

TP 2

EFREI 2A WEB

Louis Cherel - 05/2019

Préambule

Merci de lire **toutes les consignes**, elles sont écrites pour être lues.

Les TP sont tous ramassés

Tous les TP sont ramassés et doivent être rendus au plus tard à 23h59 le soir de la séance sur le [formulaire de dépôt](#) prévu à cet effet.

Chaque jour de retard vous retire 1 point sur votre note de TP, avec une limite à 3 jours de retard, ce qui, passé ce délai, vous vaudra une note de 0 sur votre TP.

Seul un des TP sera noté, mais ne pas rendre les deux autres vous vaudra quand même 0.

Les TP et le projet sont à réaliser en binôme ou trinômes, et doivent être enregistrés sur le [formulaire d'enregistrement d'équipes](#) avant la fin de la première séance.

Sans votre enregistrement d'équipe, vous ne pourrez pas rendre votre travail.

Il n'y a pas de DE. Votre note de module sera constituée à 50% de la note de TP et 50% de la note de projet.

[Cf les consignes du TP 1 pour le but et le séquençage des TP.](#)

Les ressources conseillées

Vos armes seront donc le Mozilla Developer Network (MDN), Stack Overflow, et les recherches en anglais. **Évitez au maximum** les forums (pas les cours) des sites comme OpenClassrooms ou Comment Ça Marche, qui proposent des solutions souvent peu fiables. W3Schools peut être une bonne ressource, mais les bonnes pratiques et les standards ne sont pas leur fort. Leur préférer le MDN lorsque possible.

Quantité de blogs sont très bons, notamment alsacreations (en français).

Lectures préliminaires

Les lectures **obligatoires** avant de commencer le TP:

- [Commencer avec le HTML \(MDN\)](#)
- [Le fonctionnement des CSS \(MDN\)](#)

Certains exercices nécessitent des lectures annexes, prenez soin de les lire avant de commencer le TP, même si les questions vous semblent simples.

Exercices

Exercice 1 (🕒 5 min lecture + 🕒 15 min pratique)

Lecture préalable nécessaire: [Propriété background](#) (MDN)

- 1) Créez un fichier .html avec votre éditeur préféré.
- 2) Créez une page web avec les éléments de base (html, head, body)
- 3) Donnez lui un titre avec la balise **<title>**
- 4) Ajoutez un entête à votre page (<header>)
- 5) Ajoutez-y une image (idéalement un logo, carré)
- 6) Grâce au CSS, faites en sorte que cette image soit centrée horizontalement par rapport à la page, et qu'elle fasse une taille d'à peu près 100px (hauteur et largeur)
- 7) Toujours grâce au CSS, faites en sorte que l'image devienne ronde
- 8) Ajoutez une légère ombre à votre image
- 9) Faites en sorte que votre image ne soit pas collée au bord de la page, verticalement
- 10) Faites en sorte que votre page aie une couleur de fond qui soit une teinte légère de gris

Vous devriez atteindre quelque chose qui ressemble à cela:



Exercice 2 (🕒 10 min lecture + 🕒 30 min pratique)


Lecture préalable nécessaire:

- [Les formulaires HTML](#) (MDN)
- [Comment structurer un formulaire](#) (MDN)

- 1) À la suite de votre image, toujours dans l'entête, ajoutez un formulaire qui contient les champs suivants: prénom, nom, email, rôle (enseignant, étudiant (utilisez <select>))
- 2) Ajoutez un texte de remplissage lorsque le champ est vide, et qui disparaît lorsque l'utilisateur entre lui-même du texte
- 3) Ajoutez deux boutons "Ajouter" et "Tout supprimer" à votre formulaire
- 4) Appliquez le style *width: 100%; border: 1px solid black;* à votre formulaire, puis ajoutez-lui une marge intérieure de 30px. Que remarquez-vous ?
- 5) Si vous rajoutez le style [box-sizing: border-box](#) à votre formulaire, que se passe-t-il ?
- 6) Pour le reste du TP, retirez les propriétés de la question 4 et 5, et ajoutez en haut de votre style le bloc suivant:

