

6) Les formulaires:

- * les formulaires HTML sont la principale source d'interaction d'un utilisateur avec un site web car ils permettent à l'utilisateur de fournir des données.
- * une formulaire est composé d'un ou de plusieurs widgets HTML.
- * les widgets permettent :
 - i) d'entrer des informations par l'intermédiaire de zone de texte
 - ii) de cases à cocher : checkbox
 - iii) de boutons radio : radio
 - iv) de liste sélectiionnable (select list).

i) L'élément <form>:

- l'élément form permet de déclarer un formulaire qui va contenir un ou plusieurs widgets.
- les attributs action et method :
 - * action est un attribut permettant de définir l'URL où devront être envoyées les données du formulaire
 - * method est un attribut permettant de définir la méthode HTTP utilisée pour envoyer ses données (post)

Ex -

```
<form action = "URL-server" method = "Post">  
</form>
```

ii) Les attributs communs à tous les widgets:

* autofocus:

cet attribut permet de sélectionner automatiquement le widget (dans lequel on écrit l'attribut autofocus) lors du chargement de la page.

* disabled:

cet attribut permet de désactiver le widget, l'utilisateur ne peut alors pas interagir avec le widget.

* name:

cet attribut permet de nommer le widget, le nom sera utilisé pour contenir les données du champ lors de l'envoie du formulaire.

* value:

cet attribut permet de donner une valeur initiale au widget (sauf pour textarea).

iii) les champs pour saisir du texte:

- * Il existe plusieurs types de widgets pour saisir du texte dans des formulaires: **input** et **textarea**.

- * les attributs communs aux champs de saisie de texte:

- * readonly:

L'utilisateur ne peut pas modifier la valeur du champ.

- * required:

indique que le champ est obligatoire avant de soumettre la formulaire (sauf pour certains types de champ input).

- * tabindex:

permet à l'utilisateur de naviguer avec tab entre ses champs.

- * placeholder:

permet de faire apparaître un texte indicatif dans le champ qui décrit ce qu'il doit contenir, le texte disparaîtra lorsque l'utilisateur commencera à taper.

v) le widget input:

* input type = "text":

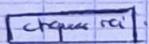
l'élément input permet de créer un champ permettant de saisir un texte sur une ligne :

```
<input type="text" name="Nom" placeholder="Nom" />
```

* input type = "button":

l'élément `<input>` de type button sont affichés comme des boutons possédant (□):

```
<input type="button" value="cliquez ici" />
```

= 

* input type = "checkbox":

l'élément input de type checkbox sont affichés sous la forme de boîtes à cocher ; Elles permettent de sélectionner un ou plusieurs valeurs dans un formulaire.

```
<input type="checkbox" name="---" value="---" /> Nom...
```

* input type = "color":

l'élément input de type color permettent de sélectionner une couleur via une interface où en saisissant le code hexadécimal de la couleur dans l'attribut value :

```
<input type="color" value="#e66465" />
```

* input type = "date" & type = "datetime-local": (type = "month" et (year, month, date))

l'élément input de type = "date" permettent de créer des champs où saisir des dates

```
<input type="date" />
```

mais `<input type="datetime-local" />` permet en plus d'afficher l'heure.

* input type = "email":

cet élément permet de faire utiliser de saisir un email (il exige l'utilisation de `@`)

- input type = "file":

L'élément input de type file permet à un utilisateur de sélectionner un ou plusieurs fichiers.

- input type = "image":

Les éléments input de type image sont utilisés pour créer des boutons d'envoi de formulaire graphique, Ex : Il s'agit de boutons d'envoi qui affichent une image plutôt qu'un texte.

- input type = "number":

permettent à une utilisateur de saisir des nombres dans une formulaire.

<input type = "number" min = "10" max = "100">

- input type = "password":

permettent à l'utilisateur de saisir un mot de passe sans que celui-ci ne soit pas visible à l'écran.

- input type = "radio":

Sont généralement utilisés pour construire des groupes d'options parmi lesquelles on ne peut choisir qu'une valeur, pour cela il faut que l'attribut name soit identique pour tous les éléments de type radio.)

Ex :

<input name = "sport" type = "radio" valeur = "foot"> Foot
<input name = "sport" type = "radio" valeur = "badminton"> Badminton

Ré, l'attribut checked permet de préselectionner un input de type radio ou checkbox.

<input type = "checkbox" checked> Value - 1

- input de type = "range":

- permet à l'utilisateur d'indiquer une valeur numérique comprise entre deux bornes
- Ces éléments sont représentés avec un curseur sur une ligne (----o----

- input de type = "reset":

Les éléments <input> de type "reset" sont affichés sous la forme de boutons permettant de réinitialiser l'ensemble des champs du formulaire avec leurs valeurs initiales.

