étant auto .

1. Faire en sorte que tous les noms de personnages qui sont présentement en gras dans les paragraphes, soient écrits plus gros. Le sélecteur doit utiliser la hiérarchie des balises.
2. Utiliser la pseudo-class **:hover** pour changer la couleur de background des divisions comportant une image lorsque la souris survole ces divisions.
3. Utiliser le [pseudo-element](https://www.w3schools.com/css/css_pseudo_elements.asp) **::first-letter** pour que la 1ère lettre des h3 soit plus grosse (style lettrine). Utiliser une des valeurs suivantes pour donner une taille au texte : larger, x-large ou xx-large.
4. L’article parlant d’Alice et celui parlant de la Belle au Bois Dormant possèdent des mots surlignés. Utiliser les [pseudo-class](https://www.w3schools.com/css/css_pseudo_elements.asp) **:first-of-type** et **:nth-of-type** pour que :

* Le 1er mot surligné de chaque paragraphe possède un background vert
* Le 3ième mot surligné de chaque paragraphe possède un background bleu

1. Lire la rubrique [Css Link](https://www.w3schools.com/css/css_link.asp) et faire en sorte que :

* les liens hypertextes ne soient pas soulignés
* le texte des liens soit écrit en vert : doit s’appliquer aux liens visités et non visités
* lorsqu’on place la souris sur un lien alors le texte devienne blanc avec un background noir

1. La rubrique [Css List](https://www.w3schools.com/css/css_list.asp) indique qu’il est possible de modifier la puce d’une liste. Inscrire une règle pour faire disparaître les puces.
2. La règle précédente est automatiquement appliquée au niveau principal et au 2ième niveau de la liste. Faire en sorte que les puces du **2ième niveau** de la liste soit un cercle plein. Attention au sélecteur.

* Comprendre la notion de display et de float. (Ne pas modifier le fichier dumbo.html)

À l’aide de la page **dumbo.html et du fichier dumbo.css**

1. Comprendre le positionnement naturel

**1ère section de la page : positionnement naturel**

Cette section **démontre** que certaines balises sont de type **block** tandis que d’autres balises sont de type **inline**. **Important de bien comprendre.**

* <h3> et <p> sont de type **block** et font un saut de ligne
* <a> et <img> sont de type **inline** et s’affichent un à côté de l’autre

**2ième section: positionnement avec inversion des display**

Cette section **démontre** qu’il est possible de modifier le mode d’affichage d’une balise.

* <h3> et <p> ont une règle css **display :inline**, elles ne font plus de saut de ligne
* <a> et <img> ont une règle **display :block,** elles font donc un saut de ligne

1. Expérimenter le positionnement flottant d’image en créant un habillage de texte autour d’une image (comme dans Word ☺ ). Dans la **3ième section: positionnement flottant d’une image**

* La propriété **float** permet de placer un élément le plus à droite ou le plus à gauche possible. On dit qu’il ne fait plus partie du flux naturel.
* Un élément flottant doit apparaître en 1er dans le code html et ensuite les éléments qui suivent se placent à côté de l’élément flottant.
* Utiliser le bon sélecteur et la propriété **float :right** pour que l’image de dumbo flotte à la droite du 1er paragraphe de texte.

1. Faire flotter le 1er paragraphe à la gauche du 2ième. Dans la **4ième section: positionnement flottant de texte.**

Pour que les règles CSS s’appliquent seulement au 1er paragraphe, utiliser le sélecteur #section4 p :first-of-type dans le fichier css.

* Il faut donner une largeur au 1er paragraphe, sinon il occupera toute la largeur disponible et il n’y aura plus de place à sa droite pour le 2ième paragraphe.

Donner une largeur de 50% au paragraphe.

* Afin de mieux distinguer le 1er paragraphe, lui donner un background-color.
* Faire flotter le paragraphe à gauche. Ainsi le 2ième paragraphe se place automatiquement à sa droite.

1. Créer des problèmes de positionnement flottant.

* **Dans la section 4,** modifier la largeur du paragraphe pour qu’elle soit à **40%**.
* Faire afficher le résultat dans le navigateur :
  + La bordure n’inclut pas toute la hauteur du 1er paragraphe. Cela peut causer des problèmes de présentation.
  + Le titre suivant se place à côté du 1er paragraphe lui aussi!!

1. Résoudre les problèmes occasionnés par le flottement. Dans **5ième section: résoudre les problèmes.**

Pour régler le problème du titre qui se place à côté du 1er paragraphe de la section précédente, il faut arrêter le flottement en utilisant la propriété **clear.**

