codecademy

Formes

`<input>`: Type de case à cocher

Lors de l'utilisation d'un `input` élément HTML , l' `type="checkbox"` attribut affichera un seul élément de case à cocher. Pour créer un groupe de cases à cocher liées au même sujet, elles doivent toutes utiliser le même `name` attribut. Puisqu'il s'agit d'une case à cocher, plusieurs cases à cocher peuvent être sélectionnées pour le même sujet.

```
<input type="checkbox"
name="breakfast" value="bacon">Bacon
🥓<br>
<input type="checkbox"
name="breakfast" value="eggs">Eggs 🍳
<br>
<input type="checkbox"
name="breakfast"
value="pancakes">Pancakes 🥞<br>
```

`<textarea>` Élément

L' `textarea` élément est utilisé lors de la création d'une zone de texte pour une entrée multiligne (par exemple une section de commentaire). L'élément prend en charge les attributs `rows` et `cols` qui déterminent respectivement la hauteur et la largeur de l'élément.
Lorsqu'ils sont rendus par le navigateur, les `textarea` champs peuvent être étirés/réduits en taille par l'utilisateur, mais les attributs `rows` et `cols` déterminent la taille initiale.
Contrairement à l' `input` élément, l' `<textarea>` élément possède à la fois des balises d'ouverture et de fermeture. Le `value` de l'élément est le contenu entre ces balises (un peu comme un `<p>` élément). Le bloc de code montre un `<textarea>` de taille 10x30 et avec un `name` de `"comment"` .

```
<textarea rows="10" cols="30"
name="comment"></textarea>
```

`<form>` Élément

L' `<form>` élément HTML est utilisé pour collecter et envoyer des informations à une source externe. `<form>` peut contenir divers éléments d'entrée. Lorsqu'un utilisateur soumet le formulaire, les informations contenues dans ces éléments d'entrée sont transmises à la source qui est nommée dans l' `action` attribut du formulaire.

```
<form method="post"
action="http://server1">
  Enter your name:
  <input type="text" name="fname">
  <br/>
  Enter your age:
  <input type="text" name="age">
  <br/>
  <input type="submit"
value="Submit">
</form>
```

`<input>`: Type de numéro

Les éléments d'entrée HTML peuvent être de type `number` . Ces champs de saisie permettent à l'utilisateur de saisir uniquement des chiffres et quelques caractères spéciaux à l'intérieur du champ.
L'exemple de bloc de code montre une entrée avec un type `number` et un nom de `balance` . Lorsque le champ de saisie fait partie d'un formulaire, le formulaire recevra une paire clé-valeur avec le format : `name: value` après la soumission du formulaire.

```
<input type="number" name="balance"
/>
```

`<input>` Élément

L' `<input>` élément HTML est utilisé pour afficher une variété de champs de saisie sur une page Web, notamment des champs de texte, des cases à cocher, des boutons, etc. L' `<input>` élément possède un `type` attribut qui détermine comment il est rendu sur une page.
L'exemple de bloc de code créera un champ de saisie de texte et un champ de saisie de case à cocher sur une page Web.

```
<label for="fname">First name:
</label>
<input type="text" name="fname"
id="fname"><br>

<input type="checkbox" name="vehicle"
value="Bike"> I own a bike
```

`<input>`: Type de gamme

Un curseur peut être créé en utilisant l' `type="range"` attribut sur un `input` élément HTML . Le curseur de plage agira comme un sélecteur entre une valeur minimale et une valeur maximale. Ces valeurs sont définies à l'aide des attributs `min` et `max` respectivement. Le curseur peut être ajusté pour se déplacer par étapes ou incréments différents à l'aide de l' `step` attribut.
Le curseur de plage est destiné à agir davantage comme un widget visuel pour ajuster entre 2 valeurs, où la position relative est importante, mais la valeur précise n'est pas aussi importante. Un exemple de ceci peut être le réglage du niveau de volume d'une application.

```
<input type="range" name="movie-
rating" min="0" max="10" step="0.1">
```

