TP 3

Mise en forme avec CSS

Rappel : le code CSS doit être *séparé* du code HTML, et placé dans un fichier . css externe. Aucun code CSS n'est autorisé dans les pages HTML codées au cours de ce TP; la modification des fichiers HTML fournis n'est en aucun cas autorisée. Aucun code JavaScript n'est demandé au cours de ce TP.



Exercice 3.1 (Débuts avec CSS). Le but de cet exercice est de rédiger le code CSS de textes.css, qui doit mettre en forme le fichier fourni textes.html. L'apparence finale de la page à la fin de cet exercice devra être la suivante :

Textes et couleurs
Voici du texte en serif, taille 0,9em, italique et de couleur #0c7b25. À vous de jouer !
Ceci est du texte sans marge, qui prend 80% de la largeur et avec une couleur de fond #ffe562 ; le texte doit être justifié. Mais, vous savez, moi je ne crois pas qu'il y ait de bonne ou de mauvaise situation. Moi, si je devais résumer ma vie aujourd'hui avec vous, je dirais que c'est d'abord des rencontres, des gens qui m'ont tendu la main, peut-être à un moment où je ne pouvais pas, où j'étais seul chez moi. Et c'est assez curieux de se dire que les hasards, les rencontres forgent une destinée Parce que quand on a le goût de la chose, quand on a le goût de la chose bien faite, le beau geste, parfois on ne trouve pas l'interlocuteur en face, je dirais, le miroir qui
vois alte à availteir. Availteir de la cas pas mon cas, comme je le usais la, plasque moi au contraire, j'ai pu ; et je dis merci à la vie, je lui dis merci, je chante la vie, je danse la vie Je ne suis qu'amour ! Et finalement, quand beaucoup de gens aujourd'hui me disent : « Mais comment fais-tu pour avoir cette humanité ? » Eh bien je leur réponds très simplement, je leur dis que c'est ce goût de l'amour, ce goût donc qui m'a poussé aujourd'hui à entreprendre une construction mécanique, mais demain, qui sait, peut-être simplement à me mettre au service de la communauté, à faire le don, le don de soi
Ceci est du texte sans marge, qui prend 80% de la largeur et avec la même couleur de fond ; le texte doit être justifié. -
La première lettre de chaque paragraphe est trois fois plus large que les autres (donc 3em).
$\mathbf{D}_{\mathrm{\acute{e}brouille-toi}}$ pour que ces pierres n'arrivent jamais au chantier. Pas de pierre, pas de construction. Pas de construction, pas de palais. Pas de palais pas de palais.
Document réalisé dans le cadre du cours de développement Web.

Question 1 : Mettre en forme le titre (balise h1). La police doit être sans-serif, la couleur est #1c3805, et le texte est centré. Propriétés utiles : color, font-family, text-align.

Question 2: Mettre en forme le premier paragraphe (les indications de style sont données dans le contenu du texte!). Utiliser l'identifiant donné dans le code HTML (id="premier") pour y accéder facilement dans le code CSS. Propriétés utiles : color, font-family, font-size, font-style.

Question 3 : Mettre en forme le second paragraphe (les indications de style sont données *dans* le contenu du texte!). Propriétés utiles : margin, width, background-color, text-align.

Question 4 : Mettre en forme le troisième paragraphe. Le changement de taille de la première lettre de chaque paragraphe demandera d'utiliser quelque part le sélecteur :first-letter, avec une propriété font-size choisie comme indiqué dans le texte fourni.

Question 5: Mettre en forme le paragraphe de bas de page (footer). Cette phrase est séparée de ce qui précède (propriété margin-top) d'un espace de 2em, les barres supérieure (border-top) et inférieure (border-bottom) sont de couleur #1c3805 et d'une largeur de 5 pixels, le texte est centré et de couleur #1c3805, et la taille du texte est x-small.



Exercice 3.2 (Mise en forme d'un descriptif de film). Le but de cet exercice est de rédiger le code CSS de film.css, qui doit mettre en forme le fichier fourni film.html. L'apparence finale de la page à la fin de cet exercice devra être la suivante :



Question 1 : Mettre en forme le titre de premier niveau pour qu'il soit centré.

