

## TP 001 : Une page de A à Z

---

*Préliminaire : créez un fichier index.html et style.css*

*Tout au long de ce TP les instructions nécessitent une recherche ou une interprétation de votre part mais les mots ont été choisis avec attention. Obtenir la solution auprès d'un camarade ou de votre intervenant, sans une recherche préalable de votre part, serait contre-productif.*

### **Exercice 1 : La base**

On va réaliser le layout de notre site qui se composera, si possible, d'éléments ayant une sémantique.

Votre page sera composée de :

- Un bloc d'en-tête
- Un bloc pour la barre de navigation
- Un bloc pour le cœur de la page
- Un bloc pour le pied de page
- Un bloc pour la publicité

Commencez par placer ces 5 blocs dans votre page.

Le contenu des blocs indiquera sa fonctionnalité.

Dans votre fichier css, avec 1 seul sélecteur, mettez un style qui permette à **tous** les éléments de votre page **contenus** dans le **body**, d'avoir une bordure noire de 2 pixels d'épaisseur et une marge extérieure de 10 pixels.

Vous devriez arriver à ce résultat

L'en-tête
Menu
Cœur de la page
pub
Le pied de page

Si vous observez des bordures supplémentaires, modifiez le sélecteur css pour mieux viser les éléments voulus.

## Exercice 2 : La grille

Vous allez maintenant placer vos éléments grâce aux grilles css (grid layout).

Vous aurez besoin pour cela de la documentation : <https://www.w3schools.com/cssref/>, ou, si vous préférez, d'un moteur de recherche.

Les propriétés qui vous intéressent commencent par grid-xxx ; en y regardant de plus près, vous pourrez diviser ces éléments en famille :

La gestion des lignes (grid-row-xxx), la gestion des colonnes (grid-column-xxx), la gestion du template (grid-template-xxx) et les autres propriétés.

**1.1** A présent vous devez indiquer que ces blocs se trouvent dans une grille, pour cela placez un bloc avec l'**identifiant** container qui contient les 5 autres blocs.

L'en-tête
Menu
Coeur de la page
pub
Le pied de page

Si l'exercice 1 a été fait correctement, une bordure supplémentaire devrait apparaître.

**1.2** Dans le css vous indiquerait que le container est une grille (propriété display), et que les éléments sont séparés de cette grille sont séparés de 20px (gap).

L'en-tête
Menu
Coeur de la page
pub
Le pied de page

Les éléments ont dû s'espacer.

**1.3** Vous avez une grille css il ne manque plus qu'à placer les éléments correctement dans cette grille.

D'abord, donnez un **identifiant** html à chacun de vos blocs.

Ensuite, dans le css vous leur donnerez un nom d'item qui correspondra, de préférence, à son identifiant html (pour ne pas s'y perdre) mais ce n'est pas obligatoire. Pour cela vous aurez besoin de

*vous référez à la documentation et de chercher dans les property values des propriétés laquelle sert à donner un nom d'item dans une grille.*

A ce stade vous avez donc 6 blocs html, chacun ayant un identifiant et dans le css vous avez 7 blocs de propriétés (1 pour chaque bloc de l'exercice 1, 1 pour le container et 1 pour les bordures).

**1.4** Il faut maintenant placer nos blocs, cherchez dans les propriétés de grid-template-xxx celle qui vous permet d'utiliser les noms d'item que vous avez donnés plus tôt et celle qui vous permet de définir la taille des colonnes.

Le template sera composée de 3 lignes sur 3 colonnes.

