

Les formulaires

Table des matières



Introduction	3
I - Création de formulaire	4
1. Les éléments constituant un formulaire	4
2. Comment bien créer un formulaire ?	5
3. Exemple de création de formulaire	5
II - Exercice : Activité d'auto-évaluation 1	9
III - Traitement des formulaires en PHP	10
1. Récupération du contenu d'un formulaire avec la méthode get et request	10
2. Récupération du contenu d'un formulaire avec la méthode post	11
3. Travaux dirigés	12
IV - Exercice : Activité d'auto-évaluation 2	14
V - Validation d'un formulaire	16
1. La validation par défaut	16
2. La validation en JavaScript ou JQuery	18
3. La validation par bootstrap	20
4. La validation par plugin JQuery ou Javascript	25
VI - Exercice : Activité d'auto-évaluation 3	28
VII - Envoi et traitement des fichiers	30
1. Envoi d'un fichier et récupération d'un fichier	30
2. travaux dirigés	31
VIII - Exercice : Activité d'auto-évaluation 4	39

Introduction



Les formulaires permettent à l'utilisateur de saisir et renseigner des informations, puis de les transmettre au serveur par le biais d'un navigateur. Un formulaire se fait en HTML grâce à la balise `<form>` `</form>`.

Pour savoir comment construire un formulaire, veuillez vous référer au cours de HTML et CSS



Création de formulaire



1. Les éléments constituant un formulaire

Pour créer un formulaire on définit d'abord une balise `<form></form>`. A cette balise, on peut ajouter plusieurs éléments HTML en fonction de ce que l'on veut faire. Voici ci-dessous quelques éléments :

- la balise `<input />` permet de saisir ou d'ajouter des informations dans un formulaire. Il y a plusieurs types de balises input en fonction de la nature de l'information. Une balise input de :
 - type *text* : permet de saisir une information textuelle sur une ligne (une information pas assez volumineuse tel que le nom, le prénom, l'adresse, le numéro de carte, etc.). Plusieurs informations textuelles peuvent être saisies avec cette balise tel que l'email, la date, les chiffres, etc. Toute fois, pour une grande précision sur la nature des informations, et pour permettre aux navigateurs de mieux savoir sur la nature de l'information et offrir un meilleur affichage de saisie, il est conseillé d'utiliser des types plus précis nous allons énumérer ci-dessous.
 - type *email* : permet de renseigner une adresse email. L'avantage d'utiliser un type email au lieu d'un type text, est que le type email permet au navigateur de savoir que l'information attendue est une adresse email. Ainsi, le navigateur va vérifier si l'information saisie est au format d'une adresse email, si tel n'est pas le cas, il va obliger l'utilisateur à renseigner une adresse email correct.
 - type *tel* : permet de renseigner un numéro de téléphone
 - type *number* : permet de renseigner un nombre
 - type *password* : permet de renseigner un mot de passe
 - type *date* permet de renseigner une date
 - type *time* permet de renseigner l'heure
 - type *file* permet de joindre un fichier (image, pdf, word, video, ect.)
 - type *checkbox* permet de créer une liste de cases à cocher à choix multiple
 - type *radio* permet de créer une liste de cases à cocher à choix unique
 - type *submit* pour valider le formulaire
 - etc.
- la balise `<select> </select>` permet de créer une liste de sélection. Cette balise est très importante si nous voulons créer une liste d'éléments à choisir, par exemple la ville de l'utilisateur. Elle évite que l'utilisateur saisisse lui même l'élément et fasse des erreurs de saisie.
- la balise `<textarea> </textarea>` permet de saisir une information multi-lignes. Elle est très importante lorsque nous voulons saisir une information volumineuse
- la balise `<label> </label>` pour les textes à afficher à côté de chaque champ de saisie.

NB : La balise `<input />` est beaucoup utilisées dans la création d'un formulaire, pour préciser le type de balise input, on utilise l'attribut *type*

2. Comment bien créer un formulaire ?

Pour bien créer un formulaire, il faut :

- Savoir bien utiliser les types de balises appropriés en fonction de la nature de l'information à saisir ou à renseigner comme vu ci-dessus.
- Attribuer à chaque balise `<input />`, `<select></select>`, ou `<textarea> </textarea>` un *attribut name unique* (à l'exception près des cases à cocher) dans le formulaire. L'attribut *name* permet d'identifier chaque information à renseigner et c'est grâce à cet identifiant que nous allons récupérer les informations sur le serveur en PHP.
NB : Il est obligatoire de donner un *name* à chaque champ en renseigner pour pouvoir récupérer l'information sur le serveur
- Définir un texte à afficher à côté de chaque champ de saisie pour que l'utilisateur puisse savoir l'information à saisir. Très souvent, on utilise pour cela la balise `<label> </label>`
- définir l'attribut *required* pour tous les champs obligatoires. Pour les champs obligatoires, il est conseillé de mettre une étoile à la fin du texte qui est à côté.
- définir un texte à afficher par défaut pour chaque champ grâce à l'attribut *placeholder*. Cette information n'est pas obligatoire, mais permet aussi d'apporter des renseignements sur l'information à saisir.
- Bien disposer les balises dans le formulaire, pour cela, il est conseillé d'aligner les balises dans un tableau.
- définir la méthode d'envoi des informations sur le serveur grâce à l'attribut *method* de la balise `<form> </form>`. Il existe deux méthodes, la méthode *get* et la méthode *post* que nous allons bien expliquer un peu plus bas.
- définir la page PHP qui doit traiter le formulaire grâce à l'attribut *action* de la balise `<form> </form>`. On verra mieux comment ça se passe un peu plus bas
- Avant d'envoyer le formulaire sur le serveur vérifier que les informations ont été correctement saisies. Pour faire cette vérification on passe par le JavaScript.

NB : Il est conseillé pour une meilleure disposition et une meilleure qualité d'affichage, de construire les formulaires grâce au *framework css bootstrap*

3. Exemple de création de formulaire

Libellé

Nous allons créer un formulaire permettant à un utilisateur de s'inscrire sur un site. Pour l'inscription, l'utilisateur doit renseigner les informations suivantes :

- nom (nom+prénom)
- email
- tel
- date de naissance
- ville de résidence
- sexe
- matricule
- mot de passe

le nom, le tel et le matricule sont obligatoires

Correction

- Nous utiliserons un tableau pour bien disposer les éléments
- Nous opterons pour la méthode d'envoi post
- nous appellerons la page contenant le formulaire `formulaire.php` (pour cela il faut créer une page php avec votre éditeur de code préféré et l'enregistrer sous le nom de `formulaire.php`)
- Nous appellerons la page qui va traiter le formulaire `inscription.php` (pour cela il faut créer une page php avec votre éditeur de code préféré et l'enregistrer sous le nom de `inscription.php`)

Ci-dessous le contenu de la page `formulaire.php`. Nous verrons le contenu de la page `inscription.php` dans la suite de la leçon.