* { box-sizing: border-box; }
- 7) Ajoutez une légère ombre à votre formulaire, ainsi qu'une marge intérieure d'une dizaine de pixels. Le fond de votre formulaire doit être blanc
- 8) Centrez horizontalement votre formulaire sur la page, et faites en sorte qu'il prenne toute la largeur de la page, mais pas plus de 1000px. Vous pouvez vérifier comment votre formulaire réagit aux différentes largeurs de page grâce à l'outil "**Vue adaptative**" dans la console de développement de votre navigateur.
- 9) Faites en sorte que vos boutons soient sur une ligne à part, séparée des champs de texte, et ajoutez une légère marge autour des boutons afin de bien les espacer

Vous devriez atteindre quelque chose qui ressemble à cela:



<input type="text" value="Louis"/>	<input type="text" value="Cherel"/>	<input type="text" value="louis.cherel@platypus.academ"/>	Enseignant ▼
<input type="button" value="Ajouter"/>	<input type="button" value="Tout supprimer"/>		

Exercice 3 (🕒 15 min lecture + 🕒 30 min pratique)

Lectures préalables nécessaires:

- [Les flexbox](#) (MDN)
- [La validation des formulaires](#) (MDN)

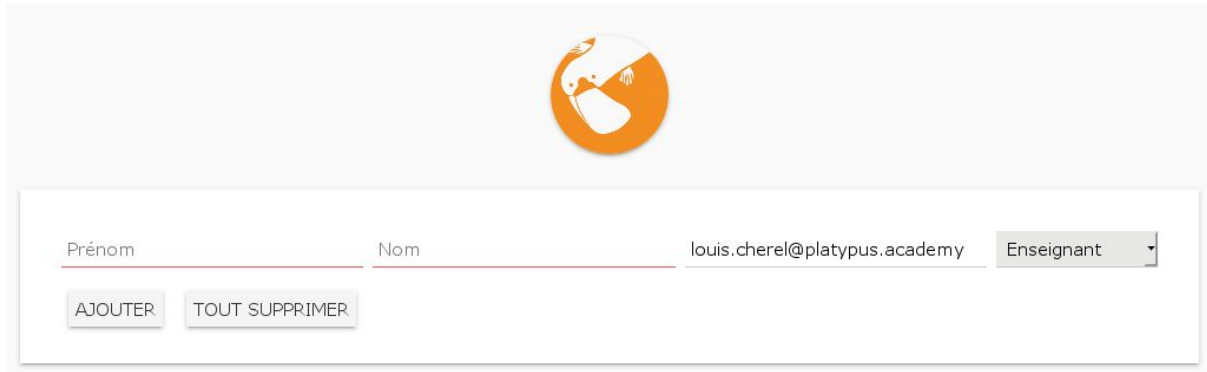
Ressources utiles:

- [Guide sur les flexbox](#) (CSS Tricks)
- [Material Design](#) (Google)

- 1) Faites en sorte que la ligne de champs de formulaire prenne tout l'espace disponible en largeur, quelle que soit la largeur de votre formulaire, grâce aux flexbox. La largeur de chacun des champs doit s'adapter automatiquement
- 2) Enlevez les bordures des champs de texte, et ajoutez une simple bordure, avec une légère teinte de gris en bas de chacun des champs. Une marge intérieure ne sera pas de trop, et même une marge extérieure pour ne pas coller chacun des champs
- 3) Faites en sorte que lorsque la largeur de la fenêtre est trop petite, les champs retournent à la ligne, toujours grâce aux flexbox
- 4) Faites en sorte que lorsque l'utilisateur n'a pas rentré correctement les informations dans les champs (nom, prénom vides, email non correct), la bordure du bas du champ ainsi que le "caret" (le curseur qui clignote) soient rouges
- 5) Transformez vos boutons pour que, quel que soit le texte, la police soit toujours en majuscules. Enlevez les bords des boutons, ajoutez une ombre légère, ainsi qu'une couleur de fond légèrement grisée, et une marge intérieure
- 6) Faites en sorte que lorsque l'utilisateur survole le bouton, sa couleur change (gris plus foncé)

