Henallux – DA1 – web : principes de base

Laboratoire 5

Formulaires HTML

Sommaire

[1. (Exo) Quiz sur le langage HTML 2](#_Toc160453494)

[2. (Exo) Évaluation d’un cours 11](#_Toc160453495)

# (Exo) Quiz sur le langage HTML

Objectifs

* construire un formulaire HTML
* construire une feuille de style pour un formulaire
* observer le fonctionnement des méthodes GET et POST

Pour cet exercice, créez un nouveau document HTML à partir de zéro (vous pouvez l’appeler quiz.html).

## Étape 1 : le document de base

Dans un premier temps, rédigez les parties standards du document HTML, à savoir :

* déclaration !doctype ;
* balises <html>, <head> et <body> ;
* encodage des caractères, langue du document ;
* titre de l’onglet : « Quiz HTML ».

## Étape 2 : début de formulaire

Le quiz créé dans cet exercice va prendre la forme d’un formulaire composé de trois sections : des informations personnelles et deux questions. En plus de ces champs, il comportera deux boutons : un bouton permettant de réinitialiser / remettre à zéro le contenu du formulaire et un second bouton permettant d’envoyer les informations encodées.

Commençons par la première partie : la fiche d’informations personnelles. Dans la partie <body> du document HTML, insérez un formulaire (via la balise <form>). Pour le moment, ne vous préoccupez pas des attributs à ajouter à cette balise ; vous pouvez n’en mettre aucun à ce stade.

À l’intérieur de cette balise, ajoutez une liste non ordonnée <ul> avec 3 éléments <li> contenant les composantes suivantes :

1. A white rectangular object with black text

   Description automatically generatedl’étiquette (<label>) « Prénom » suivie d’un champ textuel pour recevoir le prénom de l’utilisateur ;
2. l’étiquette « Nom » suivie d’un champ textuel pour recevoir le nom de l’utilisateur ;
3. l’étiquette « Adresse mail » suivie d’un champ textuel de type email pour recevoir l’adresse mail de l’utilisateur ;
4. une boîte à cocher (checkbox) suivi de l’étiquette « C’est la première fois que je réponds à ce quiz. »

Pour améliorer la lisibilité du formulaire, ajoutez deux règles CSS (en interne ou dans un fichier externe, au choix).

1. La première règle ciblera les listes non ordonnées situées dans un formulaire et utilisera la propriété list-style: none, qui permet de supprimer les puces devant les éléments de la liste. Ajoutez-y également des propriétés pour réduire la marge et le rembourrage (margin et padding) à 0 pour supprimer les espaces inutiles autour de la liste.
2. La seconde règle ciblera les éléments à l’intérieur de telles listes et leur ajoutera une marge (un espacement) de 12 pixels sur le haut, via margin-top: 12px.

## Étape 3 : bouton de soumission

Dans le formulaire, en-dessous de la liste non ordonnée, ajoutez un bouton de soumission. Vous pouvez utiliser, au choix, une balise <button> (conseillé pour la clarté) ou une balise <input/>.

Le texte à afficher est « Envoyer » et le bouton doit être associé à l’action de soumission de informations.

Pour pouvoir tester l’envoi de données, il faut tout d’abord compléter les informations données dans la balise <form>. Revenez donc à cette balise et ajoutez les attributs

action="https://testphpig3.000webhostapp.com/showGetPost.php" method="GET"

Le fichier en question, showGetPost.php, est un script PHP qui se contente de récupérer toutes les données reçues via GET ou POST pour créer une table HTML qui les cite. La réponse renvoyée est un page HTML qui contient cette table.

Vous trouverez ce script dans les ressources du laboratoire. Si vous préférez, vous pouvez placer ce fichier sur le site que vous avez créé à la fin du laboratoire précédent (dans le répertoire public\_html) et utiliser l’adresse de votre site au lieu de celle renseignée ci-dessus.

Testez votre formulaire sous Firefox. Sauf si vous avez déjà pris un peu d’avance sur l’étape suivante, l’envoi de données ne devrait rien donner : le script PHP affiche une liste vide. Par contre, même à ce stade, si vous tentez d’entrer quelque chose qui ne ressemble pas à une adresse mail dans le troisième champ du formulaire, Firefox devrait bloquer l’envoi et vous signaler le problème.

## Étape 4 : données à envoyer

Vous l’avez peut-être déjà fait de vous-même (si vous avez été attentif au cours) mais, pour que le formulaire envoie des données, il faut préciser les clefs auxquelles ces données seront associées.