Question 2 : Mettre en forme le titre de second niveau pour qu'il s'affiche en couleur #184f79.

Question 3: Éditer la feuille CSS afin que le texte s'affiche en #1d8791, et plus large que la police de base (utiliser la taille relative larger).

Question 4: Mettre en forme le bas de la page afin d'être conforme à l'affichage. La taille devra être x-small, et la couleur #e5b8bb.

Question 5 : Éditer la feuille CSS afin que le texte en « fluo » s'affiche ainsi (on utilisera un fond #4dc8b9).

Question 6 : Éditer le sélecteur article afin que l'article s'affiche dans un cadre : on utilisera une bordure de 0, 15 em de couleur #184f79. Ajouter un *padding* de 1 em. En outre, le cadre devra mesurer 90 % de la longueur de la page. La couleur de fond est #e2c5a6.

Question 7: Centrer l'article en utilisant margin: auto.



Exercice 3.3 (Mise en forme de puces et autres). Le but de cet exercice est de rédiger le code CSS de puces.css, qui doit mettre en forme le fichier fourni puces.html. L'apparence finale de la page à la fin de cet exercice devra être la suivante :



Question 1: Ajouter la couleur de fond **#eff3e4** à la page.

Question 2: Mettre le titre de niveau 1 en **#0c1735** et en italique.

Question 3 : Justifier les paragraphes.

Question 4 : Éditer la feuille CSS afin que les listes ordonnées soit décalées d'une longueur de 5 em vers la droite, et que le texte venant après soit décalé de 3 em vers le bas. Penser à la propriété margin et ses variantes.

Question 5 : Supprimer l'affichage des puces dans les listes non ordonnées.

Question 6 : Mettre les items des listes ordonnées et non ordonnées en bleu.

Question 7: Mettre la couleur des liens en **#bbc8a6**.

Question 8: Mettre en couleur #3f8374 le texte contenu dans les balises .

Question 9 : Mettre en couleur **#bbc8a6**, gras et italique le texte contenu dans les balises au sein des listes (ordonnées ou non ordonnées); les autres textes dans les balises doivent rester comme spécifié ci-dessus.

Question 10: Mettre en forme le pied de page (police toute petite, alignement à droite).

3/16



Exercice 3.4 (Mise en forme d'une présentation). Le but de cet exercice est de rédiger le code CSS de presentation.css, qui doit mettre en forme le fichier fourni presentation.html. L'apparence finale de la page à la fin de cet exercice devra être la suivante :

f f	Titre de niveau	1	
<u><u></u></u>	Titre de niveau 2		
┶┶┶┶┶┶┶┶┶┶	 Le texte par défaut est en police sans-serif La marge gauche fait 4em La couleur par défaut du texte est le bleu navy Tous les titres et sous-titres sont centrés Le titre de niveau 1 est blanc sur fond navy Le bas de page possède exactement les mêmes attributs que Le texte en <pre></pre> ou de classe tt est en couleur blue et en police à chasse fixe (e le titre de pre monospace)	mier niveau
4	Couleur de fond silver		
5	Ligne impaire en police sans empattement	sans-serif	
-	Ligne paire en police avec empattement	serif	
-	Ligne impaire sur fond bleu	#D0D0FF	
-	Ligne paire sur fond vert	#D0FFD0	
5	Document réalisé dans le cadre du cours de dév	eloppement W	eb.

Question 1 : Mettre toute la page en forme en suivant les indications contenues dans presentation.html, à l'exception de l'image de fond et du tableau.

Question 2: Mettre l'aspect général du tableau en forme : celui-ci doit avoir une largeur de 60 %, et être centré. Attention, pour centrer un tableau, on n'utilise pas text-align, mais :

```
margin-left : auto;
margin-right : auto;
```

Les cellules d'en-tête ont un fond silver.

Question 3: Mettre en forme les lignes paires et impaires en suivant les indications contenues dans presentation.html. Voir la documentation de nth-child, et chercher comment obtenir le sélecteur des lignes respectivement paires et impaires.