- input de type = "search":

Les éléments <input> dont l'attribut type vaut search permettent de saisir des termes de recherche sur ce plan fonctionnel.

- input de type = "submit":

Sont affichés comme des boutons permettant d'envoyer des données d'un formulaire.

<input type = "submit" value = "Envoyer le formulaire">

- input de type = "tel":

• permettent de saisir un numéro de téléphone

•

<input type = "tel" pattern = "[0-9]{3}-[0-9]{3}-[0-9]{4}">

• l'attribut pattern permet de définir le format du N° de tel

- input de type = "time":

• permettent de créer des contrôles où l'utilisateur peut saisir une heure

•

<input type = "time" min = "09:00" max = "18:00" required>

- input type = "url":

permet de # utilisation de saisir un URL.

v) le widget textarea:

- L'élément textarea permet de définir un champ texte sur plusieurs lignes. Elle prend en compte les retours à la ligne entrés par l'utilisateur.
- On peut fixer le champ de sa zone de texte avec :
 - cols : fixe sa largeur
 - rows : fixe la hauteur
- pour donner une valeur initiale au textarea, il suffit de placer du texte entre des balises ouvrante et fermante.
Ex

```
<textarea name="longtext" cols="30" rows="10">Valeur Init</textarea>
```

vi) * les Libellés label

- L'élément <label> est une légende pour un champ
- Il peut être associé à un champ avec l'attribut for.
- pour associer le label à un champ avec for , il faut que le champ ait un id

Ex :

```
<label for="nom"> Nom </label>  
<input type="text" id="nom" />
```

* L'élément button:

- <button> Sauvegarde </button> # permet de créer un bouton
- <button type="submit"> Sauvegarder </button> # bouton pour transférer de formulaire.

vi) Les Listes Sélectionnables : Select

- * L'élément `Select` permet de créer une liste déroulante permettant de sélectionner une option parmi une liste d'options.

< select >

* Ex :

```
<label for="sexe"> Sexe </label>
<select id="sexe">
    <option value="man"> Homme </option>
    <option value="woman"> Femme </option>
</select>
```

- * Les éléments "option" permettent de définir ses options sélectionnables dans sa liste.

- si l'attribut `value` est défini sur un élément `option`, alors ce sera cette valeur qui est envoyée si l'option est sélectionnée.
- si il n'est pas défini, ce sera le texte contenu entre les balises `<option>`.

- * L'élément `optgroup` permet de définir un label qui est affiché pour un group d'options, cet élément n'est pas sélectionnable

- * L'attribut `selected` défini sur une option permet de définir l'option sélectionnée par défaut

Ex :

```
<select>
    <optgroup label="sidi">
        <option> livre_1 </option>
        <option selected> livre_2 </option>
    </optgroup>
</select>
```

- * L'attribut `multiple`:

Il est possible d'utiliser l'attribut `multiple` sur `select` afin de rendre plusieurs options sélectionnables en utilisant : ctrl ou maj.

Vii) L'élément `fieldset`:

- L'élément HTML `<fieldset>` permet de regrouper plusieurs champs en les encadrent, il permet aussi de définir une `<legend>` décrivant le group

Ex :

```
<fieldset>
    <legend> Sélectionnez vos sports favoris </legend>
    <label for="sport"> Sport </label><br>
    <input name="sport" type="radio" id="sport"> Football<br>
    <input name="sport" type="radio" id="sport"> Badminton<br>
</fieldset>
```

Viii) Compteur et barres de progression : "meter et progress".

* L'élément `progress`:

- une barre de progression, définir avec `<progress>` représente une valeur qui évolue dans le temps jusqu'à une valeur maximale définie par l'attribut `max`
- cela permet par exemple d'afficher l'avancement d'un téléchargement

Ex :

```
<fieldset>
    <progress max="100" value="33"> 33/100 </progress>
</fieldset>
```

- le contenu entre les deux balises est affiché si il y'a un erreur (comme pour `alt`)

* L'élément `meter`:

- Un étalon, défini avec `<meter>` est une valeur dans une plage délimitée par une valeur minimale `min` et une valeur maximale.
- L'étalonnage est divisé en trois parties: `min` et `low`, `low` et `high` et entre `high` et `max`.
- `optimum`: permet de définir la valeur optimale pour l'étalonnage.
`<meter min="0" max="100" low="30" high="80" value="30" optimum="100"> 30 </meter>`