

Programmation Web

CM 2 : Forms/Formulaires

Mussab Zneika
CY-Tech
2022-2023

LES FORMULAIRES

Introduction aux formulaires

- Traditionnellement, "formulaire" fait référence à un document imprimé qui contient des espaces pour remplir certaines informations.
- HTML emprunte le concept de formulaire pour faire référence à différents éléments qui vous permettent de collecter des informations auprès des visiteurs de nos sites.
- Les formulaires sont partout autour de nous (login, formulaire de contact et sign up, etc...).



amazon.fr

Ouvrir une session

Quelle est votre adresse e-mail ?

Votre adresse e-mail est:

Disposez-vous d'un mot de passe Amazon.fr ?

☐ Vous êtes un nouveau client.
(vous créerez un mot de passe plus tard)

☒ Vous êtes déjà client Amazon ?
Votre mot de passe est :

☐ Maintenir ma session ouverte. [Détails](#)

[Continuer \(serveur sécurisé\)](#)

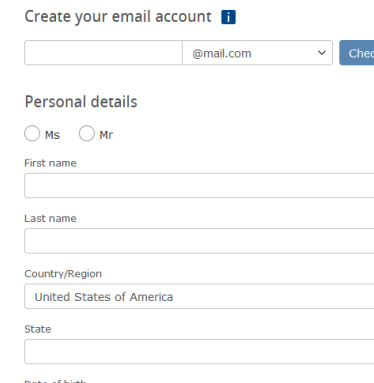
[Vous avez oublié votre mot de passe ?](#)

[Vous avez changé d'adresse e-mail depuis votre dernière commande ?](#)



Illustration of a person sitting at a desk with a computer, with the Google logo and binary code in the background.

Recherche Google J'ai de la chance



Create your email account [?](#)

@mail.com

Personal details

☐ Ms ☐ Mr

First name

Last name

Country/Region

State

Date of birth

LES FORMULAIRES

Introduction aux formulaires

- Le formulaire est utilisé pour envoyer les données que l'utilisateur saisira au serveur qui les traitera.
- Comment fonctionnent les formulaires :
 1. Un utilisateur remplit un formulaire puis il appuie sur un bouton de soumission pour soumettre les informations au serveur.
 2. Le nom de chaque champ de formulaire est envoyé au serveur avec la valeur saisie ou sélectionnée par l'utilisateur.
 3. Le serveur traite les informations à l'aide d'un langage de programmation tel que PHP, ASP.net, JSP ou Python.
- Les actions et les champs que vous autorisez dans votre formulaire déterminent ce que l'utilisateur est autorisé à faire et quelles informations il est autorisé à voir.

LES FORMULAIRES

La balise Formulaire <Form>

- Pour créer une section de formulaire, nous fournissons au formulaire un nom, un identifiant, une action et une méthode. Un exemple avec des attributs vides ressemble à ceci :

```
<form action="" method=""> </form>
```

- Le nom et l'identifiant de notre formulaire sont la façon dont nous nous y référons dans notre code lors de l'interaction avec lui en utilisant CSS, JavaScript ou d'autres langages.
- L'action est l'endroit où la page doit envoyer les informations (et où le navigateur ira lorsque nous appuierons sur envoyer).
- Notre méthode est la façon dont nous enverrons nos informations, en utilisant GET ou POST.

LES FORMULAIRES

Les attributs de formulaire :

Attribut:	Description
action	Sa valeur est l'URL spécifiant où envoyer les données lorsque le bouton Soumettre est cliqué.
method	Les formulaires peuvent être envoyés en utilisant l'un des deux méthodes : <ul style="list-style-type: none">→ "get" : informations passées dans la barre d'adresse→ "post" : envoie les données dans le corps de la requête
target	Indique où ouvrir la page envoyée à la suite de la demande <ul style="list-style-type: none">→ target = _blank signifie ouvrir dans une nouvelle fenêtre→ target = _top signifie utiliser la même fenêtre
id	la valeur est utilisée pour identifier la forme distinctement de autres éléments de la page
name	

LES FORMULAIRES

Method (Get)

- L'envoi des données à l'aide de la méthode get place tous les champs du formulaire par nom et valeur (appelée paire clé et valeur) dans la barre d'adresse, ce qui allonge notre URI en ajoutant chaque élément à l'adresse de la page de réception.