Question 4 : Ajouter l'image de fond (l'image spirale.png vous est fournie). Celle-ci ne doit se répéter que verticalement; utiliser à cette fin la propriété background-repeat, en consultant sa documentation.

EXERCICE

Exercice 3.5 (Correction d'une mauvaise pratique). Le but de cet exercice est de corriger le code HTML et CSS de boitePASBIEN.html; ce fichier est très mauvais pour plusieurs raisons :

- le code CSS est imbriqué avec le HTML;
- le titre est écrit en majuscules au sein du code HTML;

— les balises <div> sont utilisées là où des nouvelles balises sémantiques (<article>, <nav>...) pourraient être utilisées.

Question 1 : Dupliquer boitePASBIEN.html vers boite.html; nous travaillerons donc désormais sur boite.html. Créer un fichier vierge boite.css, et lier cette feuille de style depuis boite.html.

Question 2: Transférer toutes les informations de style vers votre fichier boite.css, et ce afin que boite.html ne contienne plus aucun code CSS, c'est-à-dire aucune occurence de style="...". Une fois cette opération terminée, le rendu de boite.html doit être exactement celui de boitePASBIEN.html.

Question 3 : Remplacer dans boite.html « TITRE PRINCIPAL » par « Titre principal », et éditer la feuille CSS pour que le titre s'affiche en majuscules. Vérifier que le rendu de boite.html reste exactement celui de boitePASBIEN.html.

Question 4: Supprimer tous les id de boite.html, en trouvant un bon moyen de s'y référer tout de même depuis votre feuille de style. Penser aux nouvelles balises sémantiques! Vérifier que le rendu de boite.html reste exactement celui de boitePASBIEN.html.

Remarque : l'usage des id n'est pas du tout une mauvaise pratique en HTML; simplement, ici, il peut être évité grâce aux balises dites sémantiques.

Question 5: Nous allons maintenant améliorer le menu. Insérer les trois concepts endessous de « Menu » au sein d'une liste non ordonnée. Puis modifier la feuille CSS afin que la liste soit sans style (propriété list-style), sans marge et sans *padding*. Vérifier que le rendu de boite.html reste exactement celui de boitePASBIEN.html.

Question 6: Nous allons maintenant supprimer la balise (rappel : les balises telles que et devraient être évitées, et remplacées par du CSS). Supprimer cette balise au sein du code HTML et, sans ajouter d'identifiant ni de classe, réintroduire le gras via la feuille CSS. Indice : on peut d'abord mettre en gras (via le CSS) tout l'intérieur du menu, puis « annuler » le gras pour les éléments de la liste. Vérifier que le rendu de boite.html reste exactement celui de boitePASBIEN.html.

Question 7 : Enfin, nous allons de même supprimer la balise . Trouver un mécanisme approprié. Vérifier que le rendu de boite.html reste exactement celui de boitePASBIEN.html.

Faire valider par votre enseignant e

EXERCICE

Exercice 3.6 (Dégradés). Le but de cet exercice est de rédiger le code CSS de degrades.css, qui doit mettre en forme le fichier fourni degrades.html. L'apparence finale de la page à la fin de cet exercice devra être la suivante :



Ce TP présente un aperçu de quelques possibilités de dégradés de couleurs réalisés avec CSS.
Un dégradé vertical de bleu à jaune
Un dégradé horizontal de vert pâle à blanc
Un dégradé en diagonale de haut à droite vers le bas à gauche
Vive la République, vive la France
Rainbow warrior
Un dégradé radial vert et blanc
Armée de l'air et de l'espace
Vive la République, vive la France
Vive la République, vive la France
Oukraïna
Rainbow warrior
Document réalisé dans le cadre du cours de développement Web.

Question 1 : Mettre en forme tous les <div> afin qu'ils aient une largeur de 30 % de la largeur de la page, une hauteur de 4 em, et une marge inférieure (c'est-à-dire une distance avec ce qui suit en bas) de 1 em.

Question 2: Mettre en forme l'<article> afin que la bordure soit bleue de largeur 5 pixels, avec des bords arrondis d'un rayon de 40 pixels. La largeur de l'article doit être la moitié de la largeur de la page, l'espace entre la bordure et le contenu doit être de 2 em, et l'espace avec les éléments qui suivent dessous doit être de 1 em.

