CPE Lyon - 3ETI - Année 2023/2024

Techniques et Langages du Web

Session 1 du 10 janvier 2024 - Durée : 2h

Sujet

L'objectif est de réaliser une page de présentation des cartes de la nouvelle extension du jeu ***Magic : l'Assemblée*** intitulée ***Le Seigneur des Anneaux : chroniques de la Terre du Milieu***.

La figure 1 ci-dessous donne un aperçu du résultat à obtenir :

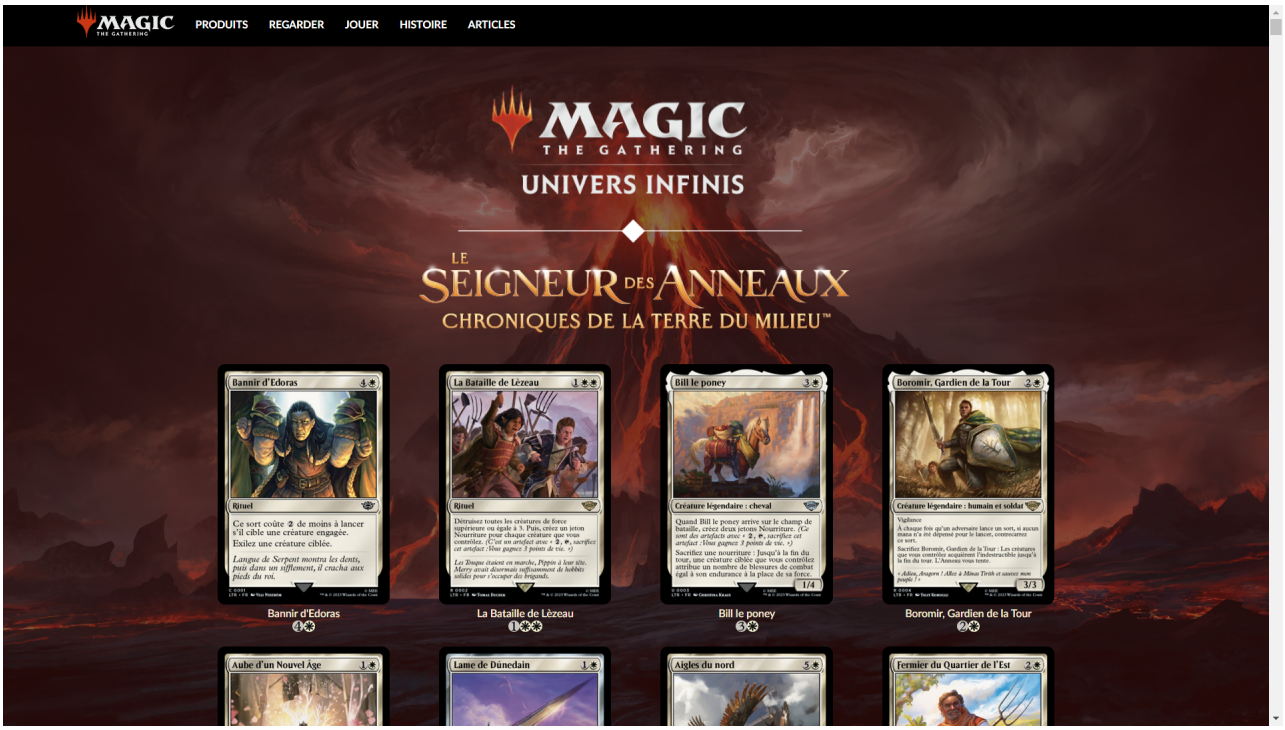

FIGURE 1 – Aperçu de la page web à construire

Cette page comporte :

- un **en-tête**, qui occupe **toute la largeur de la page**, et dont la structure est détaillée en figure 2 ;
- le **contenu principal**, qui présente essentiellement les cartes, et dont la structure est détaillée en figure 3.

