

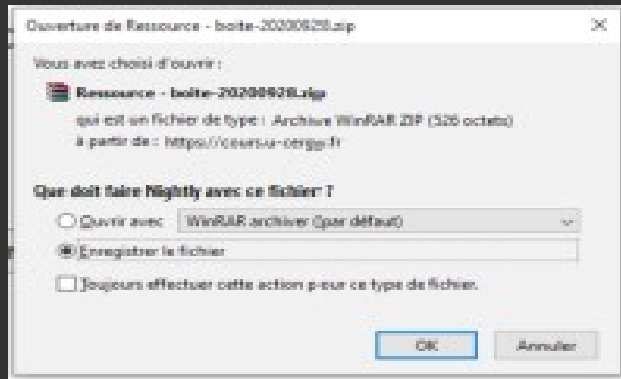
Intégration Web

MMI 1 – TP#6 S1

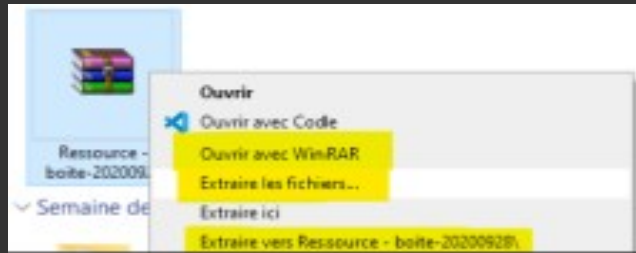


Danielo **JEAN-LOUIS**
Michele **LINARDI**

Petit point sur les ressources de cours

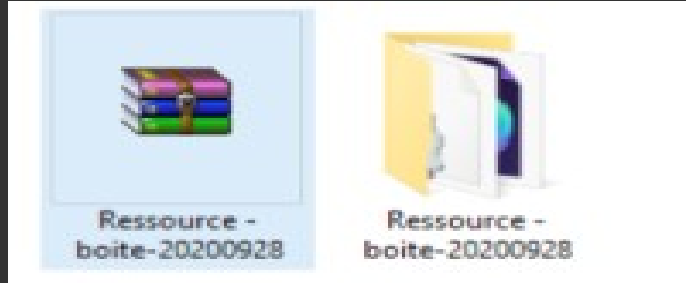


- Les ressources se présentent sous la forme d'une archive une fois téléchargées
- L'archive doit être impérativement extraite sinon vous ne serez pas capables de réaliser correctement les tp



Menu contextuel sur l'archive (clic droit). Choisir un des choix surligné.

Petit point sur les ressources de cours



- Une fois extraite, vous devez obtenir un dossier avec le contenu de l'archive
- Libre à vous de travailler dans le dossier OU en copier le contenu pour le coller ailleurs
- **En aucun vous devez travailler dans l'archive**
- Une fois fait, glissez le dossier dans VS Code ou Sublime Text

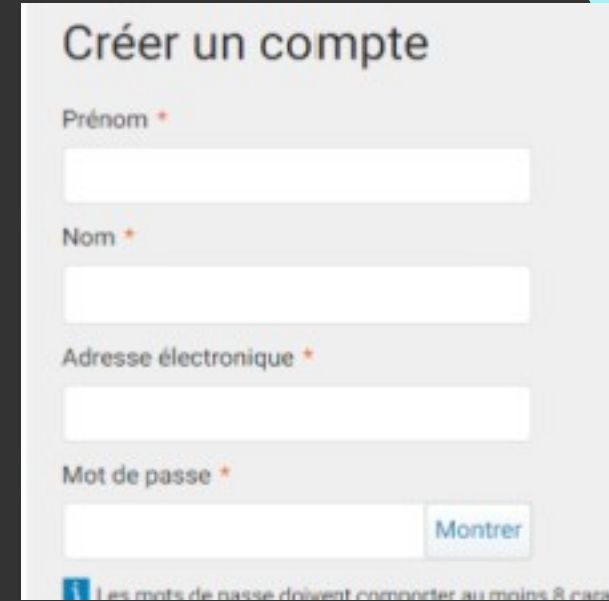
Entrons dans le web 2.0

Formulaires

- Outil permettant à l'utilisateur de saisir des données pour les envoyer à un serveur
- Ensemble de balises contenues dans une balise `<form>`
 - **Tous les éléments d'un formulaire doivent s'y trouver**
 - **Balise indispensable**