Question 3: Mettre en forme le premier dégradé. Utiliser la valeur linear-gradient.

Question 4: Mettre en forme les 2^e et 3^e dégradés. Utiliser to [direction] en premier paramètre de la valeur linear-gradient et consulter en ligne sa documentation.

Question 5 : Mettre en forme les 4^e et 5^e dégradés. Pour plus de deux couleurs dans un dégradé, il suffit de les passer en paramètres les unes à la suite des autres.

Question 6 : Mettre en forme le dégradé radial vert et blanc. Utiliser radial-gradient et consulter en ligne sa documentation.

Question 7 : Mettre en forme le dégradé radial à trois couleurs; il suffit là aussi d'ajouter simplement une troisième couleur.

Question 8: Mettre en forme le 8^e dégradé. L'idée est que la couleur est d'abord un bleu plein (sans dégradé), avant de subir un léger dégradé vers le blanc, suivi d'un blanc plein, suivi d'un léger dégradé vers le rouge, suivi du rouge plein. Il faut donc ajouter aux arguments de linear-gradient des pourcentages après les couleurs. Par exemple, purple 20% 30% indique que la zone du dégradé va rester stable (en violet plein, sans dégradé) sur une portion de 20% à 30% de la largeur (ou de la hauteur). Le dégradé se fera entre le pourcentage de fin d'une couleur et le pourcentage de début de la couleur qui suit. NB : la première couleur et la dernière couleur n'ont qu'un seul pourcentage.

Question 9: Mettre en forme le 9^e dégradé... qui n'est en réalité pas affiché comme un dégradé (plutôt comme une suite de couleurs pleines), mais dont les couleurs sont toujours obtenues grâce à linear-gradient. Il suffit de reprendre la technique de la question précédente, de façon à ce que le pourcentage de début d'une couleur soit exactement le même que le pourcentage de fin de la couleur précédente.

Question 10 : Mettre en forme les deux derniers dégradés avec la même technique.



Faire valider par votre enseignant∙e

Question (optionnelle) 11 : Les véritables couleurs du drapeau français ne sont *pas* «bleu» (au sens de rgb(0, 0, 255), «blanc» et «rouge» (au sens de rgb(255, 0, 0). Trouver les couleurs exactes du drapeau français, et les utiliser dans votre feuille de style.



Exercice 3.7 (Positionnement). Le but de cet exercice est de rédiger le code CSS de trois feuilles successives, qui devront mettre en forme de façon différente le fichier fourni positionnement.html.

Première mise en forme

Nous allons rédiger le code CSS d'une première mise en forme nommée positionnement1.css. Cette mise en forme est élégante et relativement ergonomique, et de taille entièrement dynamique. Le positionnement des différents blocs est absolu.

L'apparence finale de la page à la fin de cette partie devra être la suivante :



Question 1 : Imposer un positionnement fixé (position : fixed;) aux trois grandes parties <header>, <nav> et <footer>. Imposer qu'elles soient positionnées complètement sur la gauche (c'est-à-dire avec une distance de 0), en utilisant la propriété left.

Question 2: Centrer le texte du <header> et du <footer>. Leur imposer une largeur de 100 % de la largeur de la page.

Question 3 : Positionner le <header> complètement en haut, c'est-à-dire à 0 du haut (en utilisant la propriété top). Définir une hauteur de 10%. La couleur de fond est jaune.

Question 4 : Positionner le <nav> à 10% du haut (avec la propriété top), c'est-à-dire immédiatement sous le <header> puisque celui-ci a une hauteur de 10%. Définir une largeur de 15%. Définir un *padding* avec le haut de 2% (pas très critique dans cet exercice). Définir la couleur de fond.

Question 5 : Positionner le <footer> complètement en bas (avec la propriété bottom à 0). Définir une hauteur de 6 % et aucune bordure. Définir la couleur de fond.