Si le code ne s'affiche pas correctement en format web. Veuillez cliquer sur le lien en bleu

```

1 <!DOCTYPE HTML>
2 <html>
3 <head>
4   <meta http-equiv="content-type" content="text/html" />
5   <title>Inscription</title>
6   <style type="text/css">
7     form {
8       padding: 10px;
9       background-color: rgb(245,245,245);
10      padding: 10px;
11      border-radius: 7px 7px 0px 0px;
12      display: inline-block;
13      box-shadow: 0px 2px 6px rgba(32,32,32,0.5);
14    }
15    form table td {
16      padding: 5px 10px;
17    }
18    input[type="text"],input[type="email"],input[type="password"],input[type=
"tel"],input[type="date"],select {
19      width: 300px;
20      height: 30px;
21      border-radius: 7px;
22      border: none;
23      padding-left: 5px;
24    }
25    input[type="reset"],input[type="submit"]{
26      background-color: rgb(0,0,150);
27      color: white;
28      border: none;
29      border-radius: 7px;
30      padding: 5px 10px;
31      transition: all 0.5s;
32    }
33    input[type="reset"]{
34      background-color: red;
35    }
36    input[type="reset"]:hover,input[type="submit"]:hover{
37      background-color: rgb(255,128,0);
38      padding: 5px 25px;
39      border-radius: 10px;
40      cursor: pointer;
41    }
42  </style>

```

```


43 </head>
44 <body>
45     <h1 style="text-decoration: underline;"><center>Formulaire d'inscription</
46     <center>
47     <form method="post" action="inscription.php">
48         <table>
49             <tr>
50                 <td>Nom* : </td> <td><input type="text" name="nom" required=""
51             /></td>
52             <tr>
53                 <td>Email : </td> <td><input type="email" name="email" /></td>
54             <tr>
55             <tr>
56                 <td>Tel* : </td> <td><input type="tel" name="tel" required="" /></
57             <td>
58             <tr>
59                 <td>Date de naissance : </td> <td><input type="date" name=
60                 "date_naissance" placeholder="Ex: 20/10/2000" /></td>
61             <tr>
62                 <td>Ville de residence : </td>
63                 <td>
64                     <select name="ville">
65                         <option></option>
66                         <option>Abidjan</option>
67                         <option>Abengourou</option>
68                         <option>Bouaké</option>
69                         <option>Daloa</option>
70                         <option>Gagnoa</option>
71                         <option>Korogho</option>
72                         <option>Man</option>
73                         <option>Yamoussoukro</option>
74                     </select>
75                 </td>
76             <tr>
77             <tr>
78                 <td>Sexe : </td>
79                 <td>
80                     <label for="sexe1">Masculin</label> <input type="radio" value=
81                     "M" name="sexe" id="sexe1" />
82                     <label for="sexe2">Féminin</label> <input type="radio" value=
83                     "F" name="sexe" id="sexe2" />
84                 </td>
85             <tr>
86                 <td>Matricule* : </td> <td><input type="text" name="matricule"
87                 placeholder="Ex:1245AZ" required="" /></td>
88             <tr>
89                 <td>Mot de passe :</td> <td><input type="password" /></td>
90             <tr>
91                 <td>Confirmez le mot de passe :</td> <td><input type="password"
92                 name="motdepasse" /></td>
93             <tr>
94                 <td><input type="reset" value="Annuler" /></td> <td><input type=
95                 "submit" value="Valider"/></td>

```

```
95         </tr>
96     </table>
97 </form>
98 </center>
99 </body>
100 </html>
```

Cela donne ça :

Formulaire d'inscription

Nom* :	<input type="text"/>
Email :	<input type="text"/>
Tel* :	<input type="text"/>
Date de naissance :	<input type="text" value="Ex: 20/10/2000"/>
Ville de residence :	<input type="text"/> 
Sexe :	Masculin <input type="radio"/> Féminin <input type="radio"/>
Matricule* :	<input type="text"/>
Mot de passe :	<input type="password"/>
Confirmez le mot de passe :	<input type="password"/>
<input type="button" value="Annuler"/> <input type="button" value="Valider"/>	

Liens utiles

<https://openclassrooms.com/fr/courses/1603881-apprenez-a-creer-votre-site-web-avec-html5-et-css3/1607171-les-formulaires>

Exercice : Activité d'auto-évaluation 1

II

Exercice

Les balises qu'on utilise pour créer des formulaires sont (4 réponses) :

- ☐ `<select />`
- ☐ `<form> </form>`
- ☐ `<input />`
- ☐ `<file />`
- ☐ `<select> </select>`
- ☐ `<input> </input>`
- ☐ `<texterea> </texterea>`
- ☐ `<textarea> </textarea>`

Exercice

Pour rendre un champ obligatoire, on utilise l'attribut

Exercice

pour valider un formulaire on utilise la balise `<input type="checkbox" />` de type `checkbox`

Exercice

Pour réinitialiser un formulaire, on utilise la balise `<input type="reset" />` de type `reset`

Exercice

Pour indiquer quelle page php doit traiter le formulaire, on utilise l'attribut `action` de la balise `<form>`

Exercice

La méthode `POST` est plus sécurisée et plus robuste que la méthode `GET`

Exercice

On récupère les informations d'un formulaire sur le serveur grâce à l'attribut `$_POST` des balises

Traitement des formulaires en PHP



Lorsqu'un formulaire est transmis au serveur, les informations contenues dans le formulaire sont stockées au sein de variables superglobales.

- Si la méthode utilisée pour envoyer le formulaire est post, alors PHP crée une variable superglobale `$_POST` au sein de laquelle les informations sont stockées.
- Si la méthode utilisée pour envoyer le formulaire est get, alors PHP crée une variable superglobale `$_GET` au sein de laquelle les informations sont stockées.
- Si le formulaire contient des fichiers, alors PHP crée une variable superglobale `$_FILES` dans laquelle les fichiers sont stockés.
- Si les informations sont transmises par url, alors PHP crée une variable superglobale `$_REQUEST` au sein de laquelle les informations sont stockées.

La méthode get et request fonctionnent de la même manière en PHP et sont toutes deux équivalentes

1. Récupération du contenu d'un formulaire avec la méthode get et request

- Lorsqu'un formulaire est transmis au serveur de façon synchrone par le navigateur en utilisant la méthode get, les informations du formulaire sont injectées dans l'URL transmis au serveur. On peut voir ces informations apparaître dans l'url qui s'affiche dans la barre d'adresse du navigateur.
- Si un formulaire contient deux champs nom et age de valeurs respectives Koffi et 21, si ce formulaire est envoyé par la méthode get à une page inscription.php sur le serveur, on verra dans la barre d'adresse du navigateur une url de la forme : `http://.../inscription.php?nom=Koffi&age=21`
- Il est possible sans créer de formulaire de saisir directement cet url dans la barre d'adresse du navigateur, alors le navigateur va transmettre ces informations directement au serveur. On appelle cela transmission d'informations au serveur par request (ou get)

Remarque

Peu importe la méthode utilisée pour envoyer des informations sur le serveur, pour récupérer le contenu du formulaire il faut passer par une variable super-globale qui est sous forme de tableau associatif dont les clés sont les *name* des champs du formulaire et les valeurs sont les valeurs attribuées à ces champs par l'utilisateur.

- Dans l'exemple précédent, pour récupérer le nom transmis au serveur on fait : `$_GET["nom"]` ou bien `$_REQUEST["nom"]`

- Dans le même exemple, pour récupérer l'âge transmis au serveur on fait : `$_GET["age"]` ou bien `$_REQUEST["age"]`
- De façon générale pour récupérer une information avec get ou request on fait : `$_GET["name_du_champ"]` ou bien `$_REQUEST["name_du_champ"]`



Exemple : Affichage du nom et de l'âge de l'utilisateur

```
1 <?php
2     $nom= $_GET["nom"];
3     echo "Votre nom est : $nom <br/>";
4     echo "Votre âge est : ". $_GET['age'];
5 ?>
```



Remarque

Il est conseillé avant de récupérer une information dans un formulaire de vérifier d'abord que cette information existe.

- Pour vérifier si le nom existe, on utilise la fonction `isset()`. La fonction `isset()` renvoie true si la variable existe et false dans le cas contraire.

Ex: `if (isset($_GET["nom"])) { echo $_GET["nom"] ; }`

- Une information peut exister mais être vide, pour vérifier que l'information est non vide, on utilise la fonction `empty()`. La fonction `empty()` renvoie true si la variable est vide et false dans le cas contraire. Pour tester si la variable est donc non vide, on fait précéder `empty` de `!` (not : opérateur de négation).