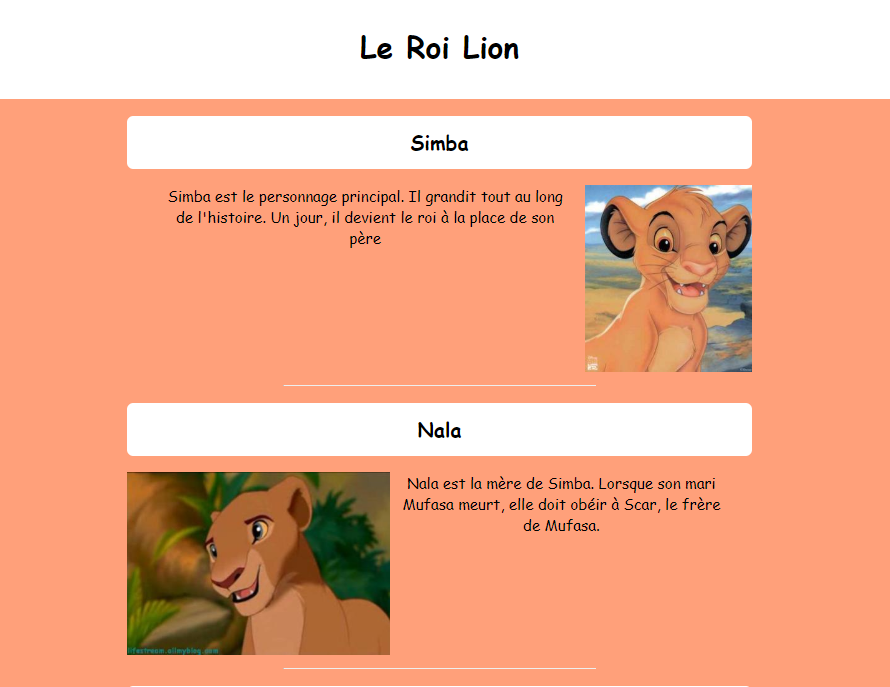
* Inscrire la règle **clear :both;** pour le id section5.
* Faire afficher dans le navigateur pour voir la différence.

Pour régler le problème des 2 blocs de texte qui ne sont pas de la même hauteur, il faut utiliser la propriété **overflow** (<https://www.w3schools.com/css/css_overflow.asp> ) qui permet de contrôler le contenu trop grand pour son conteneur (ex : trop de texte pour la division contenant des dimensions).

* Appliquer les mêmes règles CSS à la section 5 qu’à la section 4 pour reproduire le problème.
* Ajouter la règle **overflow :auto;** au **<div>** de la section 5 pour que le navigateur gère tout ce qui sort de la division.

À l’aide de la page **roi\_lion.html et du fichier roi\_lion.css** (Ne pas modifier le fichier roi\_lion.html)

* Note : 2 fichiers css sont utilisés pour la page roi\_lion.html. Le navigateur applique les règles qui se trouvent dans disney.css et ensuite les règles de roi\_lion.css.
* Reproduire le design qui suit en suivant les directives



1. Faire flotter les images.

* Le code html est déjà structuré afin que les images flottent à côté du contenu. Dans le code, elles sont donc placées avant le texte.
* Une classe est déjà appliquée à chaque image. Il suffit d’utiliser ces classes pour faire flotter les images.
* Écrire les règles CSS pour que les images de la classe droite flottent à la droite de la petite description et que les images de la classe gauche flottent à la gauche des descriptions.

1. Faire afficher le résultat dans le navigateur et … problème : les contenus se chevauchent. Cela est causé par les images qui sont plus grandes que le texte (problème des flottements vus précédemment).

* Appliquer une des solutions suivantes pour régler le problème :
  + **overflow : auto** au parent (class histoire)

**OU**

* + **clear :both** à l’élément qui doit arrêter le flottement (hr)

1. Ajouter des padding afin que le texte ne soit pas collé sur les images.

À l’aide de la page **aladdin.html** et du fichier **aladdin.css** (modifier aladdin.html tel que indiqué)

1. Créer une galerie d’images en les faisant afficher une à côté de l’autre.

* Comprendre la structure du code html
  + Le code contient des divisions appartenant à la classe ‘ligne’. Ces divisions représentent donc 1 ligne d’images.
  + Une ligne contient 3 divisions appartenant à la classe ‘image’. Il y aura donc 3 images sur chaque ligne.
  + Chacune des divisions ‘image’ est constituée de l’image comme telle et d’une description
* Comprendre le css existant
  + Chaque division ‘image’ est dimensionnée en lui donnant une largeur de 30% pour avoir 3 images par ligne.
  + Une marge est laissée entre chaque division afin de ne pas être collée sur la précédente.
* Faire flotter les divisions ‘images’ à la gauche les unes des autres.
* En faisant afficher dans le navigateur on s’aperçoit que cela ne donne pas le résultat souhaité. On a l’impression que les images chevauchent le header…mais ce n’est pas le cas…c’est plutôt le footer qui s’affiche complètement dans le haut! Il est possible de vérifier le tout avec F12.

Il faut appliquer la règle **clear :both** pour arrêter le flottement, ainsi il n’y aura plus rien qui flotte à côté du footer.

* En faisant afficher de nouveau dans le navigateur on voit que le footer s’affiche maintenant dans le bas mais les images ne sont quand même pas bien placées. En fait, il suffit d’arrêter le flottement entre chaque ligne à l’aide de la classe ‘ligne’.

1. Agrandir les images lorsqu’on clique dessus.

* Chaque image affichée est en fait une miniature de l’image. Le dossier ‘images’ contient aussi les images grandeur réelle.

Dans le code html, il faut ajouter un lien hypertexte pour chaque miniature. Le lien doit mener vers l’image grandeur réelle.

* Valider le code html.
* Faire afficher dans le navigateur et tester les liens.
* Changer la couleur de background des divisions ‘image’ lorsque la souris est placée dessus à l’aide de la pseudo-class :hover. Ainsi l’image survolée sera un peu plus en évidence.
* Lorsqu’on se place sur une image, un bakground noir apparaît dans le bas de l’image. C’est parce qu’on avait ajouté une règle dans disney.css dans la partie facultative 1. Il suffit d’annuler cette règle en indiquant d’appliquer le comportement par défaut :

Dans aladdin.css inscire :

a :hover {

background-color : initial;

color : initial; }

NB: pas de modification demandée pour les fichiers blanche\_neige.html et blanche\_neige.css