Formulaires

- Exemple de fonctionnalités réalisables avec un formulaire :
 - Formulaire de connexion / inscription
 - Clavardage (ou chat)
 - [...]
- Balise pairée : `<form></form>`



The image shows a web form titled "Créer un compte". It contains four input fields: "Prénom" (First name), "Nom" (Last name), "Adresse électronique" (Email address), and "Mot de passe" (Password). Each field has a red asterisk indicating it is required. The "Mot de passe" field has a "Montrer" (Show) button next to it. At the bottom, there is a small blue icon and a note: "Les mots de passe doivent comporter au moins 8 caractères" (Passwords must be at least 8 characters long).

Cette fonctionnalité est impossible sans un formulaire

```
<form action="traitement.php" method="get">  
  <!-- Contenu du formulaire -->  
</form>
```

Exemple de squelette de formulaire

Balise <form> - Attribut "method"

- Permet de définir la façon dont les données sont envoyées
- Possède la valeur "GET" par défaut
 - Valeur d'attribut insensible à la casse

Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Form>
- <https://developer.mozilla.org/fr/docs/Learn/Forms>

Balise <form> - Attribut "method"

- Trois valeurs possibles :
 - GET – Valeur par défaut
 - POST
 - DIALOG

Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Form>
- <https://developer.mozilla.org/fr/docs/Learn/Forms>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/form#attr-method>

Attribut "method" – Valeur "GET"

- Données passent dans l'url / barre d'adresse du navigateur
 - **Données visibles à l'envoi**
- Limitation à 255 caractères historiquement
 - ~2 000 caractères de nos jours

Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Form>
- <https://developer.mozilla.org/fr/docs/Learn/Forms>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/form#attr-method>

Attribut "method" – Valeur "GET"



The image shows a dark-themed web browser address bar. The URL `https://cours-prod.cyu.fr/mod/quiz/review.php?attempt=334955` is displayed in a light gray font. A yellow rectangular box highlights the query string portion of the URL, which is `?attempt=334955`. A vertical yellow line extends from the bottom of this box down towards the explanatory text.

```
https://cours-prod.cyu.fr/mod/quiz/review.php?attempt=334955
```

Tout ce qui se trouve après
"?" dans une URL
représente des données
envoyées par GET

Ici le serveur recevra une
variable "attempt" avec la
valeur "334955"

Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Form>
- <https://developer.mozilla.org/fr/docs/Learn/Forms>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/form#attr-method>

Attribut "method" – Valeur "GET"

- Ne permet pas d'envoyer des medias
- Valeur par défaut de l'attribut "method"
- Données envoyées vers le serveur à la soumission du formulaire

Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Form>
- <https://developer.mozilla.org/fr/docs/Learn/Forms>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/form#attr-method>

Attribut "method" – Valeur "GET"

- Exemples d'utilisation :
 - Test sur ENT
 - Recherche sur google
- Prend la forme :
 - mon.url?**val1=toto&val2=3**

Valeurs de votre formulaire
(envoyées au serveur)

Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Form>
- <https://developer.mozilla.org/fr/docs/Learn/Forms>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/form#attr-method>

Attribut "method" – Valeur "POST"

- Données passent dans le corps de la requête
 - **Données invisibles à l'envoi***
 - **Permet d'envoyer des données sensibles**
 - **Ne garde pas les données envoyées en mémoire**
- Pas de limite de caractères
- Permet d'envoyer des medias (images, vidéos...)
 - Il faudra rajouter un attribut spécial pour ce faire

