

TD2 - CSS : Mise en forme de CinéScore

Fiche de travaux dirigés - Une feuille de style unique (styles.css) pour toutes les pages

Introduction

Dans ce TD, vous allez transformer votre site HTML brut en une interface moderne et agréable. L'objectif est de créer un fichier `styles.css` unique qui sera partagé par toutes les pages de votre site CinéScore.

Vous découvrirez progressivement les principales propriétés CSS : mise en forme du texte, couleurs, espacements, positionnement, et effets visuels.

Exercice 1 - Configuration et styles de base

1.1 Création et liaison du fichier CSS

Créez un fichier `styles.css` à la racine de votre projet. Pour que ce fichier soit utilisé par vos pages HTML, vous devez le lier dans la section `<head>` de chaque page à l'aide de la balise `<link>` :

```
html
<link rel="stylesheet" href="styles.css">
```

1.2 Reset CSS : partir sur une base propre

Chaque navigateur applique des styles par défaut aux éléments HTML (marges, espacements...). Pour garantir un rendu cohérent sur tous les navigateurs, on commence par "réinitialiser" ces styles.

Le sélecteur universel `*` cible tous les éléments de la page.

Propriétés à utiliser :

- `margin` : définit l'espace extérieur autour d'un élément. Valeurs possibles : une taille (px, em, %) ou `auto`.
- `padding` : définit l'espace intérieur entre le contenu et la bordure d'un élément.

À faire : Appliquer à tous les éléments un `margin` et un `padding` de `0`.

1.3 Styles généraux du document

Le `body` : conteneur principal

On définit l'apparence générale de la page via l'élément `body`.

Propriétés à utiliser :

- `font-family` : définit la police de caractères. On indique plusieurs polices séparées par des virgules ; le navigateur utilisera la première disponible.
- `background-color` : définit la couleur de fond d'un élément.
- `color` : définit la couleur du texte.

À faire pour le body :

- Police : `Arial, sans-serif`
- Couleur de fond : `#f5f5f5`
- Couleur du texte : `#333`

Les titres : une identité visuelle cohérente

Pour donner une identité visuelle à notre site, on souhaite que tous les titres utilisent une police avec empattements (serif), plus élégante.

À faire pour h1, h2, h3, h4 :

- Police : `Georgia, serif`
- Couleur : `#1a1a1a`

Tailles et styles des titres

Chaque niveau de titre doit avoir une taille et un style différents.

Propriété à utiliser :

- `font-size` : définit la taille du texte. L'unité `em` est relative à la taille de police du parent.

À faire pour h1 :

- Taille : `2.5em`
- Marge en bas : `0.5em`

Pour h2, on souhaite ajouter un trait rouge en dessous pour les mettre en valeur.


Propriété à utiliser :

- `border-bottom` : crée une bordure sous l'élément. Syntaxe : `épaisseur style couleur` (ex: `3px solid #e63946`).
- `padding-bottom` : espace entre le contenu et la bordure du bas.

À faire pour h2 :

- Taille : 2em
- Marge en bas : 0.75em
- Bordure en bas : trait plein de 3px, couleur  #e63946
- Padding en bas : 0.3em

À faire pour h3 :

- Taille : 1.5em
- Marge en bas : 0.5em
- Couleur :  #457b9d

À faire pour h4 :

- Taille : 1.2em
- Marge en bas : 0.4em

Les paragraphes

À faire : Pour améliorer la lisibilité, ajouter une marge en bas de 1em à tous les paragraphes.

Exercice 2 - Header, navigation et mise en page



2.1 Le header : un bandeau accrocheur

Le header est la première chose que voit l'utilisateur. On veut le rendre visuellement impactant avec un dégradé de couleurs et le garder visible en haut de page lors du défilement.