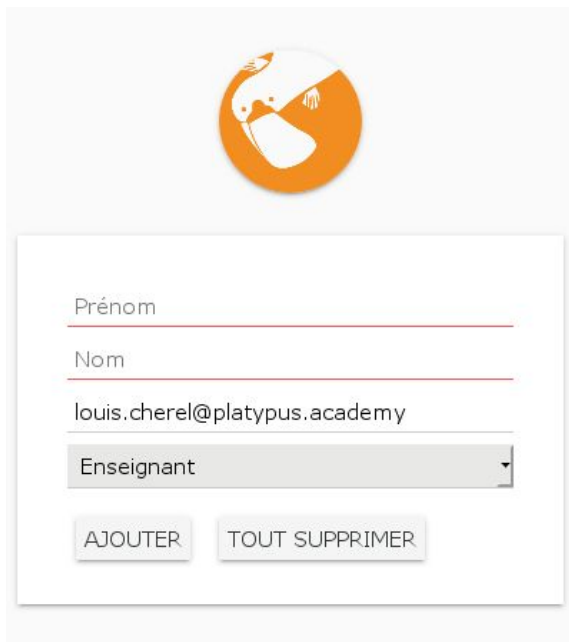
- 7) Changez la police d'écriture de tout le texte de votre page pour quelque chose de plus sympathique, comme du **Verdana**, par exemple
- 8) Note: vous pouvez éditer le style du formulaire comme bon vous semble, les consignes précédentes étant surtout un guide pour un formulaire utilisant le [Material Design](#)

Vous devriez arriver à quelque chose comme ceci:



The desktop version of the form features a circular logo at the top center. Below it, the form is organized into a single row with three input fields: 'Prénom', 'Nom', and an email field containing 'louis.cherel@platypus.academy'. To the right of the email field is a dropdown menu labeled 'Enseignant'. Below these fields are two buttons: 'AJOUTER' and 'TOUT SUPPRIMER'.

Et en version mobile:



The mobile version of the form displays the same circular logo at the top. The input fields are stacked vertically: 'Prénom', 'Nom', the email field with 'louis.cherel@platypus.academy', and the 'Enseignant' dropdown menu. The 'AJOUTER' and 'TOUT SUPPRIMER' buttons remain at the bottom.

Exercice 4 (🕒 30 min lecture + 🕒 20 min pratique)

Lectures préalables:

- [Qu'est-ce que le JavaScript ?](#) (MDN)
- [Comment ajouter du JavaScript à votre page ?](#) (MDN)
- [Notre premier code JavaScript](#) (MDN)

- 1) À la suite de votre header, ajoutez le contenu principal (<main>) de votre page. Faites en sorte de réutiliser le style appliqué à votre formulaire via une classe commune (couleur, largeur, ombre, etc.)
- 2) Dans votre conteneur principal, ajoutez un tableau, dont les colonnes sont Prénom, Nom, Email, et Rôle. Le tableau doit faire toute la largeur du conteneur, avec une légère marge afin de ne pas les "coller"

Notre but ici va être de faire en sorte de pouvoir ajouter une ligne à notre tableau HTML lorsque l'utilisateur clique sur le bouton "Ajouter". Nous allons devoir faire appel à du JavaScript. Lorsque vous travaillerez avec ce langage, pensez toujours à ouvrir la **Console** dans votre navigateur, pour voir les informations d'erreur ou les affichages console (console.log())

- 3) Créez une balise script en bas de votre page (juste avant la balise fermante </body>), et utilisez la fonction console.log() à l'intérieur pour afficher du contenu dans votre console. Vérifiez qu'il apparaît correctement
- 4) Concevez une fonction ajouter() en javascript qui appelle la fonction alert() (popup).
- 5) Appelez ajouter() juste après l'avoir définie, pour vérifier qu'elle produit bien le résultat escompté
- 6) Faites en sorte que cette fonction soit appelée lorsque l'utilisateur appuie sur le bouton "Ajouter" (mais pas sur les autres boutons)

Exercice 5 (🕒 15 min lecture + 🕒 40 min pratique)

Lecture(s) préalable(s) nécessaire(s):