`<select>` Élément

L' `<select>` élément HTML peut être utilisé pour créer une liste déroulante. Une liste de choix pour la liste déroulante peut être créée à l'aide d'un ou plusieurs `<option>` éléments. Par défaut, un seul `<option>` peut être sélectionné à la fois.
La valeur des `<select>` 's sélectionnés `name` et l' attribut `<option>` 's `value` sont envoyées sous forme de paire clé-valeur lors de la soumission du formulaire.

```
<select name="rental-option">
  <option
value="small">Small</option>
  <option value="family">Family
Sedan</option>
  <option value="lux">Luxury</option>
</select>
```

Soumettre un formulaire

Une fois que nous avons collecté des informations dans un formulaire, nous pouvons envoyer ces informations ailleurs en utilisant l' attribut `action` et `method` . L' `action` attribut indique au formulaire d'envoyer les informations. Une URL est attribuée qui détermine le destinataire des informations. L' `method` attribut indique au formulaire ce qu'il doit faire avec ces informations une fois qu'elles sont envoyées. Un verbe HTTP est affecté à l' `method` attribut qui détermine l'action à effectuer.

```
<form action="/index3.html"
method="PUT"></form>
```

<input>: Type de texte

Les `<input>` éléments HTML peuvent prendre en charge la saisie de texte en définissant l'attribut `type="text"` . Cela rend un champ de saisie à une seule ligne dans lequel les utilisateurs peuvent taper du texte.
La valeur du `<input>` 's `name` et l' `value` attribut de l'élément sont envoyés sous forme de paire clé-valeur lors de la soumission du formulaire.

```
<input type="text" name="username">
```

<datalist> Élément

Lors de l'utilisation d'une entrée HTML, une fonctionnalité de recherche/saisie semi-automatique de base peut être obtenue en associant un `<input>` avec un `<datalist>` . Pour apparier un `<input>` avec `<datalist>` la `<input>` de `list` valeur doit correspondre à la valeur du `id` du `<datalist>` . L' `datalist` élément est utilisé pour stocker une liste de `<option>` s.
La liste des données s'affiche sous forme de liste déroulante sur un `input` champ lorsqu'un utilisateur clique sur le champ de saisie. Au fur et à mesure que l'utilisateur commence à taper, la liste sera mise à jour pour afficher les éléments qui correspondent le mieux à ce qui a été tapé dans le champ de saisie. Les éléments de liste réels sont spécifiés en tant `option` qu'éléments multiples imbriqués dans le fichier `datalist` .
`datalist` s sont idéaux pour fournir aux utilisateurs une liste d'options prédéfinies, mais pour leur permettre également d'écrire des entrées alternatives.

```
<input list="ide">

<datalist id="ide">
  <option value="Visual Studio Code"
/>
  <option value="Atom" />
  <option value="Sublime Text" />
</datalist>
```

<input>: Type de bouton radio

Les `<input>` éléments HTML peuvent recevoir un `type="radio"` attribut qui affiche un seul bouton radio. Plusieurs boutons radio d'un sujet connexe reçoivent la même `name` valeur d'attribut. Une seule option peut être choisie parmi un groupe de boutons radio.
La valeur du sélectionné / vérifié `<input>` l » `name` et un `value` attribut de cet élément sont envoyés en tant que paire de valeurs de clé lorsque le formulaire est soumis.

```
<input name="delivery_option"
type="radio" value="pickup" />
<input name="delivery_option"
type="radio" value="delivery" />
```

Entrée à soumettre

Les `<input>` éléments HTML peuvent avoir un attribut type défini à soumettre, en ajoutant `type="submit"` . Avec cet attribut inclus, un bouton de soumission sera rendu et, par défaut, soumettra le `<form>` et exécutera son action.
Le texte d'un bouton d'envoi est défini `Submit` par défaut mais peut également être modifié en modifiant l' `value` attribut.