Ex: `if (!empty($_GET["nom"])) { echo $_GET["nom"] ; }`

- Pour tester si le nom existe et est non vide, on procède donc comme suit :

```
if (isset($_GET["nom"]) && !empty($_GET["nom"]))
{
    echo $_GET["nom"] ;
}
```

2. Récupération du contenu d'un formulaire avec la méthode post

Avec la méthode post, les informations transmises par le navigateur ne sont pas visibles dans l'url. Cette méthode est plus robuste et plus sécurisée que la méthode get et request.

La manière de récupérer les informations est identique à ce qu'on a vu précédemment, à la seule différence qu'on utilise la super-variable globale `$_POST`.

Dans l'exemple précédent, si le formulaire est envoyé avec la méthode post voici comment afficher les informations :

```
1 <?php
2     $nom= $_POST["nom"];
3     echo "Votre nom est : $nom <br/>";
4     echo "Votre âge est : ". $_POST['age'];
5 ?>
```

Pour faire les tests on procède comme suit :

```
< ?php
```

```
if (isset($_POST["nom"]) && !empty($_POST["nom"]))
```

```

{
    echo $_POST["nom"] ;
}
?>

```

3. Travaux dirigés

Dans le travail dirigé précédent, nous avons réalisé un formulaire qui devrait être traité par une page inscription.php, nous allons dans cette partie écrire le contenu de cette page.

La page inscription.php doit récupérer le contenu du formulaire et afficher les informations renseignées dans un tableau en HTML.

Voici le contenu de la page inscription.php

```

1 <style type="text/css">
2   table {
3     border-collapse: collapse;
4   }
5   table td {
6     padding: 5px 15px;
7     border: 1px solid rgba(32,32,32,0.5);
8   }
9   table tr:nth-child(odd) {
10    background-color: aqua;
11  }
12 </style>
13 <center>
14   <h2>Les Informations renseignées sont:</h2>
15   <table>
16     <tr>
17       <td>Nom : </td> <td> <?php echo $_POST["nom"]; ?> </td>
18     </tr>
19     <tr>
20       <td>Tel : </td> <td> <?php echo $_POST["tel"]; ?> </td>
21     </tr>
22     <tr>
23       <td>Email : </td> <td> <?php echo $_POST["email"]; ?> </td>
24     </tr>
25     <tr>
26       <td>Matricule : </td> <td> <?php echo $_POST["matricule"]; ?> </td>
27     </tr>
28     <tr>
29       <td>Résidence : </td> <td> <?php echo $_POST["ville"]; ?> </td>
30     </tr>
31     <tr>
32       <td>Date de Naissance : </td> <td> <?php echo $_POST["date_naissance"]
33     </tr>
34     <tr>
35       <td>Sexe : </td> <td> <?php echo $_POST["sexe"]; ?> </td>
36     </tr>
37     <tr>
38       <td>Mot de passe : </td> <td> <?php echo $_POST["motdepasse"]; ?> </td>
39   </tr>

```

```

40     </table>
41 </center>
42

```

Nous avons un truc de ce genre après validation du formulaire

Les Informations renseignées sont:

Nom :	dizan
Tel :	78455512
Email :	paul@yahoo.fr
Matricule :	77AAZ
Résidence :	Abidjan
Date de Naissance :	2018-08-08
Sexe :	M
Mot de passe :	tata

- Dans ce code, nous n'avons pas fait les tests sur les variables pour éviter de trop écrire de code et de le rendre complexe. Toute fois, il est possible et recommandé de le faire pour une grande prudence.
- Nous n'avons pas défini le code conformément à la structure que doit avoir une page HTML (balises <!DOCTYPE> , <head> , <body>), parce que cela n'est pas obligatoire quand nous codons dans une page PHP.

Exercice : Activité d'auto-évaluation 2

IV

Exercice

Pour récupérer le contenu d'un formulaire, on utilise les variables super-globales (4 réponses)

- ☐ \$_SERVER
- ☐ \$_ENV
- ☐ \$_COOKIE
- ☐ \$_POST
- ☐ \$_GET
- ☐ \$_REQUEST
- ☐ \$_SESSION
- ☐ \$_FILES

Exercice

On considère un formulaire contenant deux champs de name nom et email. On aimerait récupérer et afficher le contenu de ce formulaire en php avec la méthode post, Veuillez compléter le code.

```
<?php
( ( ['nom']) ! (
['nom']) ){
    "Le nom saisi est : ". ['nom']. "<br/>";
}
( ( ['email']) ! (
['email']))){
$email= ['email'];
echo "L'adresse email saisie est : ";
}
?>
```

Exercice

On aimerait créer un formulaire contenant quatre champs (texte, email, mot de passe et fichier) et l'envoyer par la méthode post à une page appelé enregistrement.php. Les champs nom et email sont obligatoires. Les éléments du formulaire sont rangés dans un tableau. Veuillez compléter le code :

```
<form method=" " " ="enregistrement.php">
< >
<tr>
<td>Nom * </td> <td><input type=" " name="nom" " ="" /></td>
</tr>
<tr>
<td>Email * </td> <td><input type=" " name="email" " ="" /></td>
</tr>
<tr>
<td>Mot de passe </td> <td><input type=" " name="mot de passe" /></td>
</tr>
<tr>
<td>Photo </td> <td><input type=" " name="photo" /></td>
</tr>
<tr>
<td>< " type=" " value="Annuler" /></td> <td>< " type="
" value="valider" /></td>
</tr>
</ >
</form>
```

Validation d'un formulaire



Avant d'envoyer un formulaire sur le serveur, il est important de faire quelques vérifications côté client pour s'assurer que toutes les informations ont été correctement renseignées (s'assurer que le formulaire est valide). Pour faire cela, on utilise soit la validation par défaut intégrée dans les navigateur, soit directement le JavaScript ou le jQuery, soit bootstrap ou bien un plugin Javascript ou jquery. Dans cette partie, nous allons voir comment valider un formulaire côté client)

1. La validation par défaut

Depuis HTML5 les navigateurs offrent certaines capacités de validation des entrées de formulaire côté client :

- avec l'attribut *required* pour rendre une saisie obligatoire
- imposant un type d'entrée textuelle avec l'attribut *type* à la balise `<input />` en fonction de la nature de l'information qu'on aimerait que l'utilisateur saisisse. Les différents types textuels sont :
 - `text` : un texte simple
 - `url` : une adresse url
 - `date` : une date (calendrier)
 - `time` : l'heure
 - `datetime` : la date et l'heure
 - `week`, `month`, `datetime-local`
 - `search` : un champ de recherche
 - `color` : une couleur
 - `email` : une adresse email
 - `number` : un nombre
 - `tel` : un numéro de téléphone
- en imposant une plage de saisie avec les attributs *min*, *max* pour définir une valeur min et max pour les champs `number` et *minlength*, *maxlength* pour définir un nombre de caractère min et max pour les texte, etc.
- L'attribut *step* pour définir le pas d'incrémentation pour les champ `number`
- l'attribut *accept* pour définir un type de fichier à choisir pour les input types `file`
- en utilisant une expression régulière avec l'attribut *pattern* qui attend une *expression régulière* comme valeur

NB : tous les types textuels input ne sont pas reconnus et interprétés de la même manière par tous les navigateurs

Notion d'expression régulière

Une expression régulière (regex) est un modèle qui peut être utilisé pour faire correspondre des combinaisons de caractères dans des chaînes de texte, de sorte qu'elles sont idéales pour la validation de formulaires (ainsi que diverses autres utilisations en JavaScript). Les Regex sont assez complexes et nous n'avons pas l'intention de vous les enseigner de manière exhaustive dans cette leçon.