➤ <https://www.google.com/search?q=cergy+université>

- L'avantage d'utiliser la méthode get est que Les paramètres de l'URL peuvent être enregistrés avec l'adresse du site Web. Cela permet de mettre une requête de recherche en marque-page et de la récupérer plus tard.
- Bien que bénéfique, il existe deux cas dans lesquels nous ne voulons PAS utiliser get : soit nous ne voulons pas que l'utilisateur (ou quiconque) voie ce qui a été envoyé, comme des mots de passe ou des informations confidentielles, soit nous voulons envoyer beaucoup d'informations .

LES FORMULAIRES

Method (Post)

- Les données publiées sont envoyées du navigateur au serveur en arrière-plan, lorsque le client et le serveur commencent à parler.
- Les données sont envoyées dans les en-têtes (voir) de la communication, et ne sont pas visibles pour l'utilisateur final.
- Les pages mises en signet avec la méthode de publication n'auront plus accès aux informations par la suite, et ces informations sont perdues si l'utilisateur quitte la page.
- La manière dont les données sont utilisées et les valeurs ou le nouveau contenu renvoyés nous amènent aux langages de script.

GET VS POST

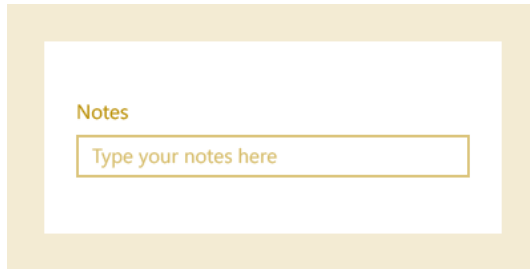
	GET	POST
Visibilité	Visible pour l'utilisateur dans le champ d'adresse	Invisible pour l'utilisateur
Marque-page et historique de navigation	Les paramètres de l'URL sont stockés en même temps que l'URL.	L'URL est enregistrée sans paramètres URL.
Cache et fichier log du serveur	Les paramètres de l'URL sont stockés sans chiffrement	Les paramètres de l'URL ne sont pas enregistrés automatiquement.
Comportement lors de l'actualisation du navigateur / Bouton « précédent »	Les paramètres de l'URL ne sont pas envoyés à nouveau.	Le navigateur avertit que les données du formulaire doivent être renvoyées.
Type de données	Caractères ASCII uniquement.	Caractères ASCII mais également données binaires.
Longueur des données	Limitée - longueur maximale de l'URL à 2 048 caractères.	Illimitée

LES FORMULAIRES

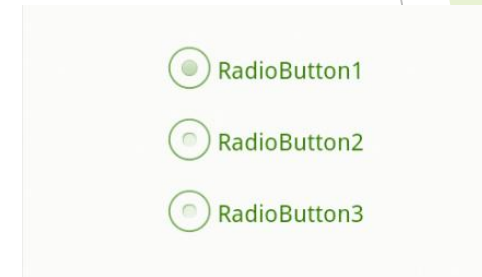
Types de contrôles de formulaire

- Il existe plusieurs types de contrôles de formulaire

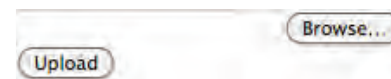
Text Box

A screenshot of a web form with a light orange border. Inside, the word "Notes" is written in orange. Below it is a text input field with the placeholder text "Type your notes here" in orange.

Radio Buttons:

A screenshot of three radio buttons stacked vertically. Each button consists of a small green circle followed by the text "RadioButton1", "RadioButton2", and "RadioButton3" respectively.

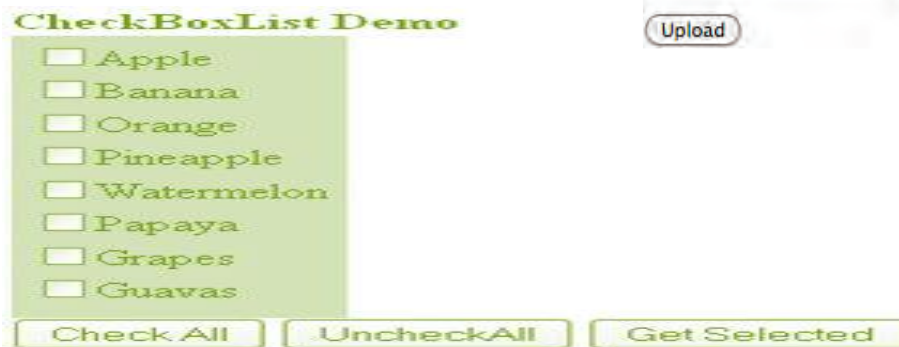
File upload

A screenshot of a file upload control. It features a text input field, a button labeled "Browse..." to its right, and a button labeled "Upload" below the input field.