* : A condition que le serveur soit https

Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Form>
- <https://developer.mozilla.org/fr/docs/Learn/Forms>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/form#attr-method>

Attribut "method" – Valeur "POST"

- Données envoyées vers le serveur à la soumission du formulaire
- Exemples d'utilisation :
 - Formulaire d'achat en ligne
 - Upload d'image sur imgur

Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Form>
- <https://developer.mozilla.org/fr/docs/Learn/Forms>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/form#attr-method>

Attribut "method" – Valeur "DIALOG"

- Cache le formulaire lors de la soumission
 - Uniquement si le formulaire est contenu dans une balise <dialog>
- Données ne sont pas envoyées vers le serveur à la soumission du formulaire
 - **Nécessite d'envoyer soi-même les données vers le serveur**

Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Form>
- <https://developer.mozilla.org/fr/docs/Learn/Forms>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/form#attr-method>

Attribut "method" – Valeur "DIALOG"

- Affiche le formulaire dans une modale
 - Le formulaire s'affiche au-dessus de la page

Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Form>
- <https://developer.mozilla.org/fr/docs/Learn/Forms>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/form#attr-method>

Balise <form> - Traitement des données

- L'attribut "action" permet de définir où les données vont être envoyées
- Données traitées sous forme de paires clés / valeurs
 - Clé définie par l'attribut "name"
 - **Tous les éléments de formulaire en ont besoin**
 - Valeur définie par l'attribut "value" (sauf pour la balise <textarea></textarea>)

Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Form>

Balise <form>

Endroit où les données vont
être envoyées et traitées

```
<form action="traitement.php" method="get">  
  <!-- Contenu du formulaire -->  
</form>
```

Comment les données
vont être envoyées

Éléments de formulaire

- Ensemble de balises permettant à l'utilisateur de remplir les formulaires
- Balise la plus courante : `<input>`
 - Très polyvalente
- Fonctionnent de pair avec la balise `<label>`
- **Doivent être dans une balise `<form>`**

Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Form>

Balise <input />

- Balise auto-fermante
- **N'accepte que du texte monoligne**
 - Ne permet pas de faire un retour à la ligne
- Attribut "value" permet de définir une valeur par défaut
 - Valeur souvent en provenance du serveur

Sources :