Vous trouverez ci-dessous quelques exemples pour vous donner une idée de base de leur fonctionnement :

- `a` : correspond à un caractère qui doit être un a (ni b, ni aa, etc.)
- `abc` : correspond à a, suivi de b, suivi de c.
- `a*` : correspond au caractère a, absent ou présent plusieurs fois (+ correspond à un caractère une ou plusieurs fois).
- `[^a]` : correspond à un caractère qui n'est pas un a.
- `a|b` : correspond à un caractère qui est a ou b.
- `[abc]` : correspond à un caractère qui est a, b ou c.
- `[^abc]` : correspond à un caractère qui n'est pas a, b ou c.
- `[a-z]` : correspond à tout caractère de la plage a–z, en minuscules seulement (utilisez `[A-Za-z]` pour minuscules et majuscules et `[A-Z]` pour les majuscules uniquement).
- `a.c` : correspond à a, suivi par n'importe quel caractère, suivi par c.
- `a{5}` : correspond à a, 5 fois.
- `a{5,7}` : correspond à a, 5 à 7 fois, mais ni plus, ni moins.

Vous pouvez utiliser des nombres ou d'autres caractères dans ces expressions, comme :

- `[-]` : correspond à une espace ou un tiret.
- `[0-9]` : correspond à tout nombre compris entre 0 et 9.

Vous pouvez combiner cela pratiquement comme vous l'entendez en précisant les différentes parties les unes après les autres :

- `[Ll].*k` : Un seul caractère L en majuscules ou minuscules, suivi de zéro ou plusieurs caractères de n'importe quel type, suivis par un k minuscules.
- `[A-Z][A-Za-z' -]+` : Un seul caractère en majuscules suivi par un ou plusieurs caractères en majuscules ou minuscules, un tiret, une apostrophe ou une espace. Cette combinaison peut s'utiliser pour valider les nom de villes dans les pays anglo-saxons ; ils débutent par une majuscule et ne contiennent pas d'autre caractère.
- `[0-9]{3}[-][0-9]{3}[-][0-9]{4}` : Un schéma pour un numéro de téléphone intérieur américain — trois chiffres, suivis par une espace ou un tiret, suivis par trois nombres, suivis par une espace ou un tiret, suivis par quatre nombres. Vous aurez peut-être à faire plus compliqué, car certains écrivent leur numéro de zone entre parenthèses, mais ici il s'agit simplement de faire une démonstration.

Exemple

```
1 <form>
2   <label for="choix">Préférez-vous la banane ou la cerise ?</label>
3   <input id="choix" name="f_like" required pattern="banane|cerise">
4   <input type="text" name="tel" pattern="[0-9]{8}" placeholder= "Veuillez saisir
un numéro de téléphone à 8 chiffres" required />
5   <input type="password" pattern="(?!^.{8,}$)((?=.*d)|(?=.*W+))(?![.n])(?=.*[A-Z])
(?=.*[a-z]).*" title="Au moins 8 caractères, un chiffre, une lettre majuscule et
une minuscule" name="code"/>
6   <button>Soumettre</button>
```

```
7 </form>
```

Les pseudo-classes de validation

Si certains champs ne sont pas valides, il est important de donner une apparence particulière à ces champs pour attirer l'attention de l'utilisateur. Pour cela, nous avons certains nombres de pseudo-classe css que sont :

- valid : permet de définir un style si le champ est valide
- invalid : permet de définir un style si le champ n'est pas valide
- in-range : valeurs dans les limites
- out-of-range : valeurs hors limites
- required : champs obligatoires

Exemple

```
1 <style type="text/css">
2   :invalid {
3     background-color: #ffdddd;
4   }
5   :valid {
6     background-color: #ddffdd;
7   }
8   input:required {
9     border: 1px solid blue;
10  }
11 </style>
12 <form>
13 <label>Username</label><input type="text" />
14 <label>Email</label><input type="email" required />
15 </form>
```

Quelques liens utiles

<https://dmouronval.developpez.com/tutoriels/javascript/api-contrainte-validation/>

<https://www.alsacreations.com/tuto/lire/1372-formulaires-html5-nouveaux-types-champs-input.html>

https://developer.mozilla.org/fr/docs/Web/Guide/HTML/Formulaires/Validation_donnees_formulaire

<https://laravel.sillo.org/bootstrap-4-les-formulaires-2-2/>

<http://www.pompage.net/traduction/formulaires-html5-css3-validation>

2. La validation en JavaScript ou JQuery

La validation par défaut intégrée dans les navigateurs est limité. Par fois, nous voulons soit :

- Vérifier que deux valeurs sont égales
- vérifier que la taille d'un fichier choisi est inférieure à une valeur
- soit faire une vérification coté serveur par ajax (Ex : vérifier que l'email saisie n'existe pas dans la base de données déjà)
- soit faire des vérifications pendant la saisie des informations dans le navigateur et afficher des messages particuliers
- Remplir automatiquement un champ à partir d'autres champs renseignés

- modifier l'état des champs pendant la saisie
- afficher des boites d'alerte
- etc.

Pour ce fait, il est important de faire des vérifications par JavaScript ou par JQuery.

Nous allons faire un exemple permettant de vérifier que deux mots de passes saisis sont équivalents, pour cela nous allons :

- travailler avec le JQuery : A télécharger à l'adresse <https://code.jquery.com/jquery-2.x-git.js> , vous maintenez la touche *ctrl* de votre clavier et vous appuyez sur la touche *S* pour l'enregistrer sur votre ordinateur, ou bien vous copiez le contenu du fichier et vous créez un fichier JavaScript dans lequel vous le collez. Moi, le nom entier de mon fichier est *jquery.js*, vous pouvez le renommer pour vous-même. Votre fichier jquery doit se trouver là où votre fichier php se trouve, puis vous mettez ce code dans la balise `<head>` de votre fichier php : `<script src="jquery.js" type="text/javascript"></script>`
- Utiliser l'événement *onblur* qui se déclenche et qui permet d'exécuter une fonction lorsqu'on quitte un champ de saisie
- Utiliser la fonction `alert()` pour afficher un message en cas d'erreur
- On appellera notre fonction de vérification `checkPassword()`

Pour récupérer le contenu d'un champ en JQuery en fait `$(sélécteur).val()`, où *sélécteur* représente un sélecteur css entre guillemets permettant de sélectionner la balise.

Ci-dessous le code à saisir

```

1 <!DOCTYPE HTML>
2 <html>
3 <head>
4   <meta http-equiv="content-type" content="text/html" />
5   <title>Inscription</title>
6   <style type="text/css">
7     form {
8       padding: 10px;
9       background-color: rgb(245,245,245);
10      padding: 10px;
11      border-radius: 7px 7px 0px 0px;
12      display: inline-block;
13      box-shadow: 0px 2px 6px rgba(32,32,32,0.5);
14    }
15    form table td {
16      padding: 5px 10px;
17    }
18    input[type="password"] {
19      width: 300px;
20      height: 30px;
21      border-radius: 7px;
22      border: none;
23      padding-left: 5px;
24    }
25    input[type="submit"]{
26      background-color: rgb(0,0,150);
27      color: white;
28      border: none;
29      border-radius: 7px;
30      padding: 5px 10px;
31      transition: all 0.5s;

```

```

32     }
33     input[type="submit"]:hover{
34         background-color: rgb(255,128,0);
35         padding: 5px 25px;
36         border-radius: 10px;
37         cursor: pointer;
38     }
39 </style>
40 <script src="jquery.js" type="text/javascript"></script>
41 <script type="text/javascript">
42     function checkPassword() {
43         $pswd1=$('#password1').val();
44         $pswd2=$('#password2').val();
45         if($pswd1!="" && $pswd2 != ""){
46             if($pswd1!=$pswd2){
47                 alert("Les deux mots de passe ne correspondent pas");
48             }
49         }
50     }
51 </script>
52 </head>
53 <body>
54     <h1 style="text-decoration: underline;"><center>Formulaire d'inscription</
55     center></h1>
56     <center>
57         <form method="post" action="inscription.php">
58             <table>
59                 <tr>
60                     <td>Mot de passe :</td> <td><input type="password" id="password1"
61                     onblur="checkPassword();" /></td>
62                 </tr>
63                 <tr>
64                     <td>Confirmez le mot de passe :</td> <td><input type="password" id
65                     ="password2" onblur="checkPassword();" name="motdepasse" /></td>
66                 </tr>
67                 <tr>
68                     <td></td> <td><input type="submit" value="Valider"/></td>
69                 </tr>
70             </table>
71         </form>
72     </center>
73 </body>
74 </html>

```

Liens utiles

<https://openclassrooms.com/fr/courses/146276-tout-sur-le-javascript/144576-td-verification-dun-formulaire>

<https://openclassrooms.com/fr/courses/146276-tout-sur-le-javascript/144456-les-formulaires-de-bons-amis>

3. La validation par bootstrap

Bootstrap est un framework css qui dispose des fonctions javascript permettant de valider un formulaire.