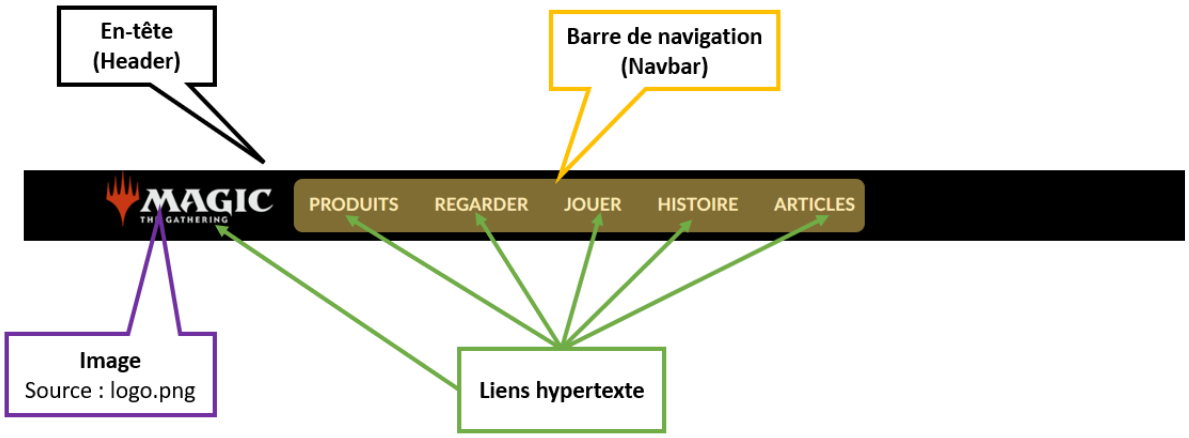

FIGURE 2 – Structure de l'en-tête

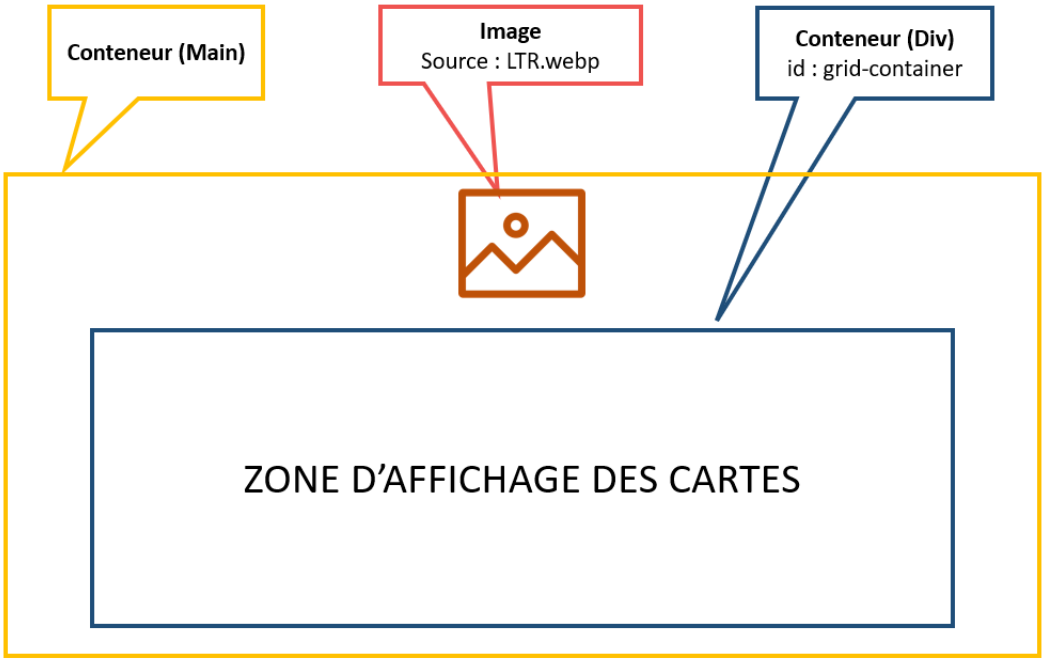

FIGURE 3 – Zone principale de la page

Instructions

Vous allez devoir construire ce site étape par étape, **à partir d'un squelette fourni**, en codant les réponses aux questions qui suivent. Les questions sont donc à traiter dans l'ordre ; néanmoins, **ne restez pas focalisé.e trop longtemps sur une question**, et revenez-y plus tard si nécessaire.

Avant de démarrer, téléchargez le fichier `magic-lotr.zip` disponible sur la page e-campus du sujet. Il contient les fichiers partiellement remplis de votre site, ainsi que certaines illustrations nécessaires. Décompressez ce fichier sur votre Bureau ; vous devriez obtenir un dossier nommé `magic-lotr`, contenant :

- un fichier `index.html` **à compléter**
- un fichier `style.css` **à compléter**
- un fichier `script.js` **à compléter**
- un dossier `images`, qui contient les illustrations.

Ne réorganisez pas le contenu du dossier `magic-lotr` !