Chaque champ du formulaire peut envoyer ses données sous la forme d’une paire clef/valeur. La valeur correspond à ce que l’utilisateur va entrer… mais la clef, elle, doit être précisée dans le code HTML, via l’attribut name.

Ajoutez à chacun des quatre champs de votre formulaire un attribut name avec, comme valeur, un identificateur qui servira de clef pour l’information associée. Par exemple :

* firstname pour le prénom,
* lastname pour le nom,
* mail pour l’email,
* et firsttime pour la boîte à cocher.

Une fois que c’est fait, testez à nouveau votre formulaire et observez le résultat envoyé par le script PHP.

## Étape 5 : première validation, Get et POST

Assurez-vous de tester les cas suivants :

1. Vous entrez des informations correctes dans chaque champ.
2. Vous entrez, pour l’adresse mail, un texte qui n’a pas la forme d’une adresse valide.
3. Observez l’information récupérée selon que la checkbox est cochée ou pas.
4. Observez l’URL indiquée par votre navigateur lorsqu’il affiche la réponse du script PHP et faites le lien avec ce qui a été dit au cours à propos de la méthode GET.
5. Quand la réponse du script PHP est affichée, rafraîchissez la page sur Firefox (F5). Notez que le rafraîchissement se fait sans problème.
6. Remplacez dans votre code la méthode GET par POST, rechargez le formulaire sous Firefox et envoyez les données. Examinez l’URL cette fois-ci.
7. Avec la réponse du script PHP affichée, rafraîchissez à nouveau la page et observez le message que Firefox affiche.

## Étape 6 : plus de confort

Revenez à votre formulaire.

Pour éditer un champ, il faut cliquer dans le cadre prévu pour recevoir l’information. Si vous cliquez sur l’étiquette (comme par exemple sur les mots « Adresse mail »), rien ne se passe. C’est parce que les étiquettes ne sont pas encore liées aux champs.

Pour établir ces liens, il faut ajouter

* dans chaque champ <input/> : un identificateur via l’attribut id, et
* dans les étiquettes, préciser l’identificateur du champ lié via l’attribut for.

Modifiez le code HTML pour établir ces liens en utilisant les identificateurs

* prenom pour le prénom,
* nom pour le nom,
* mail pour l’email,
* et premiere pour la boîte à cocher.

Notez que, en pratique, on utilise souvent le même identificateur à la fois pour l’attribut name (la clef utilisée lors de l’envoi des données) et pour l’attribut id (son identificateur au sein de la page HTML) mais rien n’y oblige. Et, dans tous les cas, il faut préciser les deux attributs explicitement.

Après avoir effectué ces modifications, testez l’effet d’un clic sur une étiquette.

## Étape 7 : bloc d’étude

L’objectif de cette étape est d’ajouter dans le formulaire un champ correspondant au bloc d’étude (1re, 2e ou 3e) de l’utilisateur.



Dans la liste <ul>, ajoutez un cinquième item <li> contenant

* une étiquette « Bloc d’étude »,
* trois boutons radios,
* trois étiquettes, une par bouton radio (vous pouvez utiliser la balise <sup> pour encadrer les parties de texte à écrire en exposant).

Pour que les trois boutons fonctionnent comme un seul groupe, il faut qu’ils portent le même attribut name (par exemple "bloc"). Ils devraient, par contre, avoir des identificateurs id différents, pour pouvoir les associer chacun à l’étiquette correspondante.

Vérifiez que vous avez bien associé les trois étiquettes des blocs aux boutons radios : l’utilisateur devrait pouvoir cliquer soit sur le bouton lui-même soit sur l’étiquette pour sélectionner une des trois options.

Si vous vous contentez de préciser le nom et l’id des trois options, vous verrez que les informations envoyées au serveur ne sont guère utilisables : quel que soit le bloc choisi, le serveur reçoit la même information, à savoir : bloc=on.

Pour envoyer une information utilisable, il faut indiquer la valeur à associer à la clef bloc pour chacune des trois options. Ajoutez à chacun des trois boutons un attribut value avec des valeurs qui permettront de les distinguer (par exemple "1", "2" et "3").

Vérifiez dans la réponse du serveur que l’information associée à la clef bloc est désormais bien utilisable.

## Étape 8 : réinitialisation

Revenez ensuite au formulaire HTML et ajoutez un second bouton de type reset intitulé Réinitialiser.

Ouvrez votre formulaire sous Firefox et utilisez ce bouton pour revenir à l’état par défaut du formulaire (a priori, tous les champs vides).