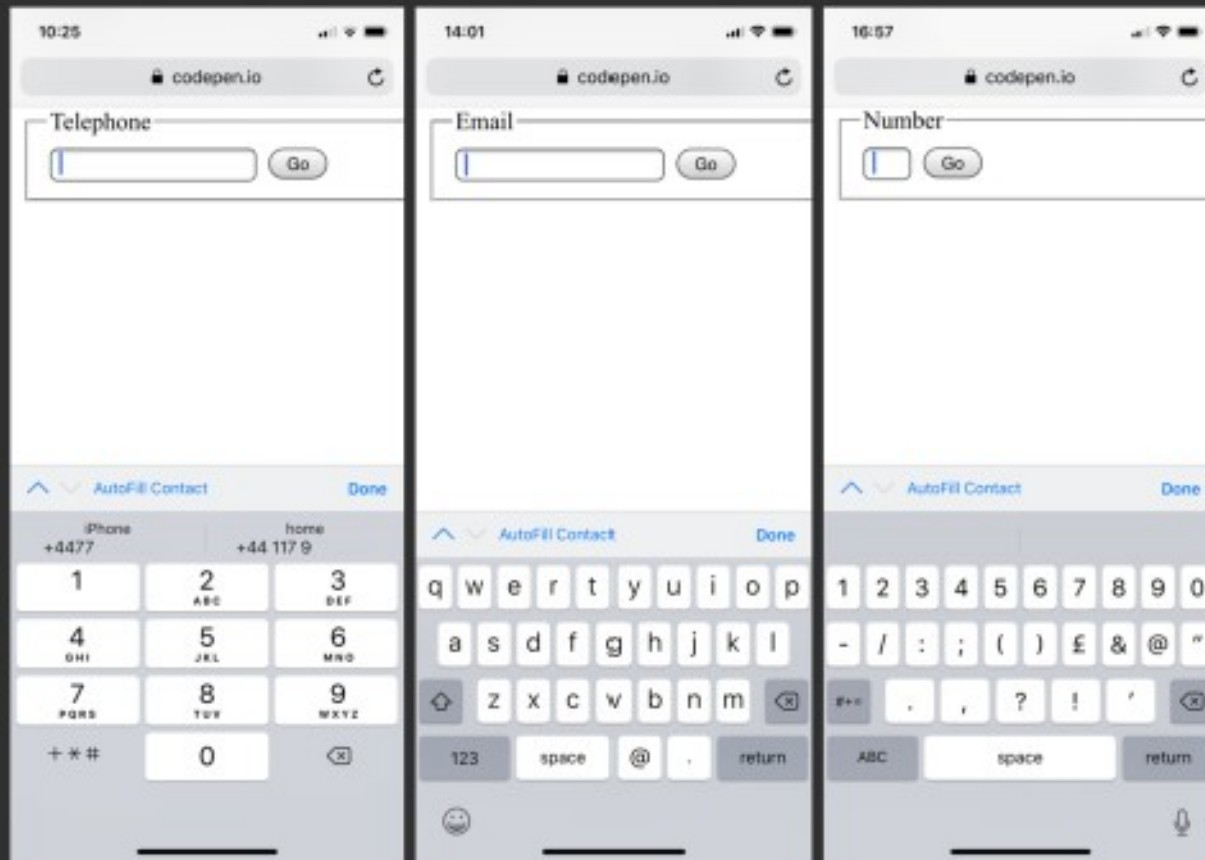
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Input/text>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Input>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Attributes/autocomplete>

Balise <input />

- Attribut "type" pour définir le type de données qui peut être entré ou le comportement du champ
 - text : texte
 - number : nombre
 - password : mot de passe
 - email : courriel (nouveau HTML5)
 - tel : téléphone (nouveau HTML5)
 - hidden : champ caché
 - [...]

Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Input/text>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Input>



Sur mobile (ici iOS) le clavier change en fonction du champ dépendamment de valeur de son attribut "type"

Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Input/text>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Input>
- <https://www.smashingmagazine.com/2019/01/html5-input-types/?ref=heydesigner> - anglais

Balise <input />

- Exemple d'utilisation :
 - Champ pour un pseudonyme
 - Champ pour un numéro de téléphone

Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Input/text>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Input>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Attributs/autocomplete>

Balise <label></label>

- Décrit le rôle d'un champ pour un être humain
 - **Ne doit en aucun cas être substitué par l'attribut "placeholder" de la balise <input />**

Sources :

- <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/label>
- https://www.w3schools.com/howto/howto_css_form_icon.asp - mauvaise pratique

Point accessibilité – l'attribut placeholder

- Doit uniquement servir à afficher un exemple de données attendues dans un champ
- **Ne doit pas remplacer la balise <label>**

Sources :

- <https://www.anysurfer.be/fr/blog/detail/lelement-placeholder-nest-pas-un-label>
- https://www.w3schools.com/howto/howto_css_form_icon.asp - mauvaise pratique
- <http://www.ohmyband.fr/CANDIDAT.html?anchor=form%20candidat>

Balise <label></label>

- **Attribut "for" indispensable pour l'accessibilité**
 - Associe un label à un champ
Le champ associé doit avoir l'attribut "id" avec la même valeur
 - Permet au clic de mettre le focus sur le champ associé

Sources :

- <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/label>
- https://www.w3schools.com/howto/howto_css_form_icon.asp - mauvaise pratique