La première colonne aura une taille fixe de 150pixel (150px)

La deuxième colonne aura une taille relative sur la place restante de 80% (8fr)

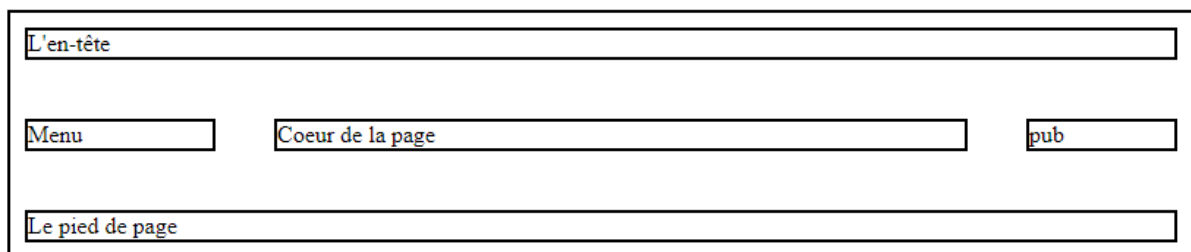
La troisième colonne aura une taille relative sur la place restante de 20% (2fr)

L'en-tête occupera toute la première ligne.

Sur la deuxième ligne :

- 1<sup>ère</sup> colonne : le bloc du menu
- 2<sup>ème</sup> colonne : le bloc du cœur de la page
- 3<sup>ème</sup> colonne : le bloc de pub

Enfin, la troisième ligne sera occupée par le pied de page



Vous devriez aboutir à ce résultat. Le nombre de sélecteur dans le css n'a pas changé (7), ni le nombre d'éléments dans le body (6). Si vous redimensionnez la fenêtre en largeur, la taille du menu ne devrait pas changer, tandis que les autres adapteront leur largeur à celle de la fenêtre.

### Exercice 3 : Un peu de style pour le confort des yeux

En modifiant le sélecteur, faites-en sorte que seuls les éléments contenus dans le container aient une bordure.

Dans ce même sélecteur, ajouter une couleur de fond aux éléments : rouge 105, vert 144, bleu 204.

Arrondissez les coins de bordure de 4 pixels.

Et changez la couleur de la bordure : rouge 65, vert 87, bleu 120.

Ajoutez une marge intérieure de 10 pixels.

Changez la couleur du texte en blanc.

Changez la police d'écriture et choisissez-en une sans empattement.



Vous devriez aboutir à ce résultat. Le nombre de sélecteurs dans le css (7) n'a pas changé, ni le nombre d'éléments dans l'html (6).

## Exercice 4 : L'en-tête

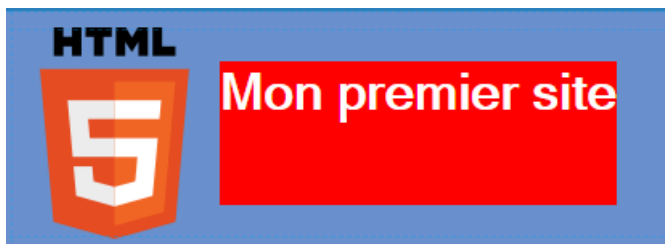
1.1 Vous allez enrichir votre en-tête avec un logo et un titre.

Pour cela placez un élément image et un élément titre 1.

L'image prendra pour source logo.png et le titre aura pour contenu « Mon premier site ».

Faites-en sorte que votre élément d'en tête (le contenant du logo et du titre) affiche son contenu en flex horizontal.

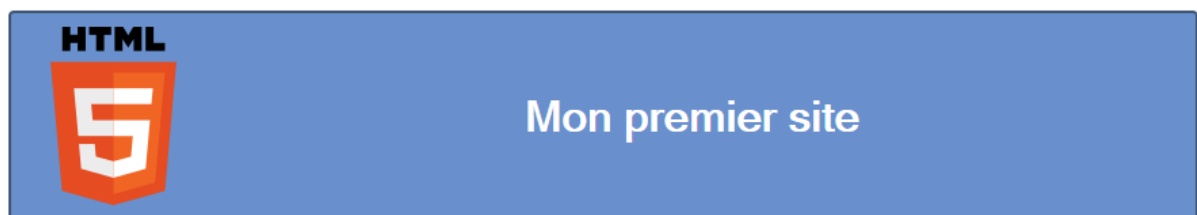
Si vous mettez une couleur de fond à votre titre vous constaterez qu'il ne remplit pas tout son contenant.