Bootstrap :

- offre des styles très attrayants et ergonomiques permettant de créer de jolis formulaire et adaptable à tous types d'ecrans

- offre des script JavaScript permettant de valider un formulaire pendant la saisie comme après la saisie
- facilite grandement la création de formulaire et se base également sur la validation par défaut intégré dans les formulaires
- Bootstrap dispose des classes tels que : `form-control-success`, `form-control-warning`, `form-control-danger` que nous pouvons injecter dynamiquement en JavaScript dans les balises après vérification pour montrer qu'une saisie a été bien faite, avertir l'utilisateur ou signaler une erreur de saisie. Voir le lien <https://v4-alpha.getbootstrap.com/components/forms/#validation>
- Pour afficher des messages d'erreur ou des indications, nous avons les class `form-control-feedback`, `form-text`, `text-muted`, que nous pouvons appliquer sur des balises

Pour utiliser bootstrap (version 4.1) :

- il faut d'abord aller à l'adresse <https://getbootstrap.com/docs/4.1/getting-started/download/> puis cliquez sur le bouton download
- Après téléchargement, il faut décompresser le fichier et mettre le dossier *bootstrap-4.1.3-dist* (ce dossier contient directement le dossier js et css de bootstrap) dans votre projet php, là où se trouve votre page php
- Après il faut rattacher les fichiers bootstrap.css et bootstrap.js à la page php (grâce aux balises `<style>` et `<script>`)
- Bootstrap a besoin de jQuery.js et popper.js (pour la version 4) pour fonctionner, donc il faut rattacher ces deux scripts à la page.
 - Pour JQuery il est conseillé d'utiliser la version 3 que vous pouvez télécharger à l'adresse <https://code.jquery.com/jquery-3.3.1.min.js> et l'enregistré sous le nom *jquery-3.3.1.min.js*
 - pour popper, vous pouvez télécharger à l'adresse <https://unpkg.com/popper.js@1.14.4/dist/umd/popper.min.js> et l'enregistré sous le nom *popper.min.js* dans votre projet

Ci-dessous quelques liens pour découvrir comment utiliser bootstrap pour la création et la validation d'un formulaire

<https://getbootstrap.com/docs/4.1/components/forms/>

<https://v4-alpha.getbootstrap.com/components/forms/>

Voici à quoi ressemble l'entête d'une page web utilisant bootstrap version 4. la balise `<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"/>` permet d'adapter l'affichage aux différents écrans.

La structure actuelle de notre projet est :

Disque local (C:) > wamp > www > helloWorld			
Nom	Modifié le	Type	Taille
bootstrap-4.1.3-dist	19/08/2018 10:45	Dossier de fichiers	
index.php	19/08/2018 11:42	PHP Script	1 Ko
inscription.php	19/08/2018 12:23	PHP Script	2 Ko
jquery-3.3.1.min.js	19/08/2018 11:18	Fichier JS	85 Ko
popper.min.js	19/08/2018 11:22	Fichier JS	21 Ko

le fichier index.php contient notre formulaire

```

1 <!DOCTYPE HTML>
2 <html>
3 <head lang="fr">
4   <meta http-equiv="content-type" content="text/html" />
5   <meta charset="utf-8"/>
6   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
  fit=no"/>
7
```

```

8     <link href="bootstrap-4.1.3-dist/css/bootstrap.min.css" rel="stylesheet" />
9
10    <script src="jquery-3.3.1.min.js" type="text/javascript"></script>
11    <script src="popper.min.js" type="text/javascript"></script>
12    <script src="bootstrap-4.1.3-dist/js/bootstrap.min.js" type="text/javascript">
13    </script>
14 </head>
15 <body>
16
17 </body>
18 </html>

```

Création de formulaire avec bootstrap

Nous allons reprendre la création de notre formulaire initiale en utilisant bootstrap.

```

1 <!DOCTYPE HTML>
2 <html>
3 <head lang="fr">
4   <meta http-equiv="content-type" content="text/html" />
5   <meta charset="utf-8"/>
6   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
7   fit=no"/>
8   <link href="bootstrap-4.1.3-dist/css/bootstrap.min.css" rel="stylesheet" />
9
10  <script src="jquery-3.3.1.min.js" type="text/javascript"></script>
11  <script src="popper.min.js" type="text/javascript"></script>
12  <script src="bootstrap-4.1.3-dist/js/bootstrap.min.js" type="text/javascript">
13  </script>
14  <style type="text/css">
15    form {
16      width: 500px;
17      max-width: 100%;
18      background-color: rgb(245,245,245);
19      border-radius: 7px 7px 0px 0px;
20      padding: 10px;
21      text-align: left;
22    }
23  </style>
24 </head>
25 <body>
26   <h2 style="text-decoration: underline;"><center>Formulaire d'inscription</
27   center></h2>
28   <center>
29     <form method="post" action="inscription.php">
30       <div class="form-group row">
31         <label class="col-sm-4 col-form-label">Nom* : </label>
32         <div class="col-sm-8">
33           <input type="text" class="form-control" required />
34         </div>
35       </div>
36       <div class="form-group row">
37         <label class="col-sm-4 col-form-label">Tel* : </label>
38         <div class="col-sm-8">
39           <input type="number" pattern="d{8}" class="form-control"
40           placeholder="Tel 8 chiffres" required />
41         </div>

```

```

39     </div>
40     <div class="form-group row">
41         <label class="col-sm-4 col-form-label">Email : </label>
42         <div class="col-sm-8">
43             <div class="input-group">
44                 <input type="email" class="form-control" required />
45                 <div class="input-group-prepend">
46                     <span class="input-group-text">@</span>
47                 </div>
48             </div>
49         </div>
50     </div>
51     <div class="form-group row">
52         <label class="col-sm-4 col-form-label">Date de Naissance : </label>
53         <div class="col-sm-8">
54             <input type="date" class="form-control" />
55         </div>
56     </div>
57     <div class="form-group row">
58         <label class="col-sm-4 col-form-label">Matricule : </label>
59         <div class="col-sm-8">
60             <input type="text" placeholder="Ex: 475AM" required="" class=
"form-control" />
61         </div>
62     </div>
63     <div class="form-group row">
64         <label class="col-sm-4 col-form-label">Ville</label>
65         <div class="col-sm-8">
66             <select class="form-control" >
67                 <option></option>
68                 <option>Abidjan</option>
69                 <option>Abengourou</option>
70                 <option>Bouaké</option>
71                 <option>Daloa</option>
72                 <option>Gagnoa</option>
73                 <option>Korogho</option>
74                 <option>Man</option>
75                 <option>Yamoussoukro</option>
76             </select>
77         </div>
78     </div>
79     <div class="form-group row">
80         <label class="col-sm-4 col-form-label">Sexe : </label>
81         <div class="col-sm-8">
82             <div class="custom-control custom-radio custom-control-inline">
83                 <input type="radio" id="sexe1" name="sexe" value="M" class=
"custom-control-input"/>
84                 <label class="custom-control-label" for="sexe1">Masculin</label>
85             </div>
86             <div class="custom-control custom-radio custom-control-inline">
87                 <input type="radio" id="sexe2" name="sexe" value="F" class=
"custom-control-input"/>
88                 <label class="custom-control-label" for="sexe2">Féminin</label>
89             </div>
90         </div>
91     </div>
92     <div class="form-group row">
93         <label class="col-sm-4 col-form-label">Mot de passe: </label>
94         <div class="col-sm-8">