Question 6 : Nous allons enfin définir les propriétés du <main>. Faire en sorte que l'alignement des paragraphes soit justifié. Définir la couleur de fond. Imposer un *padding* gauche et droite de 1 em. Définir une bordure de 0. Indiquer les deux éléments suivants, nécessaires au positionnement :

```
position : absolute;
overflow : auto;
```

Il reste à gérer le positionnement absolu et les dimensions. Le <main> doit être à 10 % du haut (donc immédiatement sous le <header> qui a une hauteur de 10 %), et doit être à 15 % de la

gauche (donc immédiatement à gauche du <nav> qui fait une largeur de 15%). Sa hauteur doit être de 84% (car 100% moins 10% pour le <header> moins 6% pour le <footer>). Sa largeur doit être la suivante :

```
width : calc(85% - 2em); /* calc permet d'effectuer un calcul en CSS */
/* attention, il faut absolument mettre des espaces de part et d'autre
du '-' */
```

Justifier pourquoi.

Seconde mise en forme

Nous allons rédiger le code CSS d'une seconde mise en forme nommée positionnement1b.css. Cette mise est très similaire à la première, mais avec des bordures qui vont compliquer le calcul des positionnements.

L'apparence finale de la page à la fin de cette partie devra être la suivante :



Question 7 : Dupliquer positionnement1.css vers positionnement1b.css. Lier positionnement1b.css depuis la page HTML.

Question 8: Changer les bordures ainsi :

- <header> : couleur orange, largeur de 6 pixels
- <nav> : couleur bleue, largeur de 4 pixels
- <main>: couleur verte, largeur de 9 pixels
- <footer> : couleur violette, largeur de 5 pixels

Constater l'affichage : pourquoi les différents éléments se chevauchent un peu?

Question 9 : Corriger toutes les dimensions nécessaires : la plupart des propriétés height et width (et seulement celles-ci) devront être modifiées (avec un calc(...) pour prendre en considération la taille des bordures. Aucune bordure ne devra se chevaucher.



Faire valider par votre enseignant e

Seconde mise en forme (variante pour les plus rapides)

Question (optionnelle) 10: Sans rien changer au code HTML, changer la feuille CSS pour que :

- le <header> et le <footer> soient inversés (<header> en bas, <footer> en haut)
- le <nav> et le <main> soient inversés (<nav> à droite, <main> à gauche)
- le <nav> s'affiche verticalement au milieu, c'est-à-dire à mi-distance entre le <footer> et le <header>.

Bien penser à changer les valeurs de positions nécessaires.

Troisième mise en forme (pour les plus rapides)

Nous allons rédiger le code CSS d'une autre mise en forme nommée **positionnement2.css**. Cette mise est complètement différente de la première, ce qui montre la puissance des feuilles de style.

L'apparence finale de la page à la fin de cette partie devra être la suivante :



Question (optionnelle) 11 : Dupliquer positionnement1.css vers positionnement2.css.Lier positionnement2.css depuis la page HTML.

Question (optionnelle) 12: Spécifier les propriétés de <body> : la position doit être fixée (valeur fixed), et le <body> doit se situer à 20 % du haut et à 20 % de la gauche de la page. En outre, sa hauteur et sa largeur font toutes deux 60 % de la page. Sa bordure est noire de largeur 2 pixels. Son rayon de bordure courbe est 20 pixels. Ajouter sa couleur de fond.

Question (optionnelle) 13: Spécifier des propriétés communes à <header>, <footer>, <nav> et <main> : indiquer une position absolue (valeur absolute), une largeur de 100 %, et une distance gauche de 0.

Question (optionnelle) 14 : Le menu (<nav>) doit être tout en haut, et mesurer 5 % de hauteur. Sa taille de police est x-small.

Question (optionnelle) 15 : L'en-tête (<header>) doit être sous le menu (trouver la bonne distance de top) et mesurer 5 em de hauteur. Centrer le texte et définir la couleur de fond (noire).

Question (optionnelle) 16 : Le pied de page (<footer>) doit être tout en bas (distance de 0 du bas) et mesurer 4 % de hauteur. Aligner le texte à droite.

Question (optionnelle) 17 : Le titre de premier niveau doit être jaune et en majuscules.