⚠ **A la fin de l'examen**, vous devez **déposer sur e-campus un seul fichier**, contenant l'intégralité de votre site (et *uniquement* votre site) au format `.zip`. **Vérifiez qu'il contient bien votre travail !!!**

💡 Pour compresser un dossier au format `.zip`, faites un `clic-droit` sur le dossier puis choisissez `Compresser...`

Questions / Réalisation du site

Avant de démarrer, lancez `Visual Studio Code` puis choisissez `Ouvrir un dossier / Open folder` et ouvrez le dossier `magic-lotr`. Faites ensuite un clic-droit sur le fichier `index.html` et choisissez `Open with Live Server` (si vous ne voyez pas cette option, installez l'extension Live Server) : votre site s'ouvre dans un navigateur ; il est vide pour l'instant, mais chaque modification dans le code sera répercutée automatiquement ici.

Si le navigateur par défaut (normalement, `Firefox`) ne s'ouvre pas, démarrer manuellement `chromium` et saisissez l'adresse `localhost:5500` (`5500` est le port utilisé par défaut par Live Server, indiqué en bas à droite de VS Code).

⚠ Attention ! La correction est **automatique** et une partie de la feuille de style vous est fournie. Aussi, veillez à respecter scrupuleusement (aux minuscules / majuscules près) l'orthographe des noms de sections, de classes, d'identifiants, etc. qui sont donnés.

Partie 1 : HTML & CSS (10 points)

Question 1. Complétez le fichier `index.html` avec les éléments suivants :

- le titre (de votre choix) de la page, qui apparaît dans l'onglet du navigateur
- le fait que le site est en français
- l'inclusion du fichier `style.css`
- l'inclusion du fichier `script.js`

Question 2. Créez **la structure HTML** de l'en-tête, telle que définie en figure 2 (remarque : pour cet exercice, les liens ne doivent pointer vers aucune page particulière). **Limitez-vous aux éléments HTML indiqués sur le schéma.**

Question 3. Créez **la structure HTML** de la zone principale, telle que définie en figure 3.

Question 4. Ajoutez les règles permettant de définir les styles par défaut du site :

- les éléments de la page n'ont ni marge ni padding ;
- la police de caractères utilisée est `Lato, sans-serif` ;
- la taille du texte est `14px`.

Question 5. Mettez en forme l'en-tête (figure 2), en respectant attentivement les contraintes suivantes :

- la couleur de fond est le `noir` ;
- la hauteur de l'en-tête est de `55 pixels` ;
- à l'aide d'une disposition `flex`, positionnez le logo et la barre de navigation côte à côte ;
- il y a un padding de `5%` à gauche ;
- le logo mesure `130px` de largeur ;
- les liens hypertexte sont écrits en `blanc` et en `gras`, doivent être mis **automatiquement en majuscules**, mais n'ont **aucune autre décoration** ; ils ont un padding de `1em` (l'*em* est une unité couramment utilisée en CSS) **à droite et à gauche** ;
- quand on survole les liens à la souris, ils sont mis en évidence par une couleur proche du rouge : `#ea3601` ;
- enfin, en appliquant une disposition `flex` **sur la barre de navigation**, utilisez une règle permettant de centrer verticalement les liens.

Question 6. Mettez à présent en forme la zone principale, en respectant attentivement les contraintes suivantes :

- le conteneur principal (`main`) doit occuper `75%` de la largeur de la page, et être centré horizontalement ; il a aussi une marge supérieure de `15px` ;
- à l'intérieur, tout le contenu doit lui-même être **centré horizontalement** ;
- le conteneur de type `div` doit être découpé en une **grille** de **4 colonnes** dont les lignes sont espacées de `15 pixels` les unes des autres (ce qui ne sera visible que quand la grille sera alimentée avec du contenu) ;
- les images de la classe `card-img` (autrement dit les cartes qui seront affichées) mesurent `320px` de hauteur ; pour renforcer l'aspect de cartes à jouer, arrondissez la bordure des images avec un rayon de **10px** ;
- enfin, modifiez le style du paragraphe situé dans le `div` du template : sa couleur doit être `antiquewhite` et sa `graisse` égale à 500.

Deux autres questions (**12** et **13**) figurent après la partie 2, car leur effet ne sera visible que lorsque la page sera alimentée par des données. Vous pouvez cependant les traiter à n'importe quel moment pour assurer quelques points supplémentaires.

Partie 2 : JavaScript (10 points)

Vous allez à présent alimenter votre page à l'aide des données disponibles *via* l'API proposée par le moteur de recherche `Scryfall` (`https://scryfall.com/docs/api`), qui fournit un nombre colossal d'informations sur toutes les cartes Magic éditées jusqu'ici.