```

```
95         <input type="password" class="form-control" />
96     </div>
97 </div>
98 <div class="form-group row">
99     <label class="col-sm-4 col-form-label">Confirmez le mot de passe: </
label>
100     <div class="col-sm-8">
101         <input type="password" class="form-control" name="motdepasse" />
102     </div>
103 </div>
104 <center><button type="submit" class="btn btn-primary btn-sm">Valider</
button></center>
105 </form>
106 </center>
107 </body>
108 </html>
```

Formulaire d'inscription

Nom* :	<input type="text"/>
Tel* :	<input type="text" value="Tel 8 chiffres"/>
Email :	<input type="text"/> @
Date de Naissance :	<input type="text" value="jj/mm/aaaa"/>
Matricule :	<input type="text" value="Ex: 475AM"/>
Ville	<input type="text"/>
Sexe :	<input type="radio"/> Masculin <input checked="" type="radio"/> Féminin
Mot de passe:	<input type="password"/>
Confirmez le mot de passe:	<input type="password"/>
<input type="button" value="Valider"/>	

4. La validation par plugin JQuery ou Javascript

Bootstrap offre de jolis style css pour la création d'un formulaire, et dispose des class css pour afficher des informations, alertes et indications, mais demeure limité dans son fonctionnement en ce qui concerne la validation d'un formulaire. Il est possible de valider en JavaScript ou jQuery un formulaire de façon plus avancée sans saisir assez de code grâce à des scripts javascript ou jquery (plugins) déjà mise en place par des développeurs. Voici quelques plugins qui peuvent grandement vous aider dans la validation de votre formulaire.

- *JQuery Form Validator* : <http://www.formvalidator.net/>
- *Bootstrap Validator* : <http://1000hz.github.io/bootstrap-validator/> et <https://github.com/1000hz/bootstrap-validator>
- *Smoke* : <http://alfredobarron.github.io/smoke/#/getting-started> et <http://alfredobarron.github.io/smoke/#/validate>
- *jQuery DjValidator* : <https://www.jqueryscript.net/form/HTML5-Form-Validation-DjValidator.html>
- *jQuery Validation Plugin* : <https://jqueryvalidation.org/> et <https://www.youtube.com/watch?v=yaxUV3Ib4vI>
- *jQuery Validate* : <https://www.pierrefay.fr/blog/jquery-validate-formulaire-validation-tutoriel.html> et <https://jqueryvalidation.org/>
- *validatr* : <http://jaymorrow.github.io/validatr/> et <http://jaymorrow.github.io/validatr/fields.html>
- *parsley* : <http://parsleyjs.org/doc/index.html> et <http://parsleyjs.org/doc/examples/events.html>
- *Validetta* : <http://lab.hasanaydogdu.com/validetta/>
- *bootstrap-validate* : <https://bootstrap-validate.js.org/usage.html> et <https://github.com/PascaleBeier/bootstrap-validate>
- autres : <https://www.jqueryscript.net/tags.php?/form%20validation/>

Vérification de formulaire avec bootstrap Validator

Vous devrez d'abord télécharger le plugin à l'adresse <https://github.com/1000hz/bootstrap-validator> et l'intégrer dans votre projet

```

1 <!DOCTYPE HTML>
2 <html>
3 <head lang="fr">
4   <meta http-equiv="content-type" content="text/html" />
5   <meta charset="utf-8"/>
6   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
  fit=no"/>
7
8   <link href="bootstrap-4.1.3-dist/css/bootstrap.min.css" rel="stylesheet" />
9
10  <script src="jquery-3.3.1.min.js" type="text/javascript"></script>
11  <script src="popper.min.js" type="text/javascript"></script>
12  <script src="bootstrap-4.1.3-dist/js/bootstrap.min.js" type="text/javascript">
  </script>
13  <script src="bootstrap-validator-master/dist/validator.min.js"></script>
14  <style type="text/css">
15    form {
16      width: 500px;
17      margin: 50px auto;
18      max-width: 100%;
19      background-color: rgb(245,245,245);
20      padding: 10px;
  
```

```

21     }
22     </style>
23 </head>
24 <body>
25
26     <form data-toggle="validator" role="form" method="post" action=
"inscription.php">
27         <div class="form-group">
28             <label for="inputName" class="control-label">Name</label>
29             <input type="text" class="form-control" id="inputName" pattern="^[A-Z]
[A-Za-z'- ]+" required>
30             <div class="help-block with-errors">Au moins 2 Caractères et commencé
par une majuscule</div>
31         </div>
32         <div class="form-group has-feedback">
33             <label for="inputTwitter" class="control-label">Twitter</label>
34             <div class="input-group">
35                 <span class="input-group-addon">@</span>
36                 <input type="text" pattern="^[_A-z0-9]{1,}$" data-minlength="4"
data-error="Au moins 4 caractères" maxlength="15" class="form-control" id=
"inputTwitter" placeholder="1000hz" required>
37             </div>
38             <span class="glyphicon form-control-feedback" aria-hidden="true"></
span>
39             <div class="help-block with-errors"> </div>
40         </div>
41         <div class="form-group">
42             <label for="inputEmail" class="control-label">Email</label>
43             <input type="email" class="form-control" id="inputEmail" placeholder=
"Email" data-error="L'adresse email est invalide" required>
44             <div class="help-block with-errors"></div>
45         </div>
46         <div class="form-group">
47             <label for="inputPassword" class="control-label">Password</label>
48             <div class="form-inline row">
49                 <div class="form-group col-sm-6">
50                     <input type="password" data-minlength="6" class="form-control" id=
"inputPassword" placeholder="Password" required>
51                     <div class="help-block">Minimum 6 caractères </div>
52                 </div>
53                 <div class="form-group col-sm-6">
54                     <input type="password" class="form-control" id=
"inputPasswordConfirm" data-match="#inputPassword" data-match-error="Les deux
mots de passe ne correspondent pas" placeholder="Confirmez" required>
55                     <div class="help-block with-errors">Minimum 6 caractères</div>
56                 </div>
57             </div>
58         </div>
59         <div class="form-group">
60             <div class="radio">
61                 <label>
62                     <input type="radio" name="underwear" required>
63                     Boxers
64                 </label>
65             </div>
66             <div class="radio">
67                 <label>
68                     <input type="radio" name="underwear" required>
69                     Briefs
70                 </label>
71             </div>
72         </div>

```

```
73         <div class="form-group">
74             <center> <button type="submit" class="btn btn-primary">Submit</button>
75         > </center>
76     </div>
77 </form>
78 </body>
79 </html>
```

Exercice : Activité d'auto-évaluation 3

VI

Exercice

Quelques plugins jQuery de validation de formulaire sont (4 réponses)

- ☐ Cake php
- ☐ JQuery Form Validator
- ☐ Smoking
- ☐ jQuery DjValidator
- ☐ php plugin
- ☐ Bootstrap Validator
- ☐ Smoke
- ☐ Strong CSS JS

Exercice

Pour définir des contraintes sur des champs de saisi, on peut utiliser les attributs : (4 réponses)