Question (optionnelle) 18: Spécifier les propriétés du <main> : choisir les bonnes valeurs de top et height afin que cet élément prenne exactement l'espace laissé libre entre les autres éléments (il va falloir utiliser calc(...)). Ajouter overflow : hidden; pour empêcher le texte de déborder. Spécifier la couleur de fond, et le fait que le texte doit être en police à chasse fixe.

Question (optionnelle) 19 : Mettre en forme les titres de second et troisième niveaux du <main>. Spécifier une marge de 0 (pour empêcher un espace entre les éléments qui précèdent ou suivent). Spécifier la couleur de fond (noire). Spécifier la couleur du texte (orange). Définir le *padding* haut, bas et gauche à respectivement 1 em, 1 em et 2 em pour <h2>, et 0,5 em, 0,5 em et 6 em pour <h3>.

Question (optionnelle) 20 : Justifier le texte des paragraphes du <main>. Définir un *padding* gauche et droite de 1 em.

Question (optionnelle) 21 : Spécifier une taille de 2 em pour la première lettre de chaque paragraphe du <main>.

Question (optionnelle) 22: Rendre horizontal le menu du <nav> : supprimer les puces (propriété list-style-type à définir pour ul), puis définir l'horizontalité par float : left pour li. Enfin, séparer les différents items en ajoutant à li une marge droite de 2 em.

Question (optionnelle) 23 : Décaler le mot « Contenu » à droite (80 % de la gauche), avec en outre une position absolue.



Exercice 3.8 (Agrandissement d'une image). Le but de cet exercice est de rédiger le code CSS de agrandissement.css, qui doit mettre en forme le fichier fourni agrandissement.html. L'apparence finale de la page à la fin de cet exercice devra être la suivante :



Question 1 : Mettre en forme l'image dans le <h1> (et seulement celle-ci) comme suit : sa hauteur doit être la même que celles des lettres. Elle doit précéder le texte par un espace de la même largeur qu'une lettre.

Question 2 : Mettre en forme l'image dans le corps de la page (et seulement celle-ci) comme suit : sa largeur doit être de 150 pixels. Le rendu est alors le même que sur la capture ci-dessus.

Question 3 : Lorsque l'on passe la souris au-dessus de l'image dans le corps de la page, sa taille doit passer à 600 pixels. Rappel : le sélecteur lorsque l'on passe la souris sur un certain élément est : hover.

Le rendu doit alors être conforme à la capture ci-après :



Question 4 : Nous allons mettre en place des animations grâce à CSS :

- lorsque l'on passe la souris sur l'image, sa taille doit graduellement augmenter en prenant l seconde (propriété : transition);
- lorsque l'on sort la souris de l'image, sa taille doit graduellement diminuer en prenant 0,2 seconde, mais seulement après un temps d'attente de 0,5 seconde (propriétés : transition-duration et transition-delay).

NB : les animations demandées n'apparaissent bien entendu pas sur la capture d'écran ci-dessus...





Exercice 3.9 (Menu avec images animées). Le but de cet exercice est de rédiger le code CSS de menu.css, qui doit mettre en forme le fichier fourni menu.html. L'apparence finale de la page à la fin de cet exercice devra être la suivante :



NB : les animations demandées n'apparaissent bien entendu pas sur la capture d'écran ci-dessus...

Question 1 : Supprimer toutes les marges de la page (cela va faciliter le bon affichage du menu sur toute la largeur de la page).

Question 2: Supprimer complètement l'affichage du pied de page (<footer>).

Question 3 : Spécifier le positionnement absolu du menu (<nav>) tout en bas de la page, sans marge, sans spécifier de hauteur particulière, en centrant son texte, et avec des bordures haute et basse de 1 em et de couleur jaune et rouge respectivement.

Question 4 : Faire s'afficher de façon horizontale les éléments de la liste dans <nav>.

Question 5 : Spécifier les propriétés des images dans une puce dans le <<u>nav</u>>, et uniquement celles-ci. (Même si, dans le cadre de ce TP, il n'y en a pas d'autres.) L'opacité doit être de 50 %.

