EXERCICE FIL ROUGE

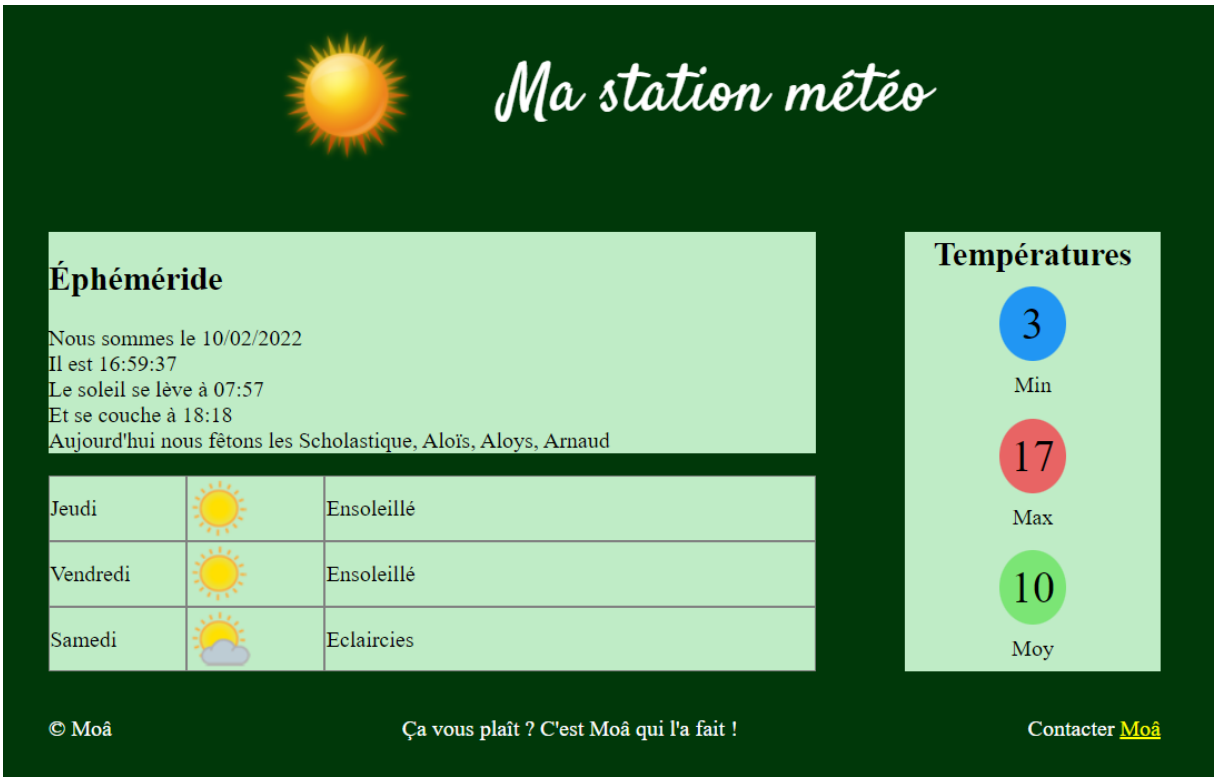
UNE STATION MÉTÉO !

Table des matières

1 Exercice fil rouge

1.1 Objectif

Notre exercice fil rouge va nous permettre d'arriver à créer une page telle que celle-ci :



Bien sûr, il va falloir un peu de temps pour avoir la page présentée telle quelle, et encore plus pour la remplir avec des données réelles.

1.2 Mode d'emploi

La vue finale demandera évidemment un peu d'acquisition de connaissance et de savoir afin de pouvoir faire la page présentée. Le savoir, je vous le fournis. Le savoir-faire, c'est votre travail, le temps que vous passerez à coder, qui vous permettra de l'acquérir.

Pour obtenir la page complète et dynamique, nous procèderons par étapes. D'abord une partie de cours, avec un exercice simple (ou un exemple à copier durant le cours) de mise en application pour chaque notion, puis vous appliquerez votre savoir nouvellement acquis à l'exercice fil rouge… Pour initier votre savoir-faire.