Bonnes pratiques – Champ de formulaire

- Un champ (input) doit impérativement avoir :
 - Un label qui lui est associé (avec l'attribut "for")
 - Les attributs suivants
 - Un attribut "id" avec une valeur identique à celle de son label associé
 - Un attribut "name" avec une valeur unique
 - Un attribut "type", de préférence en accord avec la donnée attendue

Balise <label></label>

```
<label for="prenom">Prénom</label>  
<input type="text" name="name" id="prenom" />
```

Associe le **label** à un **champ (input)** possédant la même valeur pour l'attribut **"id"**

Associe le **champ (input)** à un **label** possédant la même valeur pour l'attribut **"for"**

Sources :

- <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/label>
- https://www.w3schools.com/howto/howto_css_form_icon.asp - mauvaise pratique

Pratiquons ! - Notre premier formulaire (Partie 1)

Pré-requis :

- Avoir la ressource `ressources/formulaire-basique`

A télécharger ici :

<https://download-directory.github.io/?url=https://github.com/DanYellow/cours/tree/main/integration-web-s1/travaux-pratiques/numero-6/ressources>

Soumission du formulaire

- Deux choix possibles :
 - Balise `<button>` avec la valeur "submit" pour l'attribut "type"
 - Attention : en absence d'attribut "type" le navigateur tranche en fonction du contexte quant à son comportement
 - Balise fermante : Permet donc de mettre du texte + image dans le bouton par exemple

Soumission du formulaire

- Balise `<input>` avec la valeur "submit" pour l'attribut "type" et l'attribut "value" pour le libellé
- Troisième possibilité : appuyer sur la touche "Entrée"
 - Ne fonctionne que si le formulaire est accessible → Présence d'une balise `<form>`

Soumission du formulaire

- Permet d'envoyer les données vers un serveur
 - Attribut "action" de la balise <form>
- Se place dans le formulaire à la fin de préférence

Pratiquons ! - Notre premier formulaire (Partie 2)

Pré-requis :

- Avoir la ressource `ressources/formulaire-basique`

A télécharger ici :

<https://download-directory.github.io/?url=https://github.com/DanYellow/cours/tree/main/integration-web-s1/travaux-pratiques/numero-6/ressources>

Balise `<textarea></textarea>`

- Balise acceptant du texte multi-lignes
 - Permet à l'utilisateur de faire des retours à la ligne
- La valeur par défaut n'est pas dans l'attribut "value" mais entre les deux balises

Source :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/textarea>

Balise <textarea></textarea>

- Utilisation des propriétés "rows" et "cols" pour définir le nombre de lignes et colonnes visibles
 - Possibilité d'utiliser les attributs "width" et "height" également
- La balise, par défaut, peut se redimensionner
 - **Très mauvaise idée de désactiver cette option**

Source :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/textarea>

Balise `<textarea></textarea>`

- Exemples d'utilisation :
 - Rédaction de commentaire
 - Rédaction de messages
 - [...]

Source :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/textarea>

Balise <textarea></textarea>

```
<label for="commentaire">Commentaire</label>  
<textarea name="comment" id="commentaire" cols="30" rows="10"></textarea>
```

Exemple de <textarea> avec son <label> associé

Source :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/textarea>

Éléments à cocher

- Permettent d'effectuer des choix parmi des choix prédéfinis
- Balise input avec une valeur spécifique pour l'attribut "type" :
 - checkbox : Permet de sélectionner plusieurs éléments en même temps
 - radio : Permet de sélectionner qu'un seul élément

Source :

- https://developer.mozilla.org/fr/docs/Web/Guide/HTML/Formulaires/Les_blocs_de_formulaires_natifs#%C3%89l%C3%A9ments_%C3%A0_cocher