Password

A screenshot of a password input field, which is a rectangular box with a black border and ten black dots inside to mask the text.

Check Boxes

A screenshot of a web form titled "CheckBoxList Demo". It contains a list of seven items, each with an unchecked checkbox and a label: "Apple", "Banana", "Orange", "Pineapple", "Watermelon", "Papaya", and "Guavas". Below the list are three buttons: "Check All", "UncheckAll", and "Get Selected".

Submit buttons

A screenshot of a single submit button with a green gradient background and the text "SUBSCRIBE" in white capital letters.

LES FORMULAIRES

la balise <input>

- > la balise **<input>** est utilisée pour créer plusieurs formulaires différents.
- > Le **type** d'attribut indique de quel type d'élément il s'agit
- > la valeur de l'attribut type peut être:
 - text
 - checkbox
 - radio
 - password
 - hidden
 - Submit
 - reset
 - Button
 - file
 - image

LES FORMULAIRES

la balise <input>

- la balise **<input>** est utilisée pour créer plusieurs formulaires différents.
- Le **type** d'attribut indique de quel type d'élément il s'agit
- la valeur de l'attribut type peut être:
 - Text, checkbox, radio, password, hidden, Submit, reset, Button, file, image
- Attributs communs de la balise <input> :
 - **name** : le nom de l'élément, chaque <input> élément doit avoir un nom pour que l'on peut récupérer sa valeur . Ce nom n'est pas visible sur la page.
 - **value** : la valeur de l'élément ; utilisé de différentes manières pour différents valeurs de type.
 - **readonly** : la valeur ne peut pas être modifiée.
 - **disabled** : l'utilisateur ne peut rien faire avec cet élément

LES FORMULAIRES

➤ Zone de texte(Textbox)

- Lorsque l'attribut type a du **text** comme valeur , le navigateur crée une zone de texte sur une seule ligne.

```
<input type="text" name="text1" value="enter your name" />
```

- La balise input est auto-fermante.
- l'attribut **maxlength** définit le nombre de caractères maximum acceptés dans le champ. Par exemple, si nous demandions un code postal en France, l'attribut maxlength pourrait avoir une valeur de 5.

LES FORMULAIRES

Le Label

- chaque contrôle de formulaire doit avoir son propre élément `<label>` qui décrit aux utilisateurs la fonctionnalité du champ.
- Il peut être utilisé de deux manières :
 - Enroulez autour de la description textuelle et de l'élément de

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

- Séparé du contrôle de formulaire et utilisez le **for** pour indiquer à quel

```
<label for="lname">Last name: </label>  
  
<input type="text" id="lname"/>
```

LES FORMULAIRES

➤ Premier exemple :

```
<form action= "proce.php" method= "post">
  <label for="fname">First name:</label>
  <input type="text" id= "fname" name="fname"/>
  <label for="lname">Last name:</label>
  <input type="text" id="lname" name="lname"/>
  <label for="age">Age:</label>
  <input type="text" id="age" name="age"/>
</form>
```

First name:

Last name:

Age:

LES FORMULAIRES

Zone de texte multiligne <textarea>

- L'élément **<textarea>** est utilisé pour créer une zone de texte multiligne .
- Une balise qu'il faut ouvrir et fermer.
- Pas de value= " ", le texte par défaut se met entre les deux balises ouvrante et fermante.
- L'attribut **cols** indique la largeur de la zone de texte.
- L'attribut **rows** indique combien de lignes la zone de texte doit occuper verticalement.

```
<textarea name="comments" cols="50" rows="5">  
Enter your comments...  
</textarea>
```

LES FORMULAIRES

Champ de mot de passe(Password input)

- la balise <input> avec une valeur de type "**password**" est utilisée pour créer une zone de texte dont ses caractères sont masqués.
- Même si le mot de passe est masqué à l'écran, les données d'un champ "mot de passe" ne sont pas nécessairement envoyées de manière sécurisée au serveur.
- Il peut également avoir les attributs **size** et **maxlength**

```
<input type="password" name="password" size="20" maxlength="15" />
```