Sans rien modifier au formulaire, cliquez ensuite sur le bouton d’envoi. Les informations affichées indiquent une valeur (vide) pour trois champs, mais aucune information sur le bloc d’étude. C’est parce que, par défaut, aucun des boutons radios n’est coché.

Pour éviter ce genre de situation, il est généralement préférable d’activer par défaut un des boutons radios. Dans votre fichier HTML, ajoutez l’attribut booléen checked pour l’un des trois blocs.

Comme il n’est pas possible de simplement décocher un bouton radio (sans en cocher un autre), cela signifie qu’à l’envoi du formulaire, la clef bloc sera toujours associée à une valeur.

## Étape 9 : structuration du formulaire

Utilisez des balises <fieldset> pour marquer les trois sections du formulaire :

* un premier fieldset regroupant les « Données personnelles » (tous les champs qui se trouvent actuellement dans votre formulaire – n’oubliez pas de soigner l’indentation !) ; puis
* deux autres fieldsets vides (pour le moment) pour « Question 1 » et « Question 2 ».

Les deux boutons (réinitialisation et envoi) se trouveront après et en-dehors des fieldsets.

Dans chaque fieldset, ajoutez une balise <legend> reprenant le titre du fieldset.

A screenshot of a computer

Description automatically generated

## Étape 10 : question 1

Complétez le fieldset de la question 1 pour qu’il corresponde à la capture d’écran suivante. Pour obtenir l’écriture en police monotype, vous pouvez encadrer les textes des réponses dans une balise <code>.

A white rectangular object with black text

Description automatically generated

Regroupez les sélections possibles dans une liste <ul> (la règle CSS définie plus haut devrait s’appliquer automatiquement).

N’oubliez pas de préciser les identificateurs et noms nécessaires. Faites également en sorte que les étiquettes soient correctement associées aux choix (afin qu’on puisse cliquer soit sur la case à cocher soit sur le texte).

Comme il s’agit d’un choix multiple, il ne s’agit plus de boutons radios mais de checkboxes distinctes : chacune aura donc un name distinct !

Testez votre formulaire et observez les informations qui sont envoyées au serveur.

## Étape 11 : question 2

La question 2 porte sur l’écriture d’un hyperlien en HTML.

A screenshot of a computer

Description automatically generated

Complétez le dernier fieldset pour y inclure cette question.

Pour la liste d’options, utilisez une balise <select> à l’intérieur de laquelle les quatre options de réponse seront citées. N’oubliez pas d’associer une valeur à chacune des options – c’est cette valeur qui sera communiquée au serveur avec la clef du select.

Testez votre formulaire et observez les informations qui sont envoyées au serveur.

## Étape 12 : quelques détails de plus

Réalisez une à une les modifications suivantes. N’oubliez pas d’en tester le résultat à chaque fois.

1. Ajoutez un placeholder pour le champ email. Celui-ci comportera le texte « Entrez votre adresse mail ici. » Testez le comportement du placeholder quand vous entrez du texte et quand, ensuite, vous le supprimez.
2. Rendez les champs nom et prénom obligatoires.
3. Faites en sorte que, par défaut (et après réinitialisation du formulaire), la boîte à cocher « première fois que je réponds à ce quiz » soit cochée.
4. Faites en sorte que, par défaut (et après réinitialisation du formulaire), la 4e réponse de la question 2 soit sélectionnée.

## Étape 13 : un peu de style

Ajoutez des règles CSS en suivant les consignes suivantes.

1. Imposez une largeur de 600 px pour le formulaire (ciblez la balise <form>).
2. Arrangez-vous pour que les fieldset aient une couleur de fond « papayawhip », une bordure de 2 pixels d’épaisseur en traits solides de couleur rouge, un arrondi de coin (border-radius) de 6 pixels. Ajoutez-leur également une ombre grise (avec la propriété box-shadow : 4px 4px 2px gray) et un espace vers le bas (margin-bottom : 12px).
3. Faites en sorte que les titres des fieldsets soient écrits en utilisant une fonte de type cursive (propriété font-family), à 200% de la taille normale et en gras. Ajoutez-leur aussi une ombre rouge grâce à la propriété text-shadow : 2px 2px red.
4. Ciblez les champs dans lesquels il est possible d’écrire (c’est-à-dire le select ainsi que les input de type text ou email ici). Donnez-leur une couleur de fond « wheat », une bordure orange en trait plein de 1px d’épaisseur et un arrondi de coin de 6 pixels.
5. Ciblez les boutons. Faites en sorte que leur texte utilise une police de type cursive, ajoutez-leur une bordure de 2px d’épaisseur en trait plein de couleur rouge, un arrondi d’angle de 6 pixels et une ombre box-shadow : 2px 2px 1px red. Pour les coller contre le bord droit, ajoutez aussi la propriété float : right. Ajoutez une marge à gauche de 6px pour les séparer légèrement.