Éléments à cocher

- Doivent être associés à un label respectif pour qu'ils soient compréhensibles pour l'être humain
- Attribut "checked" permet de pré-sélectionner une valeur (ou plus)
- Nécessitent l'attribut "name" pour savoir quelle est le nom de la valeur envoyée

Source :

- https://developer.mozilla.org/fr/docs/Web/Guide/HTML/Formulaires/Les_blocs_de_formulaires_natifs#%C3%89l%C3%A9ments_%C3%A0_cocher

Element à cocher – input radio

- Nécessite la valeur "radio" pour l'attribut "type" de la balise `<input />`
- Ne permet de sélectionner qu'un seul choix

Source :

- https://developer.mozilla.org/fr/docs/Learn/Forms/Basic_native_form_controls#bouton_radio

Element à cocher – input radio

- Ne permet pas de désélectionner un choix
- Les choix liés doivent avoir la même valeur pour l'attribut "name" et une valeur unique pour l'attribut "value"

Source :

- https://developer.mozilla.org/fr/docs/Learn/Forms/Basic_native_form_controls#bouton_radio

Element à cocher – input radio

- Peut remplacer la balise `<select>` quand les choix sont limités
- Exemples d'utilisation :
 - Choix exclusif
 - VISA ou MasterCard
 - Entrée ou dessert

Source :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/textarea>

Element à cocher – input radio

Exemple d'input type "radio"

```
<input type="radio" id="gaucher" name="habilite" value="gaucher">  
<label for="gaucher">Gaucher</label><br>  
<input type="radio" id="droitier" name="habilite" value="droitier">  
<label for="droitier">Droitier</label><br>  
<input type="radio" id="ambidextre" name="habilite" value="ambidextre">  
<label for="ambidextre">Ambidextre</label><br>
```

Valeur "radio" pour avoir
le comportement attendu

Les trois input **ont la même valeur pour
l'attribut "name"** conséquemment seul un
choix parmi les trois peut être sélectionné

Source :

- https://developer.mozilla.org/fr/docs/Learn/Forms/Basic_native_form_controls#bouton_radio

Element à cocher – input radio

Exemple d'input type "radio"

```
<input type="radio" id="gaucher" name="habilite" value="gaucher">  
<label for="gaucher">Gaucher</label><br>  
<input type="radio" id="droitier" name="habilite" value="droitier">  
<label for="droitier">Droitier</label><br>  
<input type="radio" id="ambidextre" name="habilite" value="ambidextre">  
<label for="ambidextre">Ambidextre</label><br>
```

Chaque input possède une valeur unique pour l'attribut "value"

Source :

- https://developer.mozilla.org/fr/docs/Learn/Forms/Basic_native_form_controls#bouton_radio

Element à cocher – Case à cocher

- Nécessite la valeur "checkbox" pour l'attribut "type" de la balise `<input />`
- Permet de sélectionner plusieurs choix
- Permet de désélectionner un choix
- Deux façons d'envoyer les données vers le serveur : tableau ou valeurs uniques

Source :

- https://developer.mozilla.org/fr/docs/Web/Guide/HTML/Formulaires/Les_blocs_de_formulaires_natifs#%C3%89l%C3%A9ments_%C3%A0_cocher

Case à cocher – Envoi des données

- Tableau

```
<input type="checkbox" name="menu[]" value="plat" id="plat" />  
<label for="plat">Plat</label>  
  
<input type="checkbox" name="menu[]" value="fromage" id="fromage" />  
<label for="fromage">Fromage</label>
```

La valeur de l'attribut "name" est suffixée de "[]" pour envoyer les valeurs sous forme de tableau.

La valeur de l'attribut est partagée entre tous les éléments liés

Case à cocher – Envoi des données

- Valeurs uniques

```
<input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">  
<label for="vehicle1"> I have a bike</label><br>  
<input type="checkbox" id="vehicle2" name="vehicle2" value="Car">  
<label for="vehicle2"> I have a car</label><br>
```

La valeur de l'attribut "name" est unique à chaque checkbox.