Dans le css, grâce à la propriété flex, faites-en sorte que le titre remplisse tout son contenant.

Ensuite, alignez le contenu de votre titre au centre, aussi bien verticalement qu'horizontalement.

Pour cela le titre aura également un affichage « flex » et vous vous servirez des propriétés justify-content et align-items.



1.2 Maintenant, vous allez mettre une police d'écriture personnalisée à votre titre.

Allez d'abord sur <https://www.dafont.com/fr/> et téléchargez la police d'écriture « Play with fire ».

Vous obtenez un zip qui contient un fichier .ttf (c'est la police d'écriture). Mettez ce fichier dans le dossier de votre site.

Dans le css vous allez charger cette police d'écriture et lui donner le nom « maPolice ».

Toujours dans le css, faites-en sorte que votre titre utilise cette police fraîchement importée.



A ce stade, votre dossier de site se compose de 4 éléments : index.html, style.css, Play With Fire.ttf et logo.png.

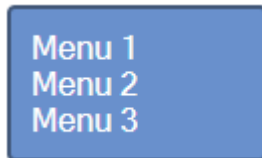
Votre titre reste centré quand on redimensionne la fenêtre et le logo est statique.

## Exercice 5 : Le menu

**1.1** Comme vu en cours, un menu de navigation est une liste de lien.

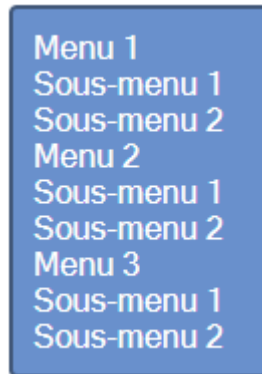
Créez donc un menu de navigation composé de 3 items : Menu 1, Menu 2, Menu 3.

Dans le css, supprimez les puces et la marge engendrée ce nouvel élément. Supprimez aussi la décoration des liens et changez leur couleur en blanc.



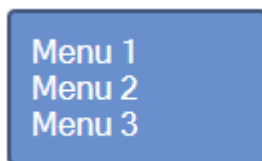
**1.2** A présent, vous allez réaliser un menu déroulant.

Pour cela, chaque item de votre menu va avoir une sous liste de liens qui prendra « sousmenu » comme valeur de l'attribut class.



Dans le css, faites-en sorte que les sousmenu ne s'affiche pas grâce à la propriété display.

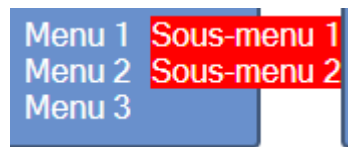
Si les liens de vos sous menu n'ont pas le même style que sur l'image adapté le sélecteur pour qu'il affecte et les liens du menu et les liens des sous-menus.



**1.3** Maintenant, grâce au sélecteur :hover, faites-en sorte qu'au survol d'un li de la première liste, le sousmenu associé change sa propriété d'affichage pour apparaître.

Vous devriez constater que le sous menu s'affiche en dessous de l'item survolé.

Avec la bonne combinaison de valeurs des propriétés display et position des sous menu, faites-en sorte que le sous menu s'affiche à droite de l'item survolé.



Rajoutez une marge extérieure, à gauche, de 10pixels et une couleur de fond pour mieux voir.

J'ai maintenant un sous menu fonctionnel.

## Exercice 6 :

### 1.1 Donnez une taille maximale de 1200 pixels au container.

Avec la propriété `margin`, centrez le container dans le `body`.

Retirez la couleur de fond des sous-menus et ajoutez une couleur de fond grise à l'élément `body` en passant par l'attribut `style` de celui-ci.

### 1.2 TABLEAU

Dans le cœur de la page, rajoutez un tableau de 5 lignes et 3 colonnes. La première ligne sera dédiée aux en-tête de colonnes (utilisez la balise appropriée). Les en-têtes afficheront Colonne 1, Colonne 2 et Colonne 3. Dans les autres cellules, mettez des nombres aléatoires entre 0 et 100.