LES FORMULAIRES

Les boutons radio <input type="radio">

- Les boutons radio permettent aux utilisateurs de sélectionner une seule option parmi plusieurs.
- La valeur est importante : c'est elle qui est renvoyée au serveur.
- La valeur de chacun des boutons dans un groupe devrait être différent .
- Les boutons radio d'un groupe doivent avoir le même nom.
- L'attribut "**checked**" est utilisé pour indiquer quelle valeur doit

```
<p>Please select your favorite Drink: <br />
  <input type="radio" name="drink" value="tea" checked="checked" />
  Tea
  <input type="radio" name="drink" value="coffee" /> coffee
  <input type="radio" name="drink" value="milk" /> Milk
</p>
```

Please select your favorite Drink:

☒ Tea ☐ coffee ☐ Milk

LES FORMULAIRES

Les cases à cocher (checkbox)

- `<input type="checkbox">`
- Les cases à cocher permettent aux utilisateurs de sélectionner(et désélectionnez) un ou plusieurs options.
- L'attribut `value` indique la valeur envoyée au serveur si la case est cochée.
- L'attribut "**checked**" est utilisé pour indiquer quelle valeur doit être

```
<p>Please select your favorite social media applications: <br />
<input type="checkbox" name="media" value="facebook" /> Facebook
<input type="checkbox" name="media" value="twitter" checked="checked" />
twitter
<input type="checkbox" name="media" value="whatsapp" /> whatsapp
</p>
```

Please select your favorite social media applications:

☐ Facebook ☒ twitter ☐ whatsapp

LES FORMULAIRES

➤ La liste déroulante(Drop Down List)

- La balise **<select>** est utilisée pour créer une liste déroulante.
- La balise `<select>` contient au moins deux **<option>** éléments.
- La balise **<option>** est utilisée pour spécifier les options que le l'utilisateur peut choisir.
- l'attribut **value** de l'élément `<option>` indique le valeur qui est envoyée au serveur.
- L'attribut **"selected"** est utilisé pour indiquer l'option qui doit être sélectionné lors du chargement de la page.

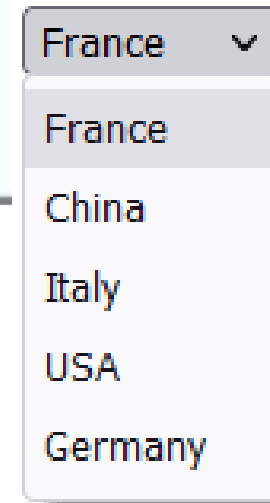
LES FORMULAIRES

La liste déroulante(Drop Down List):Exemple

```
<form action="page1.php">
  <p>Select you country?</p>
  <select name="country">
    <option value="France">France</option>
    <option value="China">China</option>
    <option value="Italy">Italy</option>
    <option value="Italy">USA</option>
    <option value="Germany">Germany</option>
  </select>
</form>
```

Si l'utilisateur ne sélectionne pas d'option, alors la première option (France dans notre exemple) sera envoyée à le serveur comme valeur pour ce contrôler.

Select your country



France ▼

France

China

Italy

USA

Germany

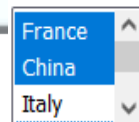
LES FORMULAIRES

La liste déroulante(Drop Down List)

- les attributs « **size** » et « **multiple** » de la balise select peut transformer une liste déroulante en une liste d'options multiples qui affiche plusieurs options à la fois

```
<form action="page1.php">
  <p> You can select more than one option by holding down control.</p>
  <select name="vistedcountries" size="3" multiple="multiple">
    <option value="France">France</option>
    <option value="China">China</option>
    <option value="Italy">Italy</option>
    <option value="Italy">USA</option>
    <option value="Germany">Germany</option>
  </select>
</form>
```

You can select more than one option by holding down control on a PC or command key on a Mac while selecting different options.)



LES FORMULAIRES

La liste déroulante(Drop Down List)

- La balise **<optgroup>**, utilisé dans un formulaire, permet de créer un groupe d'options parmi lesquelles on peut choisir dans un élément `<select>`

```
<select name="city" id="city">
  <optgroup label="France">
    <option value="Paris">Paris</option>
    <option value="Lyon">Lyon</option>
    <option value="Marseille">Marseille</option>
  </optgroup>
  <optgroup label="German">
    <option value="Berlin">Berlin</option>
    <option value="Hamburg">Hamburg</option>
  </optgroup>
</select>
```

Choose city:

France
Paris
Lyon
Marseille
German
Berlin
Hamburg

LES FORMULAIRES

Zone d'entrée de fichier <input type="file">.