Chaque donnée est envoyée indépendamment

Element à cocher – input radio

- Exemples d'utilisation :
 - Choix multiples
 - framadate / doodle → sélection de dates

Source :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/textarea>

Pratiquons ! - Notre premier formulaire (Partie 3)

Pré-requis :

- Avoir la ressource `ressources/formulaire-basique`

A télécharger ici :

<https://download-directory.github.io/?url=https://github.com/DanYellow/cours/tree/main/integration-web-s1/travaux-pratiques/numero-6/ressources>

Liste déroulante – Balise `<select>`

- Permet de sélectionner un choix parmi une liste de choix définis
- Peut se substituer à la balise `<input type="radio">` quand il y a trop de choix
- Permet de sélectionner plusieurs choix en même temps
 - A éviter car ce n'est pas naturel pour l'utilisateur

Source :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/select>

Liste déroulante – Balise <select>

- Ne peut contenir que des balises <option>
 - Contient (balise option) l'attribut "value"
 - Doit impérativement être dans une balise <select>, <optgroup> ou <datalist>
- Possibilité de grouper les options via la balise <optgroup>

Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/select>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/option>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Optgroup>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/datalist>

Liste déroulante – Balise `<select>`

- Attribut "selected" pour présélectionner un élément
- Possibilité de personnalisation très limitée
 - Se limiter qu'au select au global.
 - Ne pas faire un `<select>` custom → Crée des problèmes pour l'accessibilité

Source :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/select>

Liste déroulante – Balise <select>

- Exemple d'utilisation :
 - Liste de choix longue :
 - Sélection d'un pays parmi la liste de tous les pays

Source :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/select>

Liste déroulante – Balise <select>

```
<label for="pet-select">Choose a pet:</label>

<select name="pets" id="pet-select">
  <option value="">--Please choose an option--</option>
  <option value="dog">Dog</option>
  <option value="cat">Cat</option>
  <option value="hamster">Hamster</option>
  <option value="parrot">Parrot</option>
  <option value="spider">Spider</option>
  <option value="goldfish">Goldfish</option>
</select>
```

L'attribut "name" est défini
au niveau de la balise
<select>

La valeur est définie au
niveau de la balise "option"

Point accessibilité – Personnalisation

- **Beau n'est pas égal à fonctionnel**
- Personnaliser des éléments de formulaire pour qu'ils soient beaux est une très mauvaise idée si ça ne les rend pas clairs
- Supprime / endommage le côté accessible des éléments

Sources :

- https://www.w3schools.com/howto/howto_css_form_icon.asp

Pratiquons ! - Notre premier formulaire (Partie 4)

Pré-requis :

- Avoir la ressource `ressources/formulaire-basique`

A télécharger ici :

<https://download-directory.github.io/?url=https://github.com/DanYellow/cours/tree/main/integration-web-s1/travaux-pratiques/numero-6/ressources>

Validation des formulaires

- Do not trust user input / Ne pas faire confiance aux entrées utilisateur
- Validation automatique mais... on peut tricher
 - **Une validation côté serveur est indispensable**

Source :

- https://developer.mozilla.org/fr/docs/Web/Guide/HTML/Formulaires/Validation_donnees_formulaire
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Form#attr-novalidate>

Validation des formulaires – Champs requis

- Attribut "required"
 - Empêche la validation d'un formulaire si la valeur du champ est absente
- Pseudo-classe ":required" pour styliser dans le CSS
 - ":valid" : quand le champ est valide
 - ":invalid" : quand le champ est invalide
 - ":optional" : quand le champ est facultatif
 - ":checked" : quand le champ a été coché

Sources :

- <https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/required> – anglais
- https://developer.mozilla.org/fr/docs/Web/Guide/HTML/Formulaires/Validation_donnees_formulaire

Pratiquons ! - Notre premier formulaire (Partie 5)

