

## TD 1: manipulation simple du DOM

### Introduction

L'objectif de ce TD est de vous faire travailler les manipulations simples du DOM HTML et la gestion du temps et des dates.

Dans ce TD, Vous allez réaliser une montre analogique permettant d'afficher les heures dans différents fuseaux horaires. Vous allez aborder des notions telles que la manipulation des dates, la mise à jour dynamique du contenu de la page web et les fonctions de gestion du temps.

### Travail à faire

Créez un dossier nommé « td-clocks » à l'emplacement habituel de vos TD, puis le dossier assets et ses sous dossiers « css » pour le style et « js » pour les scripts JS.

1. A la racine de votre projet, créer le fichier « index.html » de votre page web et ajouter quatre blocs contenant un paragraphe pour afficher l'heure de Paris, Londres, New York et Tokyo.
2. Créer une mise en page pour l'affichage des différentes horloges. Utiliser « **Inconsolata** » comme police pour l'horloge.
3. Créer une fonction permettant de récupérer l'heure courante.
4. Au chargement du script, afficher l'heure dans chaque bloc au format HH :MM :SS.
5. Faire en sorte que l'heure se mette à jour automatiquement toutes les secondes avec `setTimeout()` ou `setInterval()`.

### Ressources

[https://www.w3schools.com/jsref/jsref\\_obj\\_date.asp](https://www.w3schools.com/jsref/jsref_obj_date.asp) Référence de l'objet JS Date

<https://www.pierre-giraud.com/javascript-apprendre-coder-cours/delai-settimeout-setinterval/>

[https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Asynchronous/Timeouts\\_and\\_intervals](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Asynchronous/Timeouts_and_intervals)

Gestion du temps avec `setInterval` et `setTimeout`

## TD 2 : manipulation simple du DOM

### Contexte

Ce travail pratique vise à approfondir vos compétences en manipulation du DOM . Vous concevrez une interface affichant des citations qui se renouvellent automatiquement toutes les 30 minutes, tout en variant la couleur de fond de la carte à chaque mise à jour.

### Objectifs à atteindre

1. Organisez votre espace de travail en créant un dossier nommé « td-citations », structuré avec des sous-dossiers css et js sous assets pour les styles et les scripts respectivement.
2. Construisez une page index.html intégrant :
  - Un conteneur principal dédié à la citation, incluant un élément pour le texte et un autre pour l'auteur de la citation.
  - Un composant facultatif affichant le délai restant avant la prochaine actualisation pour une meilleure visualisation de la temporisation.
3. Stylisez votre carte de citation via un fichier CSS, en veillant à un design épuré et centré, avec une typographie lisible et une adaptation automatique du contraste par rapport à la couleur d'arrière-plan.
4. En JavaScript, la collection des citations est fournie dans un fichier externe data.js. Vous devez copier intégralement le contenu de ce fichier et l'inclure dans votre projet afin d'accéder au tableau de citations depuis votre script principal.
5. À chaque renouvellement, mettez à jour simultanément la citation affichée et la couleur de fond de la carte. Veillez strictement à ce que la nouvelle citation soit différente de la précédente, et que la couleur de fond choisie soit également distincte de celle utilisée juste avant. Pour les couleurs, vous

pouvez utiliser une palette prédéfinie ou générer des nuances aléatoires tout en contrôlant leur répétition immédiate.

6. Utilisez les fonctions `setInterval()` ou une combinaison judicieuse avec `setTimeout()` pour assurer la périodicité de 30 minutes, avec possibilité d'ajuster ce délai pour les phases de tests.
7. (En option) Mettez en place un indicateur de compte à rebours dynamique indiquant le temps restant avant le prochain changement, mis à jour en temps réel au moyen de fonctions répétitives.