- La zone de saisie de fichier aide les utilisateurs à uploader leurs fichiers vers le serveur .
- Utiliser les attributs suivants:

Attribut	Description
accept	nous permet de définir les types de fichier qui doivent être acceptés.
files	Un objet Filelist qui liste les fichiers choisis
capture	La source à utiliser pour capturer des images ou des vidéos
multiple	Un attribut booléen qui, lorsqu'il est présent, indique que plusieurs fichiers peuvent être sélectionnés.

LES FORMULAIRES

Zone d'entrée de fichier <input type="file">(Exemples):

```
<!--we can select only one file with all extensions -->  
<input type="file" id="myfile" name="myfile"/>
```

```
<!--we can select multiple files with all extensions -->  
<input type="file" id="myfile" name="myfile" multiple />
```

```
<!--we can select multiple files with word and pdf formats -->  
<input type="file" id="myfile" name="myfile" multiple  
accept=".doc,.docx,.pdf"/>
```

```
<!--we can select multiple files of all the formats of image,  
audio and video -->  
<input type="file" id="myfile" name="myfile" multiple  
accept="image/*, audio/*, video/*"/>
```


LES FORMULAIRES

Bouton d'envoi(Submit Button)

- Le bouton d'envoi `<input type="submit" >` est utilisé pour envoyer les informations saisies dans les éléments de contrôle du formulaire au serveur.
- L'attribut **value** est obligatoire et permet de donner le « titre » du bouton.
- les boutons de soumission peuvent être affichés dans différents manières basées sur les navigateurs.
- CSS peut être utilisé pour contrôler l'apparence des boutons de soumi.

```
<input type="submit" value="send">
```

LES FORMULAIRES

Bouton de reset(Reset Button)

- Le bouton de reset `<input type= " reset" >` est utilisé pour remettre à zéro toutes les informations saisies dans les éléments de contrôle du formulaire.
- L'attribut `value` permet de donner le « titre » du bouton.
- les boutons de reset peuvent être affichés dans différents manières basées sur les navigateurs.
- CSS peut être utilisé pour contrôler l'apparence des boutons de reset.

```
<input type="reset" value="send">
```

LES FORMULAIRES

Bouton image (Image Button)

- > Le **<input type= " image" >** est utilisé pour ajouter une image pour le bouton d'envoi .
- > nous pouvons utiliser "**src, width, height, et alt**" attribues tels que nous les utilisons avec les Balise

```
First name:<input type="text" id= "fname" name="fname"/><br>
Last name:<input type="text" id= "lname" name="lname"/><br>

<input type="image" src="submitb.jpg" width="100" height="50">
```

First name:

Last name:



LES FORMULAIRES

Bouton

- Un bouton est similaire au bouton d'envoi, mais contrairement aux autres styles de saisie, il peut inclure du texte ou une image.
- Sa mise en forme par défaut lui donne un aspect de bouton biseauté.
- Trois types : **button**, **reset**, **submit**

```
<button type="button">Click Here!</button><br>
```

```
<button type="submit"> </button> <br>
```

```
<button type="reset">Reset all!</button><br>
```

Click Here!

Submit

Reset all!

LES FORMULAIRES

Champ caché(Hidden input)

- On peut cacher un champ avec **<input type="hidden">**.
- Les données dans les contrôles cachés ne sont pas affichées sur le page mais nous pouvons les voir si nous utilisons l'option « Afficher la source ».
- les données sont dans l'attribut value

```
<input type="hidden" name="currentpage" value="orderpage" >
```

LES FORMULAIRES

Champs de forme HTML5

- Bon nombre des nouveaux éléments HTML5 que nous examinons nous aideront également dans nos tâches de validation lorsque les utilisateurs remplissent des formulaires.
- Ces entrées tenteront de valider et/ou de limiter l'entrée de l'utilisateur aux seules données valides. En faisant cela immédiatement, nous créons une meilleure expérience à la fois pour l'utilisateur et le programmeur.
- Traditionnellement, la validation devait être effectuée lors de l'envoi des données au serveur, ce qui entraînait le rechargement de la page en cas d'erreurs.
- L'autre approche populaire consistait à effectuer une validation à l'aide de JavaScript côté client (en évitant le rechargement), mais la validation devrait toujours être répétée sur le serveur au cas où l'utilisateur final aurait désactivé JavaScript.
- Certains des types d'entrée les plus utiles sont les suivants :
- **<input type="url">** Tentera de formater le texte de l'utilisateur dans un lien approprié, ou affichera une erreur. **<input type="email">** S'assurera qu'un e-mail saisi est dans le bon format ou affichera une erreur.