Vous devriez obtenir un rendu final similaire à celui de la capture d’écran suivante.

A screenshot of a computer screen

Description automatically generated

1. Ajoutez une règle CSS pour mettre en évidence les champs qui ne sont pas valides, en changeant la couleur de leur cadre en rouge.
2. Un peu plus difficile que la précédente : faites également en sorte que les étiquettes des champs non valides soient écrites en rouge (voir ci-dessous).  
     
   A screenshot of a computer

   Description automatically generated

# (Exo) Évaluation d’un cours

Objectifs

* créer un formulaire HTML et rédiger les styles associés
* construire une page web à partir d’un design donné
* examiner quelques options avancées relatives aux formulaires HTML

L’objectif de cet exercice est de réaliser un formulaire HTML permettant (de manière simplifiée) d’évaluer une matière de cours. Il se décompose en deux parties :

* dans un premier temps, vous devrez construire la page web uniquement à partir d’une capture d’écran du rendu final désiré et de quelques indications ;
* dans un second temps, vous pourrez ajouter à votre formulaire quelques éléments plus avancés relatifs aux formulaires HTML.

## Étape 1 : voilà ce qu’on veut obtenir, à vous de jouer !

Consultez le modèle présenté ci-dessous (voir aussi les indications qui suivent) et construisez un document dont l’apparence sera aussi proche que possible. Cela nécessitera d’utiliser non seulement du code HTML mais également des règles CSS.

A screenshot of a computer screen

Description automatically generated

Voici quelques indications et remarques pour vous aider à obtenir ce rendu.

* **Couleurs et styles.** En plus des couleurs standards, on a utilisé beige (pour le fond) et powderblue. Les boutons, quand ils sont survolés par le curseur de la souris, passent à une couleur de fond powderblue. La transition s’effectue progressivement (ce qui peut se traduire par la propriété transition: background-color 1s à placer sur le bouton lui-même).
* **Organisation interne.** Pour les groupes relatifs à la « Matière évaluée » et aux « Commentaires libres », on a utilisé des paragraphes (un pour chaque groupe), avec un retour à la ligne forcé <br/>. Les deux boutons eux aussi sont regroupés dans un paragraphe.  
  Pour le choix de la situation, on a utilisé une liste non ordonnée. Pour les quatre évaluations, on a utilisé une table à deux colonnes afin d’aligner facilement les champs numériques ; cette table est centrée horizontalement (via margin-left : auto et margin-right : auto).
* **Taille des éléments.** Le formulaire occupe une largeur de 400 pixels (avec un rembourrage de 8 pixels pour écarter les bords). Le champ pour la matière évaluée est prévu pour recevoir 50 caractères. Les fieldsets sont séparés verticalement par une marge (vers le haut) de 8 pixels. Les champs d’évaluation numériques s’étendent sur une largeur de 40 pixels, avec une marge à gauche de 16 pixels. La zone de commentaire peut accueillir 4 lignes de 44 caractères.
* **Valeurs par défaut et placeholders.** Les deux champs textuels (matière évaluée et commentaires) comportent un placeholder. Par défaut, la seconde option de situation est sélectionnée (cours suivi mais pas encore passé l’examen). Par défaut, chaque évaluation est réglée à une cote moyenne : 3.
* **Remarque.** N’oubliez pas de lier les étiquettes et les champs !
* **Données à envoyer.** Pour tester votre formulaire, vous pouvez utiliser la même adresse d’envoi que dans l’exercice précédent (le script PHP showGetPost.php). Voici un exemple pour le format de données attendu :  
   A white and orange box with black text

  Description automatically generated  
  Les noms de clefs doivent être respectés. Pour la situation, on s’attend à recevoir une valeur parmi : passuivi, pasexamen, réussite et échec. Les notes d’évaluation sont des entiers entre 1 et 5.

## Étape 2 : autocomplétion

À l’intérieur de la balise <form>, juste avant le paragraphe correspondant à la matière évaluée, ajoutez les lignes suivantes (vérifiez l’indentation !).

<datalist id="matieres">

<option value="Web - HTML et CSS"></option>

<option value="Web - JavaScript"></option>

<option value="Web - PHP"></option>

<option value="Principes de programmation"></option>

<option value="Principes de programmation, compléments"></option>

<option value="Principes de programmation, algo avancée"></option>

<option value="Programmation - C"></option>

<option value="Programmation - Java"></option>

<option value="Programmation - C#"></option>

</datalist>

Il s’agit d’une liste de suggestions qui pourront être utilisées pour faciliter l’introduction de données via auto-complétion.