Les énoncés des notions à appliquer, et donc du résultat attendu, sont dans la seconde partie de ce document. Vous trouverez également en annexe un ensemble d'informations complémentaires, comme les codes des couleurs utilisées, sans que ça ne soit une obligation. C'est votre station météo, vous y mettez votre sensibilité et votre patte d'artiste.

Chaque exercice sera une évolution du précédent. Nous utiliserons Git pour faire du versionning. Vous déposerez donc systématiquement le même répertoire, mais pas au même stade d'évolution. Ça vous semble fumeux ? Nous verrons Git ensemble si vous le ne le connaissez pas.

À chaque jour son exercice, son dépôt sur le Moodle de LDNR, et votre production. Vous déposerez vos travaux dans un fichier `zip` (par `tar.gz` ni `7zip`, merci).

1.3 Les attendus

Je vais lourdement insister sur la qualité. En effet, il est inadmissible à l'heure actuelle de proposer du code HTML/CSS impropre à la consommation. J'attends de vous que vous passiez systématiquement votre code aux validateurs HTML et CSS (liens en annexe). Prenez donc un temps pour cette vérification, car il n'est pas impossible (il est en fait très probable) que vous allez faire des bêtises, et parfois pas faciles à trouver. Il vous faudra donc échanger avec moi pour trouver l'erreur, et ça peut prendre du temps.

Les premiers exercices sont faciles et ne prennent pas trop de temps. En revanche, la recherche de bug, ça peut prendre beaucoup d'énergie et de ressources. Donc attardez-vous plus volontiers sur la qualité plutôt que sur la beauté. Une page web (comme n'importe quel programme informatique d'ailleurs), c'est comme une tartine de confiture : la structure, c'est le pain. Le visuel, c'est la confiture. Si vous n'avez pas le pain, vous aurez de la confiture plein les doigts…

2 Exercices HTML

2.1 Exercice HTML 1

- Créez votre répertoire qui hébergera votre site météo.
- Créez-y votre dépôt Git.
- Créez votre fichier page d'accueil `index.html`.
- Créez la structure de votre page (commencez à utiliser `Emmet`)
- Faites un `commit`
- Créez une branche de travail. Appelez-la par exemple `exo1`.
- Placez-vous sur cette branche.
- Analysez la page finale, et trouvez les blocs constitutifs, autrement dit la **structure** de votre page. N'oubliez pas qu'un bloc peut contenir des blocs…
- Reproduisez cette structure dans votre fichier `index.html`. Pensez à faire des commits réguliers quand vous avez trouvez un « truc qui va bien ».

Le visuel ne correspond pas à l'image ? Normal, nous n'avons pas encore fait de mise en forme. La confiture, c'est à la fin de la tartine. Si vous avez correctement défini la structure de votre document, la mise en forme sera une partie de rigolade.

- **Passez votre code au validateur HTML.** Que le résultat soit correct ou pas n'est pas important. C'est la démarche qui l'est.
- Corrigez les éventuelles erreurs. **Demandez-moi au besoin**, je suis là pour vous aider.
- Passez sur la branche principale et faites un `merge` de la branche `exo1`.
- Zippez (zip, pas tar.gz ni 7zip) le répertoire et déposez l'archive dans le dépôt Moodle.

2.2 Exercice HTML 2

Nous allons repartir de l'exercice précédent et le faire évoluer, sans perdre l'initial. C'est tout l'avantage du versionning !

- Vérifiez que vous êtes bien sur la branche principale, sinon placez-vous y.
- Créez une nouvelle branche (par exemple `exo2`) et placez-vous dessus.
- Modifiez le code de votre page pour y inclure du web sémantique. Quels sont les blocs qui doivent devenir des `sections` ou des `articles` ? Y a-t-il un `header` ou un `footer` ? Pensez à `commit` régulièrement.
- Si vous ne l'aviez pas encore fait, faites apparaître dans votre code des `figure` afin de mettre une légende sur certains éléments (les températures, pour ne pas les nommer).
- **Passez votre code au validateur HTML.** (Je ne le dirai plus, mais vous devrez le faire spontanément !) et corrigez vos erreurs.
- Passez sur la branche principale et faites un `merge` de la branche `exo2`.
- Zippez (zip, pas tar.gz ni 7zip) le répertoire et déposez l'archive dans le dépôt Moodle.