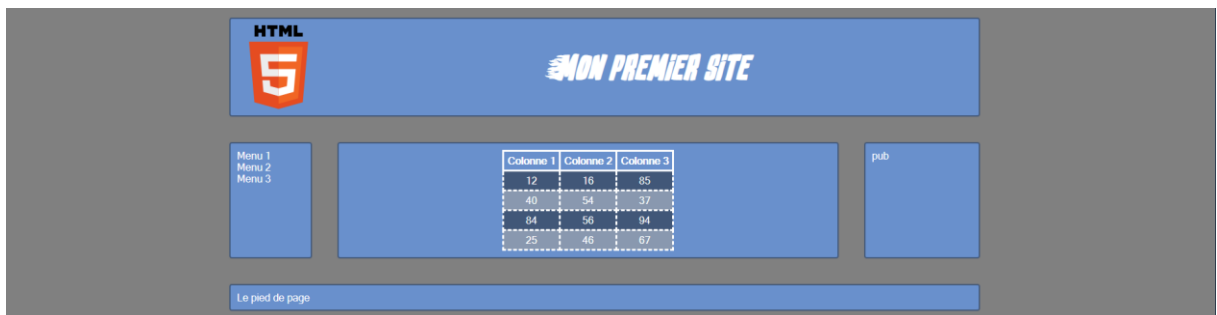
Dans le CSS, mettez une bordure au tableau, en pointillé de 3 pixels d'épaisseur et blanche. Ainsi qu'une marge intérieure de 5px.

Rajoutez les cellules au sélecteur pour leur appliquer le même style. Vous devriez constater un problème au niveau des bordures :

Colonne 1	Colonne 2	Colonne 3
12	16	85
40	54	37
84	56	94
25	46	67

Pour corriger cela servez-vous de la propriété `border-collapse`. Faites également en sorte que les bordures des en-têtes ne soient plus des pointillés mais une ligne pleine. Centrez le tableau dans son contenant, comme vous l'avez fait avec le container. Enfin, centrez le texte des cellules au centre.

Dans le CSS, grâce au sélecteur `tr:nth-child`, colorez le fond des lignes paires avec la couleur #415778. En réutilisant le sélecteur `tr:nth-child`, colorez le fond des lignes impaires avec la couleur #8998AF. Si la ligne des en-têtes se colorent également, corrigez le sélecteur.



Vous devriez aboutir à ce résultat.



## Exercice 7 : Responsive

*Vous allez maintenant faire en sorte que votre site soit responsive. Vous avez vu en cours que la méthode privilégiée est le « mobile first ». Cela veut dire que le style que vous avez défini pour l'instant – adapté à un écran d'ordinateur – ne sera vrai que qu'à condition que l'écran soit plus grand qu'un seuil que vous définirez. Vous modifierez le contenu de vos sélecteurs actuels pour qu'il convienne à un affichage mobile.*

