TP : Manipulation du DOM en JavaScript

Projet 1 : Générateur de citations

🔹 **Point de départ :**

Un conteneur avec une citation affichée, un champ pour ajouter une nouvelle citation, et deux boutons.

```
<!-- Projet 1: Générateur de citations  -->
<div id="quote-generator">
  <h2>Citation motivante</h2>
  <p id="quote-display">Cliquez pour une citation !</p>
  <button onclick="getRandomQuote()">Nouvelle citation</button>
  <input type="text" id="new-quote" placeholder="Ajouter une citation">
  <button onclick="addQuote()">Ajouter</button>
</div>
```

Objectif :

- Créez un **tableau contenant plusieurs citations**.
- Implémentez la fonction `getRandomQuote()` pour afficher une citation aléatoire dans `<p id="quote-display">`.
- Implémentez la fonction `addQuote()` pour ajouter une nouvelle citation au tableau.
- À chaque ajout, **affichez le tableau des citations en format JSON dans la console.**

Projet 2 : Chronomètre / Minuteur

🔹 Point de départ :

```
<!-- Projet 2: Chronomètre -->
<div id="chrono">
  <h2>Chronomètre</h2>
  <p id="time">00:00:00</p>
  <button onclick="startTimer()">Start</button>
  <button onclick="stopTimer()">Stop</button>
  <button onclick="resetTimer()">Reset</button>
</div>
```

Objectif :

- Implémentez `startTimer()` pour lancer le chrono avec `setInterval()`.
- `stopTimer()` doit arrêter le chrono.
- `resetTimer()` remet le compteur à zéro.
- Le format du temps doit être **hh:mm:ss**.
- Bonus : utilisez `Date` pour formater ou calculer.

Projet 3 : Générateur de mot de passe

🔹 Point de départ :

```
<!-- Projet 3: Générateur de mot de passe -->
<div id="password-generator">
```

```
  <h2>Générateur de mot de passe</h2>
  <label for="length">Longueur :</label>
  <input type="number" id="length" min="4" max="20" value="8">
  <button onclick="generatePassword()">Générer</button>
  <p id="password-result"></p>
</div>
```

Objectif :

- Créez la fonction `generatePassword()`
- Le mot de passe doit contenir :
  - des lettres majuscules et minuscules
  - des chiffres
  - des symboles spéciaux (`!@#$%^&*`)
- Longueur personnalisable à partir du champ `#length`
- Le résultat s'affiche dans `#password-result`

Projet 4 : Formulaire de contact intelligent

🔹 Point de départ :

```
<!-- Projet 4: Formulaire de contact -->
<div id="contact-form">
  <h2>Contact</h2>
  <input type="text" id="name" placeholder="Nom"><br>
  <input type="email" id="email" placeholder="Email"><br>
  <textarea id="message" placeholder="Votre message..."></textarea><br>
  <button onclick="submitForm()">Envoyer</button>
  <p id="error-message" style="color: red;"></p>
</div>
```

Objectif :

- Implémentez `submitForm()` pour :
  - Vérifier que les **champs ne sont pas vides**
  - Vérifier que **l'email est valide** via une **expression régulière**
- Afficher un **message d'erreur** s'il manque quelque chose
- Si tout est bon, afficher les infos dans la console sous forme **JSON**

Projet 5 : Convertisseur de température

🔹 Point de départ :

```
<!-- Projet 5: Convertisseur de température -->
<div id="temp-converter">
  <h2>Convertisseur °C → °F</h2>
  <input type="number" id="celsius" placeholder="Température en °C">
  <button onclick="convertTemperature()">Convertir</button>
  <p id="result"></p>
</div>
```

Objectif :

- Implémentez `convertTemperature()` :

- Récupérer la température saisie
- Convertir en Fahrenheit : `F = C * 9/5 + 32`
- Afficher le résultat dans `#result`
- Modifier la **couleur du fond de la page** selon la température :
  - Froid : bleu
  - Tempéré : vert
  - Chaud : rouge