Propriétés à utiliser :

- **background: linear-gradient()** : crée un dégradé de couleurs.
 - **Syntaxe** : `linear-gradient(angle, couleur1 position1, couleur2 position2)`
 - **L'angle** définit la direction du dégradé :
 - `0deg` = du bas vers le haut
 - `90deg` = de gauche à droite
 - `180deg` = du haut vers le bas
 - `135deg` = en diagonale, du coin supérieur gauche vers le coin inférieur droit
 - **Les positions** (0%, 100%) indiquent où commence et finit chaque couleur.
 - **Exemple** : `linear-gradient(135deg, #1d3557 0%, #457b9d 100%)` crée un dégradé diagonal partant du bleu foncé en haut à gauche vers le bleu clair en bas à droite.
- **box-shadow** : ajoute une ombre portée autour de l'élément.
 - **Syntaxe** : `décalage-X décalage-Y flou couleur`
 - **décalage-X** : déplace l'ombre horizontalement (positif = vers la droite, négatif = vers la gauche)
 - **décalage-Y** : déplace l'ombre verticalement (positif = vers le bas, négatif = vers le haut)
 - **flou** : plus la valeur est grande, plus l'ombre est diffuse et douce
 - **couleur** : souvent en `rgba()` pour ajouter de la transparence
 - **Exemple** : `box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1)` crée une ombre légère en dessous de l'élément (pas de décalage horizontal, 4px vers le bas, flou de 6px, noir à 10% d'opacité).
- **position: sticky** : l'élément reste "collé" à sa position quand on défile.
- **top** : position par rapport au haut (utilisé avec `position`).
- **z-index** : définit l'ordre d'empilement (plus la valeur est élevée, plus l'élément est au-dessus).

À faire pour le header :

- Fond dégradé à 135°, de  `#1d3557` (0%) vers  `#457b9d` (100%)
- Couleur du texte : `white`
- Padding : `20px`
- Ombre : décalage `0 4px`, flou `6px`, couleur `rgba(0, 0, 0, 0.1)`
- Position sticky, collé en haut (`top: 0`)
- z-index : `1`

Le titre du header

On veut un titre centré, en majuscules, avec un effet d'ombre pour le faire ressortir.

Propriétés à utiliser :

- `text-align` : aligne le texte (`center`, `left`, `right`, `justify`).
- `text-transform` : transforme la casse du texte (`uppercase`, `lowercase`, `capitalize`).
- `letter-spacing` : espace entre les lettres.
- `text-shadow` : ombre portée sur le texte. Même syntaxe que `box-shadow`.

À faire pour le h1 dans le header :

- Alignement : centré
- Transformation : majuscules
- Couleur : `white`
- Espacement des lettres : `3px`
- Ombre du texte : `2px 2px 4px rgba(0, 0, 0, 0.3)`
- Marge en bas : `15px`

2.2 La navigation : un menu horizontal

On veut transformer la liste de liens en menu horizontal réparti sur toute la largeur.

La liste de navigation

Propriétés à utiliser :

- `list-style` : style des puces de liste (`none` pour les supprimer).
- `display: flex` : active le modèle Flexbox pour disposer les éléments enfants.
- `justify-content` : répartition horizontale des éléments en Flexbox (`space-around`, `space-between`, `center`...).
- `align-items` : alignement vertical des éléments en Flexbox.
- `flex-wrap` : permet aux éléments de passer à la ligne si nécessaire (`wrap` ou `nowrap`).

À faire pour la liste `ul` de navigation :

- Supprimer les puces
- Affichage flex
- Répartition : `space-around`
- Alignement vertical : `center`
- Autoriser le retour à la ligne

À faire pour les `li` de navigation :

- Marge : `5px 10px`

Les liens de navigation

Propriétés à utiliser :

- **text-decoration** : décoration du texte (`none`, `underline`, `line-through`...).
- **font-weight** : graisse du texte (`normal`, `bold`, ou valeur numérique).
- **border-radius** : arrondit les coins d'un élément.
- **transition** : anime les changements de propriétés.
 - **Syntaxe** : `propriété durée courbe`
 - **propriété** : quelle propriété animer (`all` pour toutes, ou une propriété spécifique comme `color`)
 - **durée** : temps de l'animation (ex: `0.3s` pour 0.3 seconde)
 - **courbe** : rythme de l'animation. `ease` commence lentement, accélère, puis ralentit à la fin (effet naturel et fluide). Autres valeurs : `linear` (vitesse constante), `ease-in` (lent au début), `ease-out` (lent à la fin).
- **display: inline-block** : l'élément se comporte comme un inline mais accepte width/height.