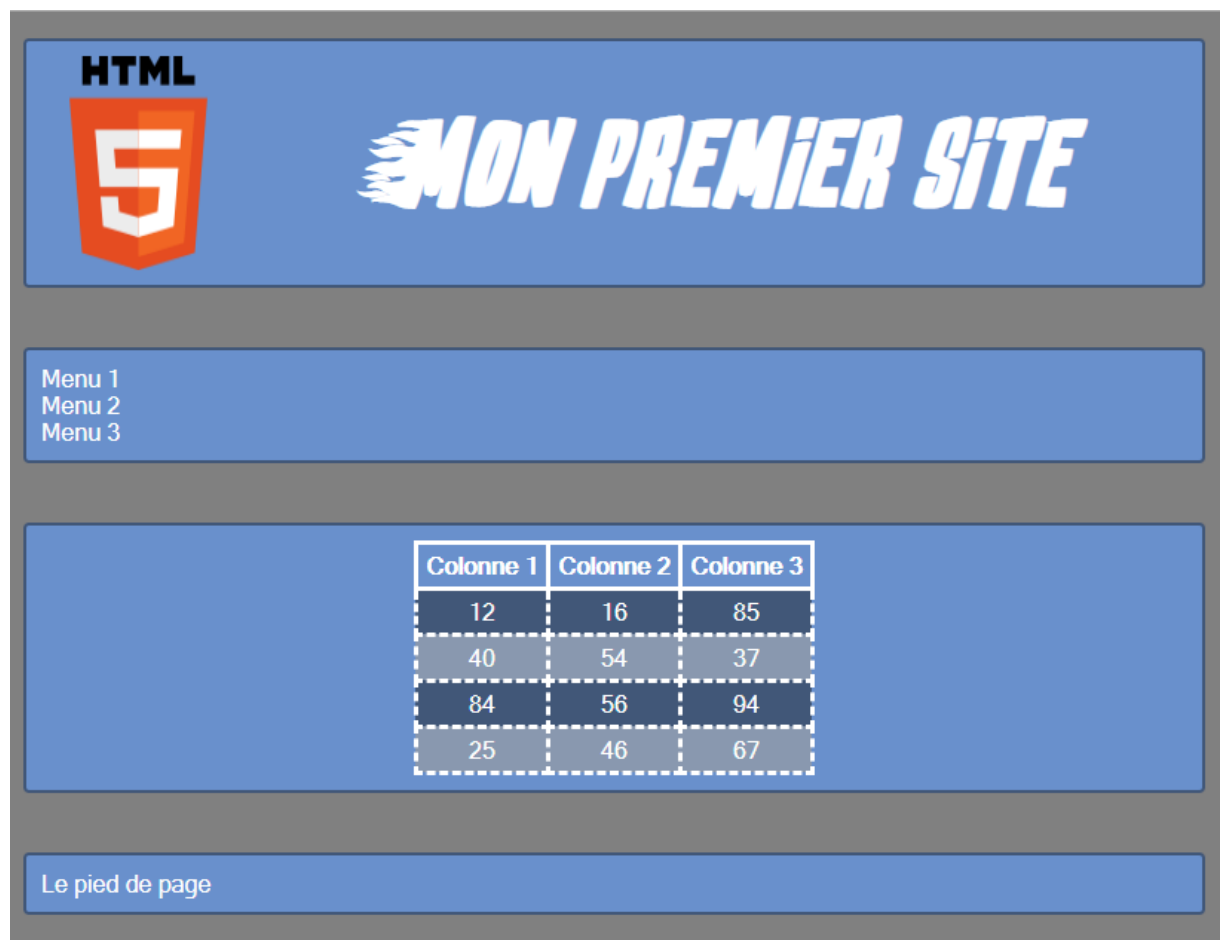
### 1.1 La boîte de pub

Pour vous faire la main, vous allez dans un premier temps masquer la boîte de la pub (propriété `display`) et l'afficher si la largeur de l'écran est d'au moins 1000 pixels (`@media`).

En réduisant la taille de la fenêtre vous devriez constater que l'élément disparaît, puis réapparaît si vous l'augmentez.

### 1.2 Le template de la grille

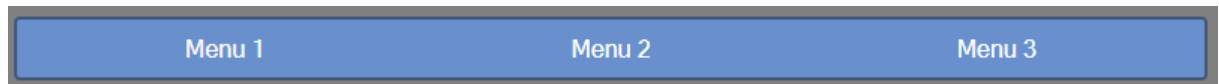
Quand la boîte de pub est masquée, il y a un trou à sa place ; il faut donc aussi modifier le template de votre grille (`grid-template-columns` et `grid-template-areas`) pour qu'il soit responsive. Pour cela l'affichage mobile de votre site se fera par empilement :



Et le template actuel s'appliquera sous condition que la largeur de l'écran soit d'au moins 1000 pixels.