`<input>` name Attribut

Pour qu'un formulaire envoie des données, il doit pouvoir les mettre dans des paires clé-valeur. Ceci est réalisé en définissant l' `name` attribut de l' `input` élément. Le `name` deviendra le `key` et le `value` de la saisie deviendra `value` le formulaire soumis correspondant à la clé.
Il est important de se rappeler que le nom n'est pas le même que l'ID en termes de soumission de formulaire. L'ID et le nom de l'entrée peuvent être identiques, mais la valeur ne sera soumise que si l' `name` attribut est spécifié.
Dans l'exemple de code, la première entrée sera soumise par le formulaire, mais pas la seconde.

```
<input name="username" id="username"
/>
<input id="address" />
```

`<label>` Élément

L' `<label>` élément HTML fournit l'identification d'un élément spécifique en `<input>` fonction des valeurs correspondantes de l' attribut `<input>` 's `id` et de l' attribut `<label>` 's `for` . Par défaut, un clic sur le `<label>` focalisera le champ du `<input>` .
L'exemple de code créera un champ de saisie de texte avec le texte d'étiquette « Mot de passe : » à côté. En cliquant sur « Mot de passe : » sur la page, le champ correspondant au fichier `<input>` .

```
<label for="password ">Password:
</label>
<input type="text" id="password"
name="password">
```

`<input>` Type de mot de passe

L' `<input>` élément HTML peut avoir l'attribut `type="password"` qui restitue un champ de saisie à une seule ligne qui permet à l'utilisateur de saisir du texte censuré à l'intérieur du champ. Il est utilisé pour saisir des informations sensibles.
La valeur de cette `<input>` « s `name` et `value` (valeur réelle et non la version censuré) attribut de cet élément sont envoyés en tant que paire de valeurs de clé lorsque le formulaire est soumis.
Le bloc de code montre un exemple des champs d'un formulaire de connexion de base - les champs nom d'utilisateur et mot de passe.

```
<input type="text" name="username" />
<input type="password"
name="password" />
```

Validateurs de formulaire HTML

Les formulaires HTML vous permettent de spécifier différents types de validation pour vos champs de saisie afin de vous assurer que les données sont saisies correctement avant d'être soumises. HTML prend en charge un certain nombre de validateurs différents, y compris des éléments tels que la valeur minimale, la longueur minimale/maximale, etc. Les validateurs sont spécifiés en tant qu'attributs sur le `input` champ.

`required` Attribut

En HTML, les champs de saisie ont un attribut appelé `required` qui spécifie que le champ doit inclure une valeur.
L'exemple de bloc de code montre un champ de saisie obligatoire. L'attribut peut être écrit comme `required="true"` ou simplement `required` .

```
<input type="password"
name="password" required >
```

`min` Attribut

En HTML, les champs de saisie avec type `number` ont un attribut appelé `min` qui spécifie la valeur minimale pouvant être saisie dans le champ. Le bloc de code fourni affiche un champ numérique d'entrée qui accepte un nombre avec la valeur minimale 1.

```
<input type="number" name="rating"
min="1" max="10">
```

`max` Attribut

Les HTML `<input>` de type `number` ont un attribut appelé `max` qui spécifie la valeur maximale du champ de saisie.
Le bloc de code affiche un `input` champ numérique défini pour avoir une valeur maximale de `20` . Toute valeur supérieure à `20` marquera le champ de saisie comme ayant une erreur.

```
<input type="number" max="20">
```

`minlength` Attribut

En HTML, un champ de saisie de type `text` a un attribut qui prend en charge la validation de longueur minimale. Pour vérifier que le texte saisi a une longueur minimale, ajoutez l' `minlength` attribut avec le nombre de caractères.
L'exemple de bloc de code montre un exemple de champ de texte d'une longueur minimale de `6` .

```
<input type="text" name="username"
minlength="6" />
```

`maxlength` Attribut

En HTML, les champs de saisie avec type `text` ont un attribut appelé `maxlength` qui spécifie le nombre maximum de caractères pouvant être saisis dans le champ. Le bloc de code affiche un champ de saisie de texte qui accepte du texte d'une longueur maximale de 140 caractères.

```
<input type="text" name="tweet"
maxlength="140">
```

`pattern` Attribut

Dans un `text` élément d'entrée, l' `pattern` attribut utilise une expression régulière pour comparer (ou valider) la valeur de `<input>` , lorsque le formulaire est soumis.

```
<form action="/action_page.php">
  Country code:
  <input type="text"
name="country_code"
         pattern="[A-Za-z]{3}"
         title="Three letter country
code">
  <input type="submit">
</form>
```