À faire pour les liens `a` de navigation :

- Couleur : `white`
- Pas de soulignement
- Texte en gras
- Padding : `10px 20px`
- Coins arrondis : `5px`
- Transition sur toutes les propriétés en `0.3s ease`
- Display : `inline-block`

Effet au survol des liens

Pseudo-classe à utiliser :

- `:hover` : s'applique quand la souris survole l'élément.

Propriété à utiliser :

- **transform** : applique des transformations géométriques à un élément.
 - `translateY(-2px)` : décale l'élément verticalement (valeur négative = vers le haut)
 - `translateX(5px)` : décale horizontalement (valeur positive = vers la droite)
 - `scale(1.05)` : agrandit l'élément (1.05 = 105% de la taille originale)
 - `rotate(45deg)` : fait pivoter l'élément

À faire au survol des liens de navigation :

- Fond : `rgba(255, 255, 255, 0.2)`
- Décalage vers le haut de `2px`

2.3 Structure de la page : main et aside

On veut que le contenu principal (`main`) et la barre latérale (`aside`) soient disposés côte à côte sur les grands écrans.

Configuration du main

Propriétés à utiliser :

- `max-width` : largeur maximale.
- `width` : largeur (peut être en `%`, `px`, `em`...).
- `margin: 0 auto` : centre horizontalement un élément de largeur définie.

À faire pour main :

- Padding : `30px`
- Largeur max : `1200px`
- Largeur : `70%`
- Centré horizontalement

Configuration de l'aside

Propriété à utiliser :

- `height: fit-content` : la hauteur s'adapte au contenu.

À faire pour aside :

- Largeur : `25%`
- Padding : `30px`
- Fond : `#fff`
- Bordure gauche : `3px solid #e63946`
- Position sticky, `top: 140px`
- Hauteur : `fit-content`

Disposition côte à côte pour les écrans larges

Règle à utiliser :

- `@media` : permet d'appliquer des styles selon des conditions (taille d'écran, orientation...).
- `vertical-align` : alignement vertical pour les éléments inline-block.

À faire dans une media query pour min-width 769px :

- Main et aside : `display: inline-block`, `vertical-align: top`

2.4 Sections de l'aside

À faire pour les sections dans aside :

- Fond : `background-color: #f8f9fa`
- Padding : `padding: 20px`
- Coins arrondis : `border-radius: 8px`
- Marge en bas : `margin-bottom: 20px`
- Ombre légère : `box-shadow: 0 2px 4px rgba(0, 0, 0, 0.05)`

À faire pour les h3 dans aside :

- Bordure en bas : `border-bottom: 2px solid #e63946`
- Padding en bas : `padding-bottom: 10px`
- Marge en bas : `margin-bottom: 15px`

À faire pour les listes (ol, ul) dans aside :

- Marge gauche : `margin-left: 20px`

À faire pour les li dans aside :

- Marge en bas : `margin-bottom: 8px`
 - `line-height` (hauteur de ligne) : `1.6`
-

Exercice 3 - Cards de films et effets visuels

3.1 Grille de films

On veut afficher les films sous forme de grille à 3 colonnes.

Propriétés à utiliser :

- `display: grid` : active le modèle CSS Grid.
- `grid-template-columns` : définit le nombre et la taille des colonnes. `repeat(3, 1fr)` crée 3 colonnes de taille égale (`1fr` = 1 fraction de l'espace).
- `gap` : espacement entre les éléments de la grille.

À faire pour la classe `.films-grid` :

- Display grid
- 3 colonnes de taille égale
- Espacement : `gap: 20px`
- Marge en haut : `margin-top: 30px`

3.2 Les cartes de films

Chaque film est affiché dans une "carte" avec un fond blanc, des coins arrondis et une ombre légère.