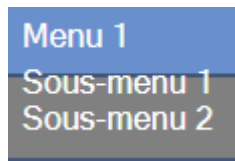
### 1.3 La barre de navigation

Vous pouvez constater que l’affichage de votre menu n’est plus adapté à cette nouvelle vue. Vous allez donc revoir l’affichage de votre liste pour qu’elle soit de type flex et que ces éléments se répartissent de façon homogène (propriété justify-content) :



Ce faisant, si vous revenez à une largeur d’écran supérieure à 1000 pixels vous constaterez que l’affichage n’est plus correct. Dans le @media rajoutez donc une propriété à la liste pour que le flex se fasse en colonne (cherchez dans la documentation la propriété ad-hoc flex-x).

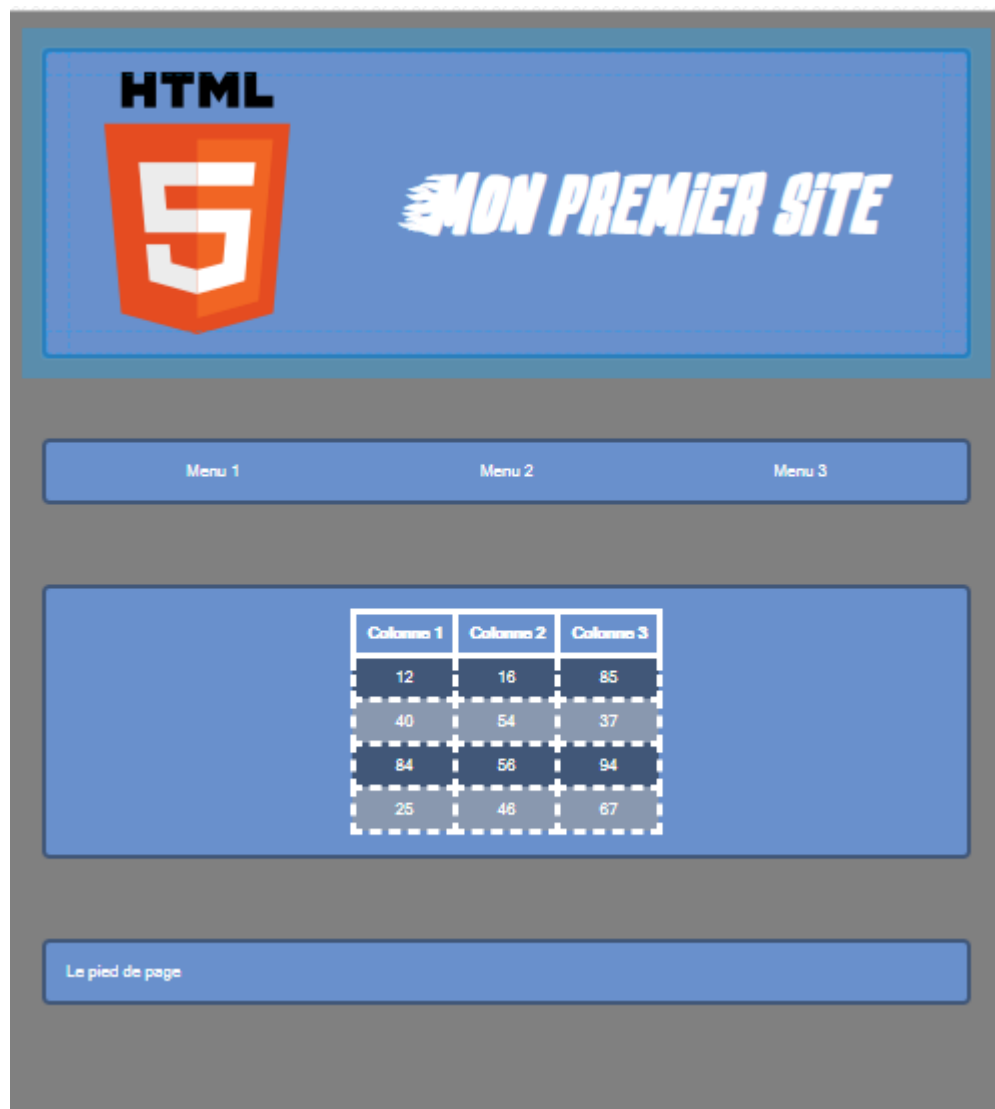
Maintenant, il faut que vous changiez l’affichage de vos sous menu pour l’affichage mobile :



N’oubliez pas de vous assurer que l’affichage du sous menu soit tel que défini précédemment quand l’écran est plus large.