LES FORMULAIRES

Champs de forme HTML5

- **<input type="email">** S'assurera qu'un e-mail saisi est dans le bon format ou affichera une erreur.

```
email: <input type="email" name="email">
```

First name:

Last name:

email:

submit

Veuillez saisir une adresse électronique valide.

- Nous pouvons également créer une entrée qui limite les valeurs à une plage fixe et incrémente les limites, que nous devons auparavant afficher à l'utilisateur sur la page, puis valider après la saisie :

```
age: <input type="range" name="age"
min="18" max="62" value="30" step="1">
```

age:

- Ces limites sur une plage (affichées sous forme de curseur) sont

```
age: <input type="number" name="age"
min="18" max="62" value="30" step="1">
```

affiché age:

LES FORMULAIRES

Champs de forme HTML5

- HTML5 introduit également une multitude de contrôles de calendrier et d'heure.
- Nous pouvons spécifier une date, une semaine ou un mois ainsi qu'une heure, un jour et une heure et un jour et une heure locaux.
- Chacun de ces champs limitera l'entrée de l'utilisateur aux champs valides pour ce type.

Birthda: `<input type="date" name="date" value="2021-11-08">`

Birthda:

<

novembre 2021

>

lun.	mar.	mer.	jeu.	ven.	sam.	dim.
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

On peut utiliser les attributs **min** et **max** afin de restreindre les dates qui peuvent être saisies par l'utilisateur. où la date que nous pouvons sélectionner doit être comprise entre la date en valeur min et la date en valeur max

LES FORMULAIRES

Champs de forme HTML5

> Options de calendrier :

- > Le **<input type="month">** crée un contrôle de mois et d'année où l'utilisateur peut entrer le format de date comme "AAAA-MM".
- > On peut utiliser les attributs **min** et **max** afin de restreindre les dates qui peuvent être saisies par l'utilisateur.

```
<input type="month" name="month" value="2021-07" min="2010-08" max="2021-08">
```

Start month: 2021 



2021

janv.	févr.	mars	avr.
mai	juin	juil.	août
sept.	oct.	nov.	déc.

Effacer

Mois en cours


Ce champ n'est implémenté que sur Chrome, Opera et Edge sur les ordinateurs. Pour les autres navigateurs qui ne supportent pas ce champ, le champ sera transformé en une simple Zone de Texte

LES FORMULAIRES



Champs de forme HTML5

- Options de calendrier :
 - Le **<input type="week">** définit un contrôle de semaine et d'année.
 - l'apparence de ce champ est différente selon les

```
Start Week: <input type="week" name="week" value="2021-W12" min="2010-W11" max="2021-W44">
```

Start Week: Semaine **12**, 2021 

Submit

mars 2021 ▾  

Semaine	lu	ma	me	je	ve	sa	di
9	1	2	3	4	5	6	7
10	8	9	10	11	12	13	14
11	15	16	17	18	19	20	21
12	22	23	24	25	26	27	28
13	29	30	31	1	2	3	4
14	5	6	7	8	9	10	11

[Effacer](#) [Cette semaine](#)

LES FORMULAIRES

Champs de forme HTML5

- Options de temps :

- l'apparence de ces champ est différente selon les navigateurs

Time: `<input type="time" name="time" value="14:30">`

Time: 14:30

Submi

14	30
15	31
16	32
17	33
18	34
19	35
20	36

Time `<input type="datetime-local" name="time2">`

Time: jj/mm/aaaa --:--

Submi

novembre 2021

lu	ma	me	je	ve	sa	di	22	26
1	2	3	4	5	6	7	23	27
8	9	10	11	12	13	14	00	28
15	16	17	18	19	20	21	01	29
22	23	24	25	26	27	28	02	30
29	30	1	2	3	4	5	03	31
6	7	8	9	10	11	12	04	32