Question 6 : Modifier la feuille de style pour que, lorsque l'on passe la souris sur une image, celle-ci voie passer sa taille à 100 pixels. En outre, son opacité devient alors 100 %.

Question 7: Dans l'état actuel, le menu est complètement désorganisé lorsque l'on passe la souris sur une image. Pour corriger cela, l'idée va être qu'une image (marges comprises!) soit exactement de la même taille lorsqu'on passe la souris que lorsque l'on ne passe pas la souris dessus. La taille de l'image elle-même (marges non comprises) reste bien entendu variable (50 pixels sans souris, 100 pixels avec souris).

Par conséquent :

- Spécifier des marges haute, droite, gauche de 0 pour les images sur lesquelles on passe la souris;
- Spécifier des marges haute, droite, gauche de la valeur appropriée pour les images normales, c'est-à-dire sur lesquelles on ne passe la souris. L'idée est que la taille de

l'image (marges comprises) fasse exactement la même taille que lorsqu'on passe la souris dessus, donc 100 pixels.

La capture ci-dessous montre un exemple de rendu lorsque l'on passe la souris sur l'une des icônes.



Question 8 : Lorsque l'on passe la souris sur une image et que sa taille grossit, une transition doit animer ce changement de taille : la durée doit être de 0, 4 seconde. Lorsque l'on enlève la souris de l'image, celle-ci doit revenir à sa taille normale en 0, 1 seconde. Modifier la feuille de style afin de réaliser ces animations.

Faire valider par votre enseignant e



Exercice 3.10 (Images de fond). Le but de cet exercice est de rédiger le code CSS de imagefond.css, qui doit mettre en forme le fichier fourni imagefond.html. L'apparence finale de la page à la fin de cet exercice devra être la suivante :

Images de fond en CSS



Question 1 : Mettre en forme le <main> : hauteur de 480 pixels, largeur de 640 pixels, marges de 10 pixels.

Question 2 : Mettre en forme les titres $\langle h2 \rangle$ et $\langle h3 \rangle$: leur couleur est bleue, et leur opacité est de 75 %. Aligner le $\langle h3 \rangle$ à droite.

Question 3 : Ajouter au <main> trois images de fond de façon à obtenir le rendu. Utiliser les images fournies images/prairie.jpg et images/lapin.png. Attention à l'ordre d'ajout des images! Propriétés utiles : background-image, background-position, background-repeat.

Question 4: Mettre en police x-small et aligner à droite le texte du <footer> : attention, l'extrémité droite du texte doit terminer précisément sous l'extrémité droite de l'image.

Question (optionnelle) 5 : Faire en sorte que le lapin en bas à droite ressemble à un fantôme, c'est-à-dire ait une opacité de 10%.

NB : il n'est pas sûr que ce soit facilement réalisable!



Exercice 3.11 (Formulaire). Le but de cet exercice est de rédiger le code CSS de formulaire.css, qui doit mettre en forme le fichier fourni formulaire.html. L'apparence finale de la page à la fin de cet exercice devra être la suivante :

Mise en forme d'un formulaire

Votre nom	
Prénom	
Votre prénom	
Ville de résidence	
Ville de résidence	
Ville de résidence	

Question 1 : Mettre en forme le <main> de façon à ce qu'il ait une bordure colorée de 1 em de large, une couleur de fond, et un espacement de 1 em entre le contenu et les bords.

Question 2 : À l'aide de la propriété margin, centrer le <main> horizontalement sur la page.

Question 3: À l'aide de la propriété display, faire en sorte que les <label> s'affichent sous forme d'un bloc sur une ligne dédiée, c'est-à-dire que le texte qui suit (le <input>) revient à la ligne.

Question 4: Ajouter un espace de 1 em à la suite de chaque couple (<label>, <input>).

Question 5 : Aligner le bouton « Validation » à droite (indice : propriétés display, margin-left, margin-right).

Question 6 : Faire en sorte que le champ du formulaire qui a le focus (c'est-à-dire celui sur lequel on a cliqué en dernier) ait une couleur de fond (par exemple **#FC1**).

Version du TP : 8 février 2024