Attention ! Vous veillerez à ne pas effectuer un trop grand nombre d'appels simultanés à l'API dans votre code, sous peine de voir votre adresse IP bannie par Scryfall et de ne pas pouvoir continuer l'examen !

Parmi tous les *types* proposés par l'API Scryfall, celui qui nous intéresse ici est **`cards`** qui, comme son nom l'indique, permet de récupérer des informations sur les cartes (titre, couleur de la bordure, illustration, nom de l'illustrateur, textes, symboles de mana, etc.). Pour cela, il faut utiliser l'*endpoint* `https://api.scryfall.com/cards/search` dont la documentation est disponible à `https://scryfall.com/docs/api/cards/search`. En particulier, vous devrez utiliser les paramètres suivants :

- `q=e:ltr lang:fr` : la requête principale, signifiant ici qu'on souhaite l'*édition* `ltr` (*Lord of the Rings*), en français (attention, il ne faut pas d'espace autour des "deux-points" !) ;
- `format=json` : le format de la réponse doit être JSON ;
- `order=set` : dans la réponse, les cartes sont triées selon leur numéro ;
- `unique=prints` : pour faire simple, renvoie toutes les cartes.

Un exemple de réponse est donné en annexe 1. Vous noterez la présence sur les troisième et quatrième lignes des attributs `has_more` (égal ici à `true`) et `next_page`. En effet, la particularité de cette API est qu'elle utilise la *pagination*. Certaines requêtes renvoient tellement de cartes que la réponse prendrait trop de temps à arriver ; aussi, au lieu d'envoyer toute la réponse d'un seul coup, celle-ci est segmentée en *pages*, correspondant pour Scryfall à **175 cartes**. Pour obtenir la suite de la réponse, il suffit de refaire une requête, à l'URL mentionnée dans l'attribut `next_page` (dont vous pouvez voir qu'elle contient le paramètre supplémentaire `page=2`), et ainsi de suite, jusqu'à obtenir une page dont l'attribut `has_more` vaut `false`. L'extension **Seigneur des Anneaux** comprend **448 cartes**, vous devrez donc utiliser ce mécanisme de pagination pour récupérer toutes les cartes.

Les données elles-mêmes sont contenues dans le tableau `data` de la réponse. Il s'agit d'un tableau d'objets de type `card`. Parmi toutes les propriétés des `card`, celles que vous devrez utiliser sont :

- `printed_name` : le titre de la carte ;
- `image_uris.normal` : l'URL de l'illustration en résolution "normale" ;
- `mana_cost` : les coûts de mana pour jouer cette carte.

Question 7. Dans un premier temps, créez une fonction `afficherCartes()` qui, **au chargement de la page**, récupère l'intégralité des données des 448 cartes au format JSON à l'aide de la fonction

JavaScript `fetch` et de l'API ci-dessus. Pour vérifier que tout fonctionne, vous pourrez par exemple ajouter le nom de chaque carte dans un tableau et vous afficherez la taille finale de ce tableau dans la console.

Attention ! La fonction `fetch` est une fonction *asynchrone* ; n'oubliez pas d'ajouter le mot-clé `async` devant la *définition* de chaque fonction dans laquelle apparaît l'instruction `fetch`, et le mot-clé `await` devant chaque *appel* de fonction asynchrone, comme indiqué dans le cours.

Question 8. Complétez la fonction `afficherCartes()` de sorte que, **pour chacune des cartes disponibles**, une copie du template HTML fourni soit ajoutée dans le `grid-container`, avec l'illustration et le titre de la carte.

La dernière étape consiste à insérer les symboles de coûts de mana sous le titre de la carte. L'endpoint `https://api.scryfall.com/symbology` (documentation : `https://scryfall.com/docs/api/card-symbols/all`) permet d'accéder à toutes les informations concernant les symboles, notamment leurs images. La seule différence est que le tableau `data` de la réponse est à présent un tableau d'objets `card_symbol` (et non plus `card` comme précédemment). Un exemple est donné en annexe 2. Parmi les propriétés d'un objet `card_symbol`, les seules qui nous intéressent sont :

- `symbol` : un symbole, par exemple `{W}`
- `svg_uri` : l'URL de l'image associée.

Question 9. Ecrivez une fonction `getSymbols()` qui interroge cette API, puis renvoie un dictionnaire, associant à chaque symbole l'URL de son image (**rappel** : en JavaScript, les dictionnaires sont tout simplement... les objets !).

