

**Objectif :** Expérimenter la transmission d'informations à l'aide de formulaires, utilisation d'un serveur pour recevoir des chaînes de requête, premières interactions avec les éléments du DOM à l'aide de javascript.

**Plan de la semaine :**

- Réponse aux questions sur TEA
- Test de connaissances sur le cours
- Exercice Try It Editor

Nous souhaitons implémenter le formulaire proposé sur le site de W3Schools permettant de tester la syntaxe des langages HTML et CSS, et l'améliorer.

Dans chaque exercice, on affichera le numéro de version de la page à gauche de la bannière, lorsqu'elle sera survolée, sans modifier le centrage de ses éléments.

<div style="display: flex; justify-content: space-between; align-items: center;"><span>Version 4</span><div style="display: flex; align-items: center;"><input style="border: 1px solid #ccc; padding: 2px 10px; margin-right: 5px;" type="button" value="Actualiser"/><input checked="" style="margin-right: 5px;" type="checkbox"/><span>automatiquement ↕</span></div></div>	
<div style="border: 1px solid #ccc; min-height: 200px; margin-bottom: 5px;"><div style="border-bottom: 1px solid #ccc; padding-bottom: 5px; margin-bottom: 5px;">interieur du div edition</div></div>	<div style="border: 1px solid #ccc; min-height: 200px; margin-bottom: 5px;"><div style="border-bottom: 1px solid #ccc; padding-bottom: 5px; margin-bottom: 5px;">interieur du div edition</div></div>

1. 1ère version : Un champ textarea permet d'éditer du code HTML, Un div placé à côté montre le rendu correspondant lorsque l'utilisateur clique sur un bouton.
  - On réalisera ce travail en 3 étapes : structure, présentation, interaction ;
  - On validera les couches structure et présentation sur <https://validator.w3.org/> et <http://jigsaw.w3.org/css-validator/> ;
  - On pensera à utiliser des variables globales pour enregistrer des références vers les cadres d'édition et d'affichage, que l'on initialisera à la fin du chargement de la page ;
  - On placera systématiquement une trace d'exécution au début de chaque fonction javascript.

2. 2ème version : Une case à cocher est placée à côté du bouton. Elle est associée au label 'automatiquement'. Si la case est cochée, nous souhaitons mettre en place un mécanisme permettant une mise à jour à chaque modification du champ textarea. Nous utiliserons l'événement keyup.
  - Quelles sont les différences entre keyup, keydown et keypress ?
  - Pourquoi l'événement onchange n'est pas conseillé pour détecter les changements ?
  - Comment connaître l'état de la case à cocher ?
  - [UX] Vérifier que la mise à jour se déclenche immédiatement au moment du clic sur la case ;
  - On décochera la case en utilisant la touche « ESC ».
    - Utiliser un “observateur” dans la console de développement pour afficher les propriétés de l'objet event représentant le “contexte”. Quelles propriétés permettent de connaître la touche appuyée ? Laquelle est la plus standard d'après le site <https://caniuse.com> ?
3. 3ème version : Afficher des infos-bulles lorsque l'utilisateur déplace sa souris au dessus du bouton, du label ou de la case à cocher
  - L'info-bulle doit se présenter comme une boîte écrite en blanc sur fond noir, située à proximité de la position de la souris au moment du survol
  - Comment s'assurer que l'info-bulle sera toujours au premier plan du navigateur quelle que soit sa position dans la page ?
  - Utiliser un seul gestionnaire d'événement onmouseover en le plaçant sur la bannière qui contient les contrôles de rafraîchissement ? Quel mécanisme du navigateur est utilisé ? Quelles propriétés du contexte de cet événement sont utiles dans ce cas ?
  - L'info-bulle doit disparaître deux secondes après être apparue. utiliser la méthode window.setTimeout. Quels sont les arguments de cette méthode ? Quelle est leur nature ?
4. 4ème version : Nous souhaitons que les mises à jour automatiques se déroulent à intervalles réguliers, en utilisant la méthode window.setInterval.
  - [UX] On informera l'utilisateur que quelquechose se passe en arrière plan, en affichant une image de chargement (que l'on peut récupérer sur le site <http://www.ajaxload.info/>) ou dans le répertoire fourni sur moodle.
  - On désactivera la mise à jour automatique en utilisant la touche « ESC ».
  - [UX] Lorsque l'on rafraîchit la fenêtre du navigateur alors que la case était cochée, elle reste cochée. Mettre en place ce qu'il faut pour que dans ce cas l'actualisation périodique se poursuive.
5. 5ème version : Quel attribut peut-on utiliser pour créer des info-bulles en pur HTML ? Nous souhaitons que les info-bulles soient entièrement réalisées en CSS. Proposez la feuille de style appropriée, sachant que nous souhaitons employer la structure suivante :

(...)

```
<div class="popable">
  <input type="button" value="Actualiser" />
  <div class="popstr">
    Cocher pour actualiser automatiquement
  </div>
</div>
```

(...)

6. 6ème version : Nous souhaitons “versionner” les différentes saisies de l'utilisateur à chaque appui sur le bouton “Actualiser”, et permettre des retours avant et arrière pour retrouver les textes saisis précédemment. Proposez une solution de stockage de ces textes en cache dans votre navigateur, et ajoutez des flèches et des raccourcis clavier permettant de mettre en oeuvre ce mécanisme.