À faire pour la classe `.film-card` :

- Fond : `white`
- Coins arrondis : `10px`
- Padding : `20px`
- Ombre : `0 4px 6px rgba(0, 0, 0, 0.1)`
- Transition : `all 0.3s ease`
- Position : `relative`

À faire pour les h3 dans `.film-card` :

- Couleur :  `#e63946`
- Marge en bas : `10px`

À faire pour les paragraphes dans `.film-card` :

- Taille : `0.95em`
- Marge en bas : `8px`



Effet au survol des cartes

À faire au survol de `.film-card` :

- Échelle : `1.05`
- Ombre renforcée : `0 8px 15px rgba(0, 0, 0, 0.2)`

3.3 Sections

À faire pour la classe `.intro` :

- Fond dégradé à 135° : de  `#f1faee` (0%) à  `#a8dadc` (100%)
- Padding : `30px`
- Coins arrondis : `10px`
- Marge en bas : `40px`

À faire pour toutes les sections :

- Marge en bas : `40px`

4.1 Le tableau de catalogue

Propriété à utiliser :

- `border-collapse` : fusionne les bordures des cellules adjacentes (`collapse` ou `separate`).

À faire pour les tableaux :

- Largeur : `100%`
- Border-collapse : `collapse`
- Marge : `20px 0`
- Fond : `white`
- Ombre : `0 2px 4px rgba(0, 0, 0, 0.1)`

À faire pour le thead :

- Fond dégradé (même que le header)
- Couleur : `white`

À faire pour th et td :

- Padding : `15px`
- Alignement : `left`
- Bordure en bas : `1px solid #ddd`

À faire pour th :

- Texte en gras
- Majuscules
- Taille : `0.9em`
- Espacement des lettres : `1px`

Alternance de couleurs sur les lignes

Pseudo-classe à utiliser :

- `:nth-child()` : cible des éléments selon leur position. `even` = pairs, `odd` = impairs.

À faire pour les lignes du tbody :

- Lignes paires : fond `#f8f9fa`
- Lignes impaires : fond `white`
- Au survol : fond `#e3f2fd`, transition `0.2s ease`

À faire pour les liens dans tbody :

- Couleur : ■ `#e63946`
- Pas de soulignement
- Texte en gras
- Transition couleur : `0.2s ease`
- Au survol : couleur ■ `#1d3557`, souligné

4.2 Section des définitions

À faire pour la classe `.definitions` :

- Fond : `white`
- Padding : `30px`
- Coins arrondis : `10px`
- Ombre : `0 2px 4px rgba(0, 0, 0, 0.1)`

À faire pour les listes de définition (`dl`, `dt`, `dd`) :

- `dl` : marge en haut `20px`
- `dt` (terme) : gras, couleur ■ `#1d3557`, taille `1.1em`, marge haut `15px`, marge bas `5px`
- `dd` (définition) : marge gauche `20px`, marge bas `15px`, padding gauche `15px`, bordure gauche `3px solid #a8dadc`, couleur `#555`

4.3 Fiche détaillée d'un film

À faire pour la classe `.film-detail` :

- Fond : `white`
- Padding : `40px`
- Coins arrondis : `10px`
- Ombre : `0 4px 8px rgba(0, 0, 0, 0.1)`

À faire pour la section `.acteurs` :

- Liste ordonnée : marge gauche `40px`, hauteur de ligne `2`
- Items : marge en bas `10px`
- Éléments `sub` (rôles) : couleur `#666`, italique

À faire pour les avis dans `.critiques` :

- Fond :
 - Padding :
 - Coins arrondis :
 - Marge en bas :
 - Bordure gauche :
 - Transition :
 - Au survol : ombre , décalage horizontal de
 - h4 : couleur , marge bas
 - em : couleur , taille
-

Exercice 4b - Formulaire

4b.1 Container et structure

À faire pour :

- Fond :
- Padding :
- Coins arrondis :
- Ombre :

À faire pour :

- Marge en haut :

À faire pour :

- Bordure :
- Coins arrondis :
- Padding :
- Marge en bas :
- Fond :

À faire pour :


- Texte gras
- Couleur :
- Taille :
- Padding :
- Fond :

4b.2 Champs de formulaire

À faire pour `.form-group` :

- Marge en bas : `20px`

À faire pour les labels :

- Display : `block`
- Texte gras
- Marge en bas : `8px`
- Couleur :  `#1d3557`


Pseudo-classe à utiliser :

- `:focus` : s'applique quand l'élément a le focus (sélectionné par l'utilisateur).

Propriété à utiliser :

- `outline` : contour qui apparaît autour des éléments focalisés.
- `resize` : contrôle le redimensionnement d'un textarea (`vertical`, `horizontal`, `both`, `none`).