- ☐ min, max, step
- ☐ minlength, maxlength
- ☐ acceptator
- ☐ accept
- ☐ data-valide
- ☐ type
- ☐ minimum, maximum
- ☐ href

Exercice

L'attribut permet de faire une vérification sur un champ de saisie par le biais d'une régulière

Exercice

Pour déclencher un traitement lorsqu'on quitte un champs de saisie, on utilise l'attribut

Exercice

Pour déclencher un traitement lorsque la valeur d'un champ change, on utilise l'attribut



Envoi et traitement des fichiers

VII

1. Envoi d'un fichier et récupération d'un fichier

Les actions côté client

- Pour envoyer un fichier on utilise la balise `<input />` de type `file`
- Il faut ajouter l'attribut `enctype="multipart/form-data"` à la balise `<form>`
- Pour choisir le type de fichier, on utilise l'attribut `accept` de la balise `file`. Ex : `<input type="file" name="photo" accept="image/*" />` ou `<input type="file" name="photo" accept=".pdf,.doc" />`

Quelques valeurs que peut prendre `accept` sont :

- `image/*` : pour tous types d'images
- `video/*` : pour tous types de fichiers vidéos
- `audio/*` : pour tous types de fichiers audio
- `image/jpeg` : pour des images jpg
- `image/png` : pour des images png
- `video/mp4` : pour des vidéos mp4
- `.pdf, .png, .gif, .doc, .docx, .mp4`, etc. : pour un ou des types de fichiers donnés
- etc.

Traitement côté serveur

- Pour récupérer les fichiers on passe par la variable super-globale `$_FILES`
- Pour enregistrer le fichier on fait appel à la fonction `move_uploaded_file()`

👉 Exemple

Soit un fichier issu d'une balise `file` de name `photo`, le code php pour enregistrer le fichier dans un dossier nommé "avatars" est :

```
1 <?php
2 if(isset($_FILES['photo']) && !empty($_FILES['photo']['name'])) {
3     $nomfichier=$_FILES['photo']['name'];
4     $tmp=$_FILES['photo']['tmp_name'];
5     $dossier="avatars/"; // le dossier avatars doit être créé dans votre projet là
    où se trouve la page php
6     if(move_uploaded_file($tmp,$dossier.$nomfichier)) {
7         echo "Le fichier a été enregistré avec succès";
8     }
```

```

9 }else echo "Le fichier photo n'existe pas";
10 ?>

```

Le code avec des tests de vérification supplémentaire est :

```

1 <?php
2 if(isset($_FILES['photo']) && !empty($_FILES['photo']['name'])){
3     $nomfichier=$_FILES['photo']['name'];
4     $tmp=$_FILES['photo']['tmp_name'];
5     $dossier="avatars/";
6     if (is_uploaded_file($tmp)) //permet de vérifier si le fichier a été uploadé
    via http
7     {
8         if(file_exists($dossier)){ //permet de vérifier que le répertoire existe
9             if(move_uploaded_file($tmp,$dossier.$nomfichier)){
10                 echo "Le fichier a été enregistré avec succès";
11             }
12         }else echo "Le dossier $dossier n'existe pas";
13     }else $resp="Le fichier n'a pas été uploadé";
14 }else echo "Le fichier photo n'existe pas";
15 ?>

```

2. travaux dirigés

Libellé

Nous allons reprendre le formulaire initial en intégrant un fichier image.






- Le fichier image doit être inférieur à 2Mo.
- Nous donnons un aperçu du fichier lorsqu'il est choisi
- Nous allons rendre plus stylé la balise `<input />` de type file (l'apparence par défaut n'est pas trop agréable)
- Le nom de l'utilisateur doit être au moins deux lettres et commencé par une lettre majuscule
- le numéro de téléphone doit être exactement de 8 chiffres
- Le matricule commence par UV et est suivi de 4 chiffres, puis se termine par deux lettres majuscules
- les champs matricules, nom et tel et email sont obligatoires
- On doit vérifier que les deux mots de passe sont égaux avec au moins 8 caractères
- On utilisera bootstrap pour la création du formulaire et bootstrap Validator pour la validation du formulaire
- Après validation du formulaire, nous enregistrons l'image dans un dossiers users et nous affichons dans une table HTML les informations saisies par l'utilisateur y compris la photo qu'il à choisi. Si aucune photo n'a été choisie, nous affichons une image par défaut.
- Il peut s'avérer que le nom de l'image choisi par l'utilisateur existe déjà sur le serveur, pour remédier à ce problème (éviter que l'image écrase une autre image sur le serveur) avant d'enregistrer l'image, nous allons vérifier si le nom existe déjà, si tel est le cas, on ajoute un suffixe entier au nom de l'image pour que le nom soit unique sur le serveur.
- on enregistrera les noms des images avec la fonction `utf8_encode()` afin de gérer le problème des caractères accentués

Correction

Ci-dessous le code à réaliser :

Contenu du projet :

- Le dossier js contient les fichiers jquery-3.3.1.min.js , popper.min.js et validator.min.js
- le dossier bootstrap-4.1.3 contient les dossier css et js de bootstrap

Disque local (C:) ▸ wamp ▸ www ▸ helloWorld				
Nom	Modifié le	Type	Taille	
 bootstrap-4.1.3	19/08/2018 10:45	Dossier de fichiers		
 js	19/08/2018 22:44	Dossier de fichiers		
 users	20/08/2018 03:47	Dossier de fichiers		
 index.php	20/08/2018 03:35	PHP Script	8 Ko	
 inscription.php	20/08/2018 03:46	PHP Script	4 Ko	

Contenu de la page index.php (la page contenant le formulaire : page d'entrée ou d'accueil)

```

1 <!DOCTYPE HTML>
2 <html>
3 <head lang="fr">
4   <meta http-equiv="content-type" content="text/html" />
5   <meta charset="utf-8"/>
6   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
  fit=no"/>
7
8   <link href="bootstrap-4.1.3-dist/css/bootstrap.min.css" rel="stylesheet" />
9
10  <script src="js/jquery-3.3.1.min.js" type="text/javascript"></script>
11  <script src="js/popper.min.js" type="text/javascript"></script>
12  <script src="bootstrap-4.1.3-dist/js/bootstrap.min.js" type="text/javascript">
</script>
13  <script src="js/validator.min.js"></script>
14  <style type="text/css">
15    form {
16      width: 800px;
17      max-width: 100%;
18      background-color: rgb(245,245,245);
19      border-radius: 7px 7px 0px 0px;
20      padding: 10px;
21      text-align: left;
22      margin: 25px;
23      box-shadow: 0px 2px 8px rgba(32,32,32,0.5);
24    }
25    .display-none {
26      display: none;
27    }
28  </style>
29  <script type="text/javascript">
30    function choisirPhoto(x){
31      $(x).parents('.image-viewer').children('input[type="file"]').click();
32    }
33    function show_apercu(x){
34      var files = jQuery(x)[0].files;
35      if (files.length > 0) {
36        var file=files[0];
37        if(file.size<=2*1024*1024){
38          var file=files[0];