2.3 Exercice HTML 3

Nouvel exercice, nouvelle branche tirée **depuis la branche principale**. Quand vous manipulez vos branches, soyez toujours sûrs de l'endroit où est placé votre pointeur `HEAD`. Pensez également à vous positionner sur la branche que vous venez de créer…

Dans cet exercice, vous allez commencer à mettre du style dans votre page. La base est d'arriver à poser des couleurs où il le faut, à dimensionner vos caractères comme vous le souhaitez, à utiliser une fonte personnalisée… Bref, avoir un contenu qui commence à avoir de la gueule !

- Étant sur votre branche `exo3`, créez un fichier `style.css`
- Appliquez les couleurs (dont vous trouverez les codes en annexe, sinon choisissez les vôtres !) sur les différents éléments de votre page. Pensez à commit chaque fois que vous obtenez un résultat définitif (c'est exagéré, mais prenez l'habitude de faire des commit, vous choisirez plus tard quand c'est le bon moment).
- Appliquez les dimensions des caractères sur les différents éléments de votre code. Pensez là encore à commit souvent.
- Ajoutez la fonte personnalisée pour le titre.
- (Validateur**s**…)
- S'il vous reste du temps, commencez à habiller un peu votre page comme vous le souhaitez (autres couleurs, tailles, fonte personnalisée, contenus complémentaires…)
- Faites un merge (je rappelle que vous devez passer sur la branche prinsipale pour faire un merge…)
- zip et dépôt

2.4 Exercice HTML 4

Maintenant que vous avez `Flexbox`, il est temps de faire la mise en page définitive. Et c'est là que vous allez voir si vous avez bien structuré votre page… Car si la structure est mauvaise, vous n'arriverez à rien avec `Flexbox` ou `grid`.

- Changez de branche pour `exo4`
- Mettez votre géométrie en place.
- Commits, merge, zip, dépôt

2.5 Exercice HTML 5

Pour finir vous allez rendre votre page compatibles avec les téléphones portables (responsive web). Choisissez le visuel que vous souhaitez obtenir sur des écrans de 425 pixels ou moins. Si vous le souhaitez, vous pouvez aussi ajouter une taille intermédiaire de 768 pixels pour les tablettes, et 1024 pour les tablettes en mode paysage ou les petits écrans d'ordinateurs.

Là plus que jamais, il est important de s'acharner sur la qualité.

3 Exercices JS

Utilisez AJAX pour interroger des API qui vous fourniront les informations nécessaires à injecter dans votre DOM.

Regardez dans les annexes pour voir les API à utiliser. Je vous conseille d'utiliser l'API suisse pour la météo, elle vous retourne des informations plus exploitables immédiatement. Mais vous pouvez tout aussi bien utiliser l'API de votre choix si vous trouvez mieux. Cela dit, le travail de recherche a déjà été fait, il est peut-être plus utile de vous concentrer sur le code…

Notez que pour utiliser certaines API il peut être nécessaire d'ajouter une extension à votre navigateur pour outrepasser les contrôles CORS. CORS (Cross Origin Resource Sharing) empêche les requêtes AJAX (ou fetch) d'aboutir si elles ne sont pas envoyées sur le même serveur que la page courante, sauf si le serveur a été configuré pour accepter des requêtes de tout le monde. Il faut donc leurrer le navigateur en réécrivant partiellement l'entête retournée par le serveur en lui faisant croire que tout va bien au niveau CORS. Ceci est fait par les extensions suivantes (par exemple, il en existe d'autres) :

- CORS Everywhere pour Firefox
- Moesif Origin & CORS Changer pour Chrome

> Évitez Allow CORS qui ne fonctionne pas trop bien sur Windows et pas du tout sur Linux.