Effacer Aujourd'hui

LES FORMULAIRES

Champs de forme HTML5

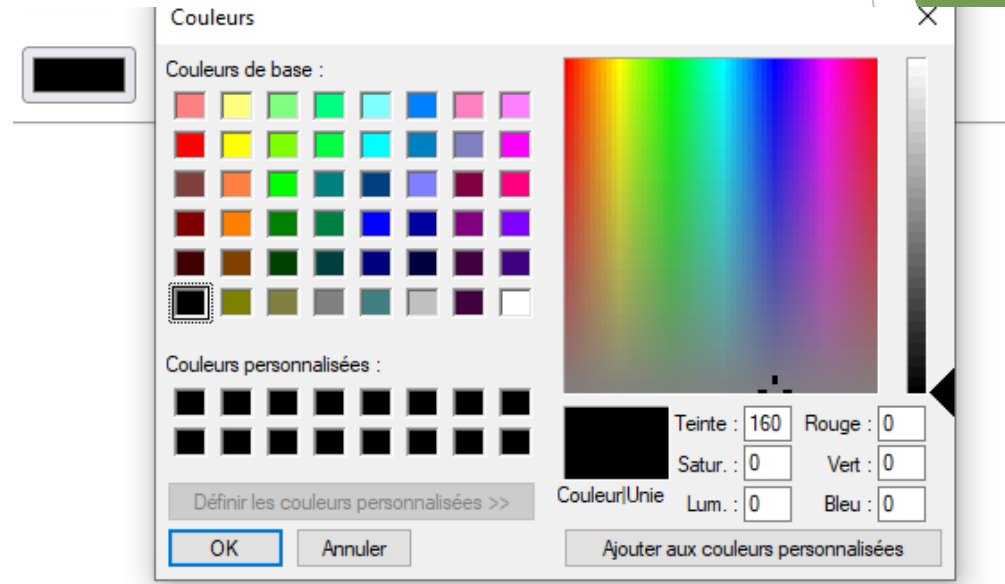
- Plus des Champs:

- `<input type="color">`

- `<input type="tel">`

- `<input type="search">`

- `<input type="url" >`



LES FORMULAIRES

Balises de jeu de champs et de légende

- La balise **<fieldset>** est utilisée pour regrouper les contrôles liés dans un formulaire qui dessine un cadre autour des contrôles du formulaire lié.
- L'attribut **form="form_id"** Spécifie à quelle forme le "fieldset" appartient.
- La balise **<legend>** représente une légende pour le contenu de son élément parent **<fieldset>**.
- l'attribut "disabled" spécifie qu'un groupe d'éléments de formulaire liés doit être désactivé

LES FORMULAIRES

Balises de jeu de champs et de légende: Example:

```
<form action="page1.php">
  <fieldset>
    <legend>Your information:</legend>
    <label for="fname">First name:</label>
    <input type="text" id="fname" name="fname"><br><br>
    <label for="lname">Last name:</label>
    <input type="text" id="lname" name="lname"><br><br>
    <input type="image" src="submitb.jpg" width="50" height="25" >
  </fieldset>
</form>
```

Your information:

First name:

Last name:

A rectangular button with a red border and the word "Submit" in a red, serif font.

SOURCES D' INFORMATION

- *MDN WEB DOCS: WEB FORMS — WORKING WITH USER DATA:*
<https://developer.mozilla.org/en-US/docs/Learn/Forms>
- W3C HTML Forms: https://www.w3schools.com/html/html_forms.asp
- *MENDEZ, MICHAEL. THE MISSING LINK: AN INTRODUCTION TO WEB DEVELOPMENT AND PROGRAMMING. 2014, CH14.*

SOURCES D'INFORMATIONS

- https://www.w3schools.com/html/html_tables.asp
- <https://developer.mozilla.org/fr/docs/Learn/HTML/Tables>
- MENDEZ, MICHAEL. *THE MISSING LINK: AN INTRODUCTION TO WEB DEVELOPMENT AND PROGRAMMING*. 2014, CH13
- MENDEZ, MICHAEL. *THE MISSING LINK: AN INTRODUCTION TO WEB DEVELOPMENT AND PROGRAMMING*. 2014, CH11,12
- MENDEZ, MICHAEL. *THE MISSING LINK: AN INTRODUCTION TO WEB DEVELOPMENT AND PROGRAMMING*. 2014, CH9,10