Enfin, vous allez rendre votre taille de police responsive. Pour cela vous allez utiliser l’unité « rem » qui est relative à la taille de la police définit dans le :root ou dans html (comme vu en cours parti 5, slide 8).

Définissez donc une font-size de 50% pour le mobile et de 100% pour les écrans plus larges. Pour le titre vous mettre une font-size de 3rem. Ainsi, la taille du titre va changer en fonction de la taille de l’écran.



Le texte du contenu est trop petit, deux possibilités :

- Augmenter la taille de police du root et réduire celle du titre (70% et 2,5rem par exemple)
- Rendre la taille de police des éléments responsive (par exemple : 2rem en mobile, 1rem pour l'ordinateur)

Identifiez la méthode la plus propre et mettez là en œuvre.

En quelques lignes vous avez rendu votre site responsive.

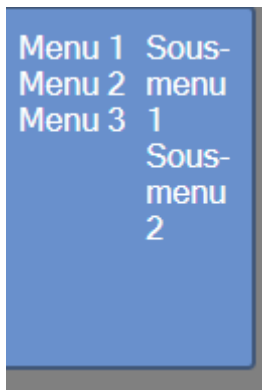
## Exercice 8 : Animation de départ

*Vous allez faire en sorte qu'à l'ouverture de la page, les différents éléments ne soient pas tout de suite visible et qu'ils s'introduisent dans la page grâce à une animation css.*

Vous allez commencer avec l'élément header. Créez une animation (keyframes) et appelez la « fromtop ». Cette animation n'aura que deux étapes : le début et la fin. L'animation modifiera la valeur de la propriété top pour aller de -1000px à 0px. Votre animation est prête à l'emploi. Il faut donc l'appeler dans le sélecteur du header et lui indiquer une durée de 2s. En actualisant la page, vous devriez constater que rien ne se passe ; c'est parce l'animation modifie une propriété (top), qui n'a de sens que si la position de l'élément est définie (essayez avec relative par exemple).

Faites de même pour la pub, le footer et le nav, avec les noms d'animations fromright, frombottom et fromleft et les propriétés right, bottom et left.

Ce faisant, vous devriez constater que le comportement des sous menus a changé.



Pour revenir au comportement de base, vous pouvez ajouter une largeur fixe au sous menu. L'autre astuce consiste à mettre la propriété de position de la nav dans l'animation. De cette manière, une fois l'animation terminée, le nav reprendre son comportement initial.

Enfin, en jouant sur la propriété d'opacité, créez une animation qui fera apparaître en fondu le cœur de la page. Cette animation démarrera une fois les autres terminées.

Maintenant si je vous demande de changer la durée de ces animations à 3s, vous allez devoir faire 6 changements : l'animation duration des 5 blocs et le retard du départ de l'animation du cœur de la page. Cela signifie que votre code n'est pas bien factorisé.

Créez une classe css qui s'appellera « animationDepart » et qui possèdera la propriété définissant la durée de l'animation. Dans l'HTML, ajoutez cette classe aux 5 éléments concernés et supprimez la durée d'animation dans le css là où c'est nécessaire.

Vous avez résolu une partie du problème. Cependant, il est toujours nécessaire de modifier à la fois la durée d'animation dans la classe css et le retard de l'animation du cœur de la page. Créez donc une

variable dans le pseudo élément :root, qui aura pour valeur la durée de l'animation. Associez cette variable à la classe animationDepart et au retard de l'animation pour le cœur de la page.