- [Document.forms](#) (MDN)
- [Document.querySelector\(\)](#) (MDN)
- [HTMLInputElement#methods](#) (MDN)
- [Element.innerHTML](#) (MDN)
- [Document.createElement\(\)](#) (MDN)
- [node.appendChild\(\)](#) (MDN)
- [node.textContent](#) (MDN)

- 1) Toujours avec la fonction ajouter(), faites en sorte d'afficher la valeur du champ *prénom* grâce à console.log() lorsque l'utilisateur clique sur le bouton Ajouter
- 2) Dans cette même fonction, récupérez l'élément <table> que vous avez créé, et affichez-le avec console.log(). Que constatez-vous ?

- 3) Créez 1 élément "tr" avec `Document.createElement()`, et conservez-le dans une variable
- 4) Faites de même avec 4 éléments "td", dont vous pouvez définir la valeur avec `textContent`. Ajoutez (`appendChild`) ces éléments à votre nouvel élément tr
- 5) Ajoutez enfin votre tr à votre balise table récupérée lors de la question 2. Si tout se passe bien, votre page HTML devrait avoir évolué
- 6) Grâce à tous ces éléments, faites en sorte que lorsque l'utilisateur clique sur "Ajouter", une ligne du tableau apparaisse avec le contenu du formulaire
- 7) Trouvez une façon de faire en sorte que la ligne n'apparaisse que lorsque le formulaire est correctement rempli
- 8) Faites en sorte que lorsque l'utilisateur clique sur "Tout supprimer", le tableau soit vidé des entrées précédemment ajoutées

Exercice 6 (🕒 5 min lecture + 🕒 10 min pratique)

Lecture préalable nécessaire: [Pseudo-classes et pseudo-éléments \(MDN\)](#)

- 1) Centrez le texte de votre tableau
- 2) Ajoutez des marges intérieures aux cellules afin d'aérer le texte
- 3) Ajoutez une bordure en bas de chacune des lignes de votre tableau, mais pas la dernière
- 4) Faites en sorte que lorsque la largeur de fenêtre est trop petite, votre tableau ne dépasse pas de son cadre, mais permette de scroller horizontalement
- 5) Faites en sorte que lorsque l'utilisateur scrolle vers le bas de la page et que le header n'est plus visible, le formulaire reste visible

Exercice 7 (🕒 10 min lecture + 🕒 20 min pratique)

Lecture préalable nécessaire: [setInterval](#) (MDN)

- 1) Faites en sorte que l'utilisateur ne puisse valider le formulaire que 10 secondes après l'ouverture de la page ou 10 secondes après la dernière validation du formulaire
- 2) Affichez le décompte dans le bouton "Ajouter" entre parenthèses, il doit s'actualiser toutes les secondes

- 3) Faites en sorte que le bouton aie l'air "désactivé" lorsque l'utilisateur doit attendre
- 4) Si l'utilisateur clique avant les dix secondes imparties, utilisez la fonction `alert()` afin de l'avertir de son mauvais comportement

Pour aller plus loin

Désactivez le `alert()` afin de pouvoir réaliser les exercices suivants

Exercice 8 (🕒 > 20 min pratique)

- 1) Trouvez un moyen pour que, lorsque l'utilisateur clique plus de 5 fois en moins de 2 secondes, le formulaire disparaisse pendant 5 secondes, afin que l'utilisateur ne puisse **vraiment** plus cliquer sur le bouton !
- 2) Ajoutez un bouton à votre formulaire qui effectue une rotation à 180° de votre formulaire (oui, oui, une [rotation](#)). Il faut que la transition soit jolie et fluide. Lorsqu'on reclique sur le bouton, le formulaire doit retrouver son orientation originale

Exercice 9 (🕒 > 20 min pratique)

- 1) Trouvez une façon de faire en sorte que, lorsque l'utilisateur tape un code secret dans l'un des champs du formulaire, le curseur se transforme en cible (pensez à une carabine), et quel que soit l'élément de la page sur lequel l'utilisateur clique, cet élément disparaisse (attention à chercher la précision, on ne veut pas supprimer tout le tableau d'un coup)