Comme indiqué plus haut, les coûts en mana pour chaque objet `card` dans sa propriété `mana_cost`. Ils sont donnés sous la forme d'une chaîne de caractères constituée de symboles (généralement, un seul caractère entre accolades). Par exemple, `{2}{W}{W}` signifie "deux manas génériques, plus deux manas blancs (plaines)". Le fichier JavaScript fourni contient une fonction nommée `parseMana()` qui, à partir d'une chaîne de caractères représentant le coût total en manas, découpe cette chaîne et retourne un **tableau** des différents symboles. Par exemple, `parseMana("{2}{W}{W}"` renvoie le tableau `["{2}","{W}","{W}"]`.

Question 10. Modifiez la fonction `afficherCartes()` : commencez par récupérer le dictionnaire renvoyé par votre fonction `getSymbols`, puis à l'aide de la fonction `parseSymbols` fournie, ajoutez l'illustration de chaque symbole de coût de chaque carte. Pour cela, vous créerez (à l'aide de JavaScript) pour chaque symbole un nouvel élément `img` dont vous modifierez les attributs `src` et `class` : chaque symbole de mana doit appartenir à la classe CSS `mana`. Enfin, ajoutez dans la feuille de style une règle indiquant que la largeur des images de la classe `mana` est de `15px`.

Question 11. Retour sur le style : faites en sorte que l'en-tête reste visible quand on fait défiler la page.

Question 12. Enfin, appliquez un effet de *parallaxe* à l'arrière-plan (i.e. un arrière-plan fixe et correctement dimensionné).

ANNEXES

Annexe 1 – Exemple de JSON retourné par l'API `cards/search`


```
{
    "object": "list",
    "total_cards": 448,
    "has_more": true,
    "next_page": "https://api.scryfall.com/cards/search?format=json&
       order=set&q=e%3Altr+lang%3Afr&unique=prints&page=2",
    "data": [
        {
            "object": "card",
            ...
            "name": "Banish from Edoras",
            "printed_name": "Bannir d'Edoras",
            ...
            "image_uris": {
                "small": "https://cards.scryfall.io/small/front/c/a/ca0
                   c5915-ba7b-47e4-8305-6a28f229451d.jpg?1688347858",
                "normal": "https://cards.scryfall.io/normal/front/c/a/
                   ca0c5915-ba7b-47e4-8305-6a28f229451d.jpg?1688347858"
                   ,
                "large": "https://cards.scryfall.io/large/front/c/a/ca0
                   c5915-ba7b-47e4-8305-6a28f229451d.jpg?1688347858",
            ...
            },
            "mana_cost": "{4}{W}",
            ...
        },
        {
            "object": "card",
            ...
            "name": "The Battle of Bywater",
            "printed_name": "La Bataille de Lèzeau",
            ...
            "image_uris": {
                "small": "https://cards.scryfall.io/small/front/8/4/84f
                   97fe8-9476-4a1a-b0a4-2506ce55f812.jpg?1688348030",
                "normal": "https://cards.scryfall.io/normal/front/8/4/8
                   4f97fe8-9476-4a1a-b0a4-2506ce55f812.jpg?1688348030",
                "large": "https://cards.scryfall.io/large/front/8/4/84f
                   97fe8-9476-4a1a-b0a4-2506ce55f812.jpg?1688348030",
            ...
            },
            "mana_cost": "{1}{W}{W}",
            ...
        },
        ...
```

Annexe 2 – Exemple de JSON retourné par l'API `symbology`


```
{
  "object": "list",
  "has_more": false,
  "data": [
    {
      "object": "card_symbol",
      "symbol": "{T}",
      "svg_uri": "https://svgs.scryfall.io/card-symbols/T.svg",
      "loose_variant": null,
      "english": "tap this permanent",
      "transposable": false,
      "represents_mana": false,
      "appears_in_mana_costs": false,
      "mana_value": 0,
      "cmc": 0,
      "funny": false,
      "colors": [],
      "gatherer_alternates": [
        "ocT",
        "oT"
      ]
    },
    {
      "object": "card_symbol",
      "symbol": "{Q}",
      "svg_uri": "https://svgs.scryfall.io/card-symbols/Q.svg",
      "loose_variant": null,
      "english": "untap this permanent",
      "transposable": false,
      "represents_mana": false,
      "appears_in_mana_costs": false,
      "mana_value": 0,
      "cmc": 0,
      "funny": false,
      "colors": [],
      "gatherer_alternates": null
    },
    ...
}
```