Si vous voulez en savoir plus sur CORS, lisez https://developer.mozilla.org/fr/docs/Web/HTTP/CORS.

4 Annexe

4.1 Outils

Le validateur HTML du W3C : https://validator.w3.org/#validate_by_input

Le validateur CSS du W3C : https://jigsaw.w3.org/css-validator/#validate_by_input

Un vérificateur WCAG en ligne : https://achecker.achecks.ca/checker/index.php

4.2 Couleurs

Voici les codes hexadécimaux des couleurs utilisées :

```
Vert foncé : #003809
Vert clair : #bfecc6
Bleu : #2196f3
Rouge : #e86464
Vert pomme : #7be575
Jaune : #FFFF00
```

4.3 Fontes

La fonte personnalisée est la fonte `Satisfy` qu'on peut trouver par exemple sur 1001 fontes : https://www.1001fonts.com/satisfy-font.html

4.4 API météo/éphéméride

4.4.1 API française

Nous allons utiliser l'API de `météo-concept`. Il faut commencer par créer une clé sur https://api.meteo-concept.com/login. Il faut vous inscrire, vous recevrez un mail de confirmation de votre adresse courriel qui vous permettra de valider votre inscription. Vous pourrez ensuite choisir l'option gratuite qui vous donne la possibilité de faire jusqu'à 500 requêtes sur l'API par jour. Une fois votre compte créé, il faudra créer un `token` (appelez-le `formation`, ou `perso`, ou `toto` peu importe) et donnez-lui le maximum d'accès possible (500 donc). C'est ce token qui vous permettra d'utiliser l'API (vous le fournirez en paramètre de la chaîne de requête).

Ensuite… Lisez la documentation de l'API pur savoir ce que vous pouvez faire. Mais comme ça prend du temps, je vous présente le nécessaire pour notre fil rouge :

Demander les heures de lever/coucher du soleil pour le jour même et pour Toulouse (code INSEE : 31555) :

```
https://api.meteo-concept.com/api/ephemeride/0?token=<Entrez votre token
perso>&insee=31555
```

Pour une documentation plus détaillée : https://api.meteo-concept.com/documentation#ephemeride

Demander la météo pour Toulouse et pour le jour même :

```
https://api.meteo-concept.com/api/forecast/daily/0?token=<Entrez votre
token perso>&insee=31555
```

Pour une documentation plus détaillée : https://api.meteo-concept.com/documentation#forecast-city. L'inconvénient est que la description de la météo est une description numérique standardisée, et que les descriptions textuelles ne sont pas intégrées dans le JSON.

Pour la table des codes des temps : https://api.meteo-concept.com/documentation#code-temps.

Cette API permet également d'avoir le saint du jour, mais malheureusement uniquement dans la version payante.

4.4.2 API suisse

Une alternative plus rapide à utiliser puisque ne demandant pas d'inscription est l'API de https://prevision-meteo.ch/.

Pour obtenir tout le nécessaire, il suffit d'interroger https://www.prevision-meteo.ch/services/json/toulouse. L'avantage avec cette API c'est qu'elle vous fournit le texte descriptif de la météo, et même les icônes correspondantes au temps prévisionnel. En revanche, il y a beaucoup d'information et le JSON retourné contient beaucoup de choses qui ne nous intéressent pas, puisqu'on a le temps heure par heure pendant 7 jours. Il faudra donc bien analyser le contenu pour en extraire ce qui nous intéresse. Mais tout y est. Il est aussi possible que cette API ne fournisse pas le temps pour toutes les localités françaises, mais Toulouse y est…

La documentation se trouve sur https://www.prevision-meteo.ch/uploads/pdf/recuperation-donnees-meteo.pdf.

4.4.3 API saint du jour

> Pour pouvoir utiliser cette API, il faut installer un module sur votre navigateur (voir l'énoncé de l'exercice)

Je vous propose d'utiliser les résultats retournés par cette API : https://nominis.cef.fr/json/nominis.php qui vous retourne un gros JSON contenant tous les saints du jour et leur descriptions ainsi que leur vie… Là encore, il faudra faire un tri !