Deux remarques à ce sujet :

* L’endroit où la liste datalist est définie importe peu… mais, pour faciliter la lecture du document HTML, c’est une bonne idée de la placer à proximité de l’endroit où elle va être utilisée (dans ce cas-ci, le champ pour la matière évaluée).
* Il s’agit bien ici d’une question de confort via l’autocomplétion : le fait de proposer une liste d’options prédéfinies n’empêche pas l’utilisateur d’entrer autre chose dans le champ !

Pour indiquer que cette liste est à utiliser pour le champ concernant la matière évaluée, ajoutez dans la balise <input/> correspondante l’attribut list="matières".

Rechargez ensuite la page sous Firefox.

À première vue, rien n’a changé. Mais, si vous double-cliquez dans le champ « Matière évaluée » (quand celui-ci est vide), Firefox devrait vous proposer la liste des options citées.

Si vous y tapez du texte, Firefox vous proposera également une liste d’options d’autocomplétion. Testez en entrant simplement la lettre « P ». Puis en continuant avec « Progra ».

## Étape 3 : champs numériques

Sous Firefox, modifiez l’une des évaluations sans utiliser les boutons permettant d’augmenter ou de diminuer la valeur mais directement en tapant la valeur numérique dans le champ.

Effectuez un premier test avec une valeur entière (par exemple 4).

Testez ensuite des valeurs non valides ; par exemple, 9 puis 3.5. À chaque fois, cliquez sur le bouton de soumission des données pour observer le comportement de Firefox.

Éditez le code HTML en ajoutant un attribut step avec une valeur de 0.5 (entre guillemets bien sûr) à l’un des champs numériques d’évaluation, puis effectuez à nouveau les tests précédents.

## Étape 4 : glissière numérique

À l’intérieur de la balise <form>, juste avant le paragraphe correspond aux commentaires, ajoutez le code suivant (ajustez l’indentation si nécessaire).

<p>

<label for="eval">Évaluation générale</label><br/>

<input id="eval" name="eval" type="range" min="1" max="5"/><br/>

<output id="outputeval" for="eval"></output>

</p>

Observez le résultat dans Firefox : les <input/> de type range permettent d’obtenir des glissières correspondant à un résultat numérique. Testez son utilisation. Vérifiez également les données reçues par le serveur.

Ajoutez à la balise de la glissière l’attribut step="0.1" et testez le changement sous Firefox.

La glissière numérique est un élément d’interface intéressant mais elle ne permet pas automatiquement à l’utilisateur d’avoir une idée précise de la valeur qu’il a choisie. Il n’y a pas de méthode simple pour remédier à ce problème, mais on peut générer un affichage en retour en utilisant JavaScript.

Cela dépasse la matière correspondant à ce cours mais, pour avoir un aperçu de ce que ça donne, ajoutez le code suivant dans l’en-tête du document HTML (donc à l’intérieur de la balise <head>).

<script>

const evalOutputs = [

"Mauvais",

"Piètre",

"Moyen",

"Bon",

"Excellent",

];

let elemEval, outputEval;

window.addEventListener("load", init);

function init () {

elemEval = document.getElementById("eval");

outputEval = document.getElementById("outputeval");

elemEval.addEventListener("input", updateEvalOutput);

updateEvalOutput();

}

function updateEvalOutput () {

if (elemEval.value < 0 || elemEval.value > 5) {

outputEval.innerHTML = "Valeur hors limite";

return;

}

const description = evalOutputs[Math.round(elemEval.value) - 1];

outputEval.innerHTML = `${elemEval.value} - ${description}`;

}

</script>

Notez qu’on peut également utiliser une datalist avec une glissière numérique. Dans ce cas-là, il ne s’agit bien sûr pas d’une fonctionnalité d’autocomplétion ; au lieu de cela, des encoches sont ajoutées pour marquer les valeurs citées dans la datalist.

Vous pouvez tester cette option en utilisant la datalist suivante :

<datalist>

<option value="1"></option>

<option value="2"></option>

<option value="3"></option>

<option value="4"></option>

<option value="5"></option>

</datalist>

Pour ce faire, vous devrez

* ajouter ce code à un endroit judicieux dans le fichier HTML ;
* donner un identificateur à la datalist ;
* établir le lien vers cette datalist dans le champ correspondant à la glissière numérique.