A présent si vous ne modifiez que la variable, cette modification s'applique à tous les éléments.

Vous savez maintenant créer des animations, mais le plus important c'est la factorisation du code, qui facilite sa lecture et sa maintenance.

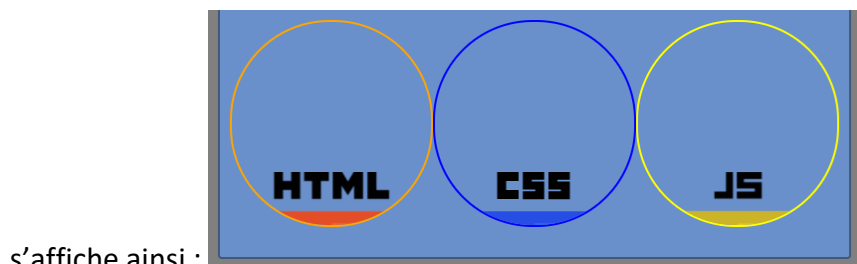
### Exercice 9 : Des images, des transitions et encore une variable.

*Gardons à l'esprit que le but est de faire un code factorisé. Dans cet exercice vous allez créer 3 éléments qui ont exactement le même comportement mais des paramètres différents. Nous ne sommes pas en JavaScript, et il n'est pas possible d'utiliser des fonctions avec différents paramètres, mais vous allez voir qu'on peut tout de même s'en sortir.*

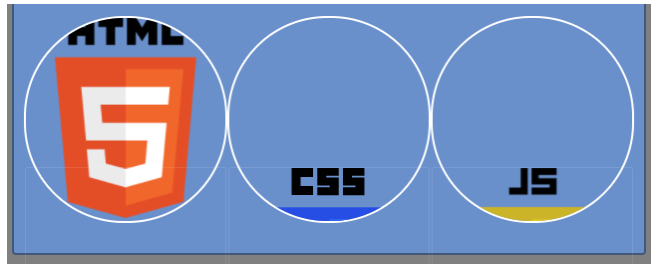
- Première étape placer votre tableau précédent dans une section. Puis créez en dessous une nouvelle section. Dans le css appliquez une marge extérieure du bas de 20 pixels.
- Dans cette nouvelle section placez trois images, respectivement pour html.webp, css.svg et js.png. Ces images seront placées chacune dans une div : `<div><img></div>`
- Créez dans votre css un sélecteur qui va redimensionner vos images à 200 pixels de largeur et 200 pixels de hauteur. Attention à ce que le logo ne soit pas affecté !!
- Ensuite, comme pour le menu horizontal, on veut répartir nos 3 images équitablement sur la ligne. Pour cela, faites usage de l'affichage en flex avec la justification adéquate :



- Pour les div de votre nouvelle section appliquez une bordure solide de 2px d'épaisseur. Et arrondissez les coins de bordure à 100 pixels.
- Vous devriez constater que les images débordent. Gérez ce débordement (overflow en anglais) de manière à ce qu'il soit masqué.
- Grâce à la propriété transform et la translation , faites en sorte que vos images



- Ajoutez un nouveau sélecteur qui, au survol de l'image, va remettre les images à leur



position initiale :

- Pour donner un peu de fluidité ajoutez une propriété de transition à vos images. Elle s'appliquera au transform, durera 0.5s et sera linéaire dans le temps.

*A présent je voudrais mettre une couleur orange sur la bordure de la première image, bleue sur la deuxième et jaune sur la dernière. Pour cela la solution la plus simple consisterait à définir un attribut style dans chacune des div et définir la propriété border-color voulue.*

*Pour l'exercice on va passer par une variable.*

- Ajoutez un attribut style dans chacune de ces div
- Dans ce style définissez une variable --color et donnez-lui la valeur de la couleur voulue (orange, blue, yellow)
- Dans le css, pour les div, rajoutez la propriété border-color et affectez-lui la variable en question.