Pré-requis :

- Avoir la ressource `ressources/premier-formulaire`

A télécharger ici :

<https://download-directory.github.io/?url=https://github.com/DanYellow/cours/tree/main/integration-web-s1/travaux-pratiques/numero-7/ressources>

Pseudo-élément

- Permet d'ajouter du contenu sans pour autant ajouter de nouvelles balises
- Syntaxe :

```
.mon-selecteur::pseudo-element {  
    /* Déclarations CSS */  
}
```

Sources :

- <https://www.creativejuiz.fr/blog/css-css3/difference-entre-pseudo-element-et-pseudo-classe>
- <https://developer.mozilla.org/fr/docs/Web/CSS/Pseudo-%C3%A9l%C3%A9ments>
- <https://accessibleweb.com/question-answer/how-is-css-pseudo-content-treated-by-screen-readers/> - anglais
- <https://developer.mozilla.org/fr/docs/Web/CSS/::first-line>

Pseudo-éléments

- Vous ne pouvez pas associer plusieurs fois le même pseudo-élément à une balise
 - Le navigateur tranchera
- Ne s'affiche pas dans le code source
- Si vous vous trompez sur la syntaxe, le navigateur est suffisamment malin pour faire la différence entre le "::" et le ":" grâce au nom

Sources :

- <https://www.creativejuiz.fr/blog/css-css3/difference-entre-pseudo-element-et-pseudo-classe>
- <https://developer.mozilla.org/fr/docs/Web/CSS/Pseudo-%C3%A9l%C3%A9ments>
- <https://accessibleweb.com/question-answer/how-is-css-pseudo-content-treated-by-screen-readers/> - anglais

Pseudo-éléments

- `::before` / `::after` sont certainement ceux que vous utiliserez le plus

Sources :

- <https://www.creativejuiz.fr/blog/css-css3/difference-entre-pseudo-element-et-pseudo-classe>
- <https://developer.mozilla.org/fr/docs/Web/CSS/Pseudo-%C3%A9l%C3%A9ments>
- <https://accessibleweb.com/question-answer/how-is-css-pseudo-content-treated-by-screen-readers/> - anglais

Pseudo-éléments - ::before / ::after

- Permettent d'ajouter des éléments avant ou après un contenu
 - On peut mettre les deux en même temps sur le même élément
- N'existent (probablement) pas pour les lecteurs d'écran
 - **Ne pas placer de données critiques dedans**

Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/::after>
- <https://developer.mozilla.org/fr/docs/Web/CSS/::before>

Pseudo-éléments - ::before / ::after

- **Nécessite impérativement la propriété "content"** pour s'afficher. Valeur pouvant être :
 - une chaîne de caractères (qui peut être vide), une image...
 - Le texte mis n'est pas sélectionnable

Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/::after>
- <https://developer.mozilla.org/fr/docs/Web/CSS/::before>

Pseudo-éléments - ::before / ::after

- Ne fonctionne pas sur les éléments émulés comme `` ou `<input>`
- Ne permet pas d'afficher du code HTML
 - Elles seront lues comme étant du texte, elles ne seront pas interprétées

Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/::after>
- <https://developer.mozilla.org/fr/docs/Web/CSS/::before>

Pseudo-éléments - ::before / ::after

- Exemples d'utilisation :
 - Afficher des guillemets au tour d'un texte
 - Mettre des séparateurs entre les éléments d'un menu
 - Ajouter un * sur un label dont le champ est obligatoire
 - Et plein d'autres : <https://a.singlediv.com/>

Pratiquons ! - Notre premier formulaire (Partie 6)

Pré-requis :

- Avoir la ressource `ressources/premier-formulaire`

A télécharger ici :

<https://download-directory.github.io/?url=https://github.com/DanYellow/cours/tree/main/integration-web-s1/travaux-pratiques/numero-7/ressources>

Questions ?