À faire pour les champs text, textarea et select :

- Largeur : `100%`
- Padding : `10px`
- Bordure : `2px solid #ddd`
- Coins arrondis : `5px`
- Taille : `1em`
- Police : `Arial, sans-serif`
- Transition bordure : `0.3s ease`
- Au focus : pas d'outline, bordure  `#457b9d`, ombre `0 0 5px rgba(69, 123, 157, 0.3)`

À faire pour textarea :

- Redimensionnement vertical uniquement
- Hauteur minimale : `100px`

4b.3 Checkboxes, radios et range

À faire pour les groupes checkbox/radio :

- Marge en bas : `10px`

À faire pour les labels dans ces groupes :

- Display : `inline`
- Pas de gras
- Marge gauche : `8px`


À faire pour les inputs checkbox/radio :

- Marge droite : `5px`
- `cursor: pointer` : change le curseur en main au survol

À faire pour input type range :

- Largeur : `100%`
- Marge : `10px 0`
- Curseur : pointer

À faire pour output :

- Display : `inline-block`
- Fond :  `#457b9d`
- Couleur : `white`
- Padding : `5px 15px`
- Coins arrondis : `20px`
- Texte gras
- Marge gauche : `10px`

4b.4 Boutons

À faire pour `.form-actions` :

- Marge en haut : `30px`
- Alignement texte : centré

À faire pour les boutons submit et reset :

- Couleur texte : `white`
- Pas de bordure
- Padding : `15px 40px`
- Taille : `1.1em`
- Texte gras
- Coins arrondis : `25px`
- Curseur : `pointer`
- Transition : `all 0.3s ease`
- Marge : `0 10px`
- Ombre : `0 4px 6px rgba(0, 0, 0, 0.1)`

Pour submit :

- Fond dégradé : de `#e63946` à `#c53030`
- Au survol : dégradé de `#c53030` à `#a02828`, décalage vers le haut de `2px`, ombre renforcée

Pour reset :

- Fond dégradé : de `#6c757d` à `#495057`
 - Au survol : dégradé de `#495057` à `#343a40`, décalage vers le haut de `2px`, ombre renforcée
-

Exercice 4c - Footer

Propriété à utiliser :

- `margin-top: auto` : pousse l'élément vers le bas de son conteneur.

À faire pour le footer :

- Fond : `#1d3557`
- Couleur : `white`
- Padding : `30px`
- Alignement texte : centré
- Marge en haut : `auto`

À faire pour les paragraphes du footer :

- Marge en bas : `15px`

À faire pour blockquote (citation) :

- Italique
 - Bordure gauche : `4px solid #e63946`
 - Padding gauche : `20px`
 - Marge : `20px auto`
 - Largeur max : `600px`
 - Couleur : `#f1faee`
-

Exercice 5 - Responsive Design

Le responsive design permet à votre site de s'adapter à différentes tailles d'écran. On utilise les **media queries** pour appliquer des styles différents selon la largeur.

5.1 Tablette (768px - 1400px)

À faire dans `@media (max-width: 1400px) and (min-width: 768px)` :

- `.films-grid` : 2 colonnes au lieu de 3
- `main` : largeur `100%`, padding `20px`
- `aside` : largeur `100%`, position `static`, supprimer la bordure gauche, ajouter une bordure en haut `3px solid #e63946`, marge en haut `30px`

5.2 Mobile (moins de 768px)

À faire dans `@media (max-width: 768px)` :

Grille :

- `.films-grid` : 1 seule colonne

Navigation :

- `nav ul` : direction colonne
- `nav ul li` : marge `10px 0`, largeur `100%`, texte centré
- `nav ul li a` : display block, largeur `100%`

Tailles de police réduites :

- `h1` : `1.8em`
- `h2` : `1.5em`
- `h3` : `1.2em`

Padding réduit :

- header : 15px
- main : 15px
- .intro, .film-detail, .formulaire-ajout : 20px
- fieldset : 15px

Tableaux :

- Taille de police : 0.9em
- Padding th/td : 10px

Boutons formulaire :

- Display : block
- Largeur : 100%
- Marge : 10px 0