```



```

39         $(x).prevAll('.file-name').html(file.name+" "+parseInt(file.
size/1024) + ' Ko');
40         $(x).prevAll('.img_aperçu').attr('src', window.URL.
createObjectURL(file));
41         $(x).prevAll('.img_aperçu').show();
42         $(x).parent('.image-viewer').children('a').show();
43     }else{
44         alert("Image trop lourde, le poids doit être inférieur à 2M");
45     }
46 }
47 }
48
49 function removeFile(x){
50     $(x).nextAll('input').val("");
51     $(x).prevAll('.file-name').html("");
52     $(x).prevAll('.img_aperçu').hide();
53     $(x).parent('.image-viewer').children('a').hide();
54 }
55 </script>
56 </head>
57 <body>
58     <h2 style="text-decoration: underline;"><center>Formulaire d'inscription</
center></h2>
59     <center>
60         <form data-toggle="validator" role="form" method="post" action="inscription.
php" enctype="multipart/form-data">
61             <div class="form-group row">
62                 <label class="col-sm-4 col-form-label">Nom* : </label>
63                 <div class="col-sm-8">
64                     <input type="text" class="form-control" name="nom" pattern="^[A-Z]
[A-Za-z'-]+> required />
65                     <div class="help-block with-errors">Au moins 2 Caractères et
commencé par une majuscule</div>
66                 </div>
67             </div>
68             <div class="form-group row">
69                 <label class="col-sm-4 col-form-label">Tel* : </label>
70                 <div class="col-sm-8">
71                     <input type="text" pattern="^[0-9]{8}$" name="tel" class="form-
control" data-error="Le Numéro doit être 8 chiffres" required />
72                     <div class="help-block with-errors"></div>
73                 </div>
74             </div>
75             <div class="form-group row">
76                 <label class="col-sm-4 col-form-label">Email : </label>
77                 <div class="col-sm-8">
78                     <div class="input-group">
79                         <input type="email" class="form-control" name="email" data-
error="Le format d'email n'est pas correct" required />
80                         <div class="input-group-prepend">
81                             <span class="input-group-text">@</span>
82                         </div>
83                     </div>
84                     <div class="help-block with-errors"></div>
85                 </div>
86             </div>
87             <div class="form-group row">
88                 <label class="col-sm-4 col-form-label">Date de Naissance : </label>
89                 <div class="col-sm-8">
90                     <input type="date" class="form-control" name="date_naissance" />
91                 </div>

```

```

92         </div>
93         <div class="form-group row">
94             <label class="col-sm-4 col-form-label">Matricule* : </label>
95             <div class="col-sm-8">
96                 <input type="text" name="matricule" placeholder="Ex: UV4578AX"
pattern="^UV[0-9]{4}[A-Z]{2}$" required class="form-control" />
97                 <div class="help-block with-errors">
98                     Le matricule doit commencer par UV suivi de 4 chiffres, puis
se termine par deux lettres Majuscule
99                 </div>
100             </div>
101         </div>
102         <div class="form-group row">
103             <label class="col-sm-4 col-form-label">Ville</label>
104             <div class="col-sm-8">
105                 <select class="form-control" name="ville" >
106                     <option></option>
107                     <option>Abidjan</option>
108                     <option>Abengourou</option>
109                     <option>Bouaké</option>
110                     <option>Daloa</option>
111                     <option>Gagnoa</option>
112                     <option>Korogho</option>
113                     <option>Man</option>
114                     <option>Yamoussoukro</option>
115                 </select>
116             </div>
117         </div>
118         <div class="form-group row">
119             <label class="col-sm-4 col-form-label">Sexe : </label>
120             <div class="col-sm-8">
121                 <div class="custom-control custom-radio custom-control-inline">
122                     <input type="radio" id="sexe1" name="sexe" value="M" class=
"custom-control-input"/>
123                     <label class="custom-control-label" for="sexe1">Masculin</label>
124                 </div>
125                 <div class="custom-control custom-radio custom-control-inline">
126                     <input type="radio" id="sexe2" name="sexe" value="F" class=
"custom-control-input"/>
127                     <label class="custom-control-label" for="sexe2">Féminin</label>
128                 </div>
129             </div>
130         </div>
131         <div class="form-group row">
132             <label class="col-sm-4 col-form-label">Mot de passe: </label>
133             <div class="col-sm-8">
134                 <input type="password" id="inputPassword" data-minlength="8" class
="form-control" />
135                 <div class="help-block with-errors">Minimum 8 caractères</div>
136             </div>
137         </div>
138         <div class="form-group row">
139             <label class="col-sm-4 col-form-label">Confirmez le mot de passe: </
label>
140             <div class="col-sm-8">
141                 <input type="password" class="form-control" name="motdepasse"
data-minlength="8" id="inputPasswordConfirm" data-match="#inputPassword" data-
match-error="Les deux mots de passe ne correspondent pas" />
142                 <div class="help-block with-errors">Minimum 8 caractères</div>
143             </div>
144         </div>


```

```

145
146         <div class="form-group row" >
147             <label class="col-sm-4">Photo:</label>
148             <div class="col-sm-8 image-viewer">
149                 <span class="btn btn-primary btn-sm onclick="choisirPhoto(this);">
Choisir une image</span>
150                 <img class="display-none img_aperçu" style="width: 150px;height:
auto;" />
151                 <span class="file-name"></span>
152                 <a id="file" class="btn btn-danger display-none btn-remove" onclick=
"removeFile(this);">X</a>
153                 <input type="file" class="form-control" style="display: none;"
name ="photo" id="photo" onchange="show_aperçu(this);" accept="image/*"/>
154             </div>
155         </div>
156         <hr />
157         <center><button type="submit" class="btn btn-danger btn-sm">Valider</
button></center>
158     </form>
159 </center>
160 </body>
161 </html>

```

Aperçu dans le navigateur de la page index.php

Nom* :	<input type="text" value="Koffi Jean"/>
	Au moins 2 Caractères et commencé par une majuscule
Tel* :	<input type="text" value="47859652"/>
Email :	<input type="text" value="jean@yahoo.fr"/> @
Date de Naissance :	<input type="text" value="20/07/2017"/>
Matricule* :	<input type="text" value="UV7845AB"/>
	Le matricule doit commencer par UV suivi de 4 chiffres, puis se termine par deux lettres Majuscule
Ville	<input type="text" value="Abidjan"/>
Sexe :	<input checked="" type="radio"/> Masculin <input type="radio"/> Féminin
Mot de passe:	<input type="password" value="....."/>
	Minimum 8 caractères
Confirmez le mot de passe:	<input type="password" value="....."/>
	Minimum 8 caractères
Photo:	<input type="button" value="Choisir une image"/>  fond.jpg 600 Ko <input type="button" value="X"/>
<input type="button" value="Valider"/>	

Contenu de la page inscription.php

```

1 <style type="text/css">
2   table {
3     border-collapse: collapse;
4   }
5   table td {
6     padding: 5px 15px;
7     border: 1px solid rgba(32,32,32,0.5);
8   }
9   table tr:nth-child(odd) {
10    background-color: aqua;
11  }
12  .photo {
13    height: 120px;
14    width: auto;
15    border-radius: 7px;
16  }
17 </style>
18 <?php
19 $repertoire="users/";
20 $fichier="";
21 if(isset($_FILES['photo']) && !empty($_FILES['photo']['name'])){
22     $nomfichier= utf8_encode($_FILES['photo']['name']);
23     $tmp=$_FILES['photo']['tmp_name'];
24     if (is_uploaded_file($tmp)) //permet de vérifier si le fichier a été uploadé
25     via http
26     {
27         if(file_exists($repertoire)){ //permet de vérifier que le répertoire
28 existe
29         if(file_exists($repertoire.$nomfichier)){ //on verifie si le fichier
30 existe déjà
31             $nomfichier =getFileUniqueName($nomfichier,$repertoire);
32         }
33         if(move_uploaded_file($tmp,$repertoire.$nomfichier)){
34             echo "Le fichier a été enregistré avec succès";
35             $fichier=$nomfichier;
36         }
37     }
38     }else echo "Le dossier $repertoire n'existe pas";
39 }else $resp="Le fichier n'a pas été uploadé";
40 }else echo "Le fichier photo n'existe pas";
41 ?>
42
43 <center>
44   <h2>Les Informations renseignées sont:</h2>
45   <table>
46     <tr>
47       <td>Nom : </td> <td> <?php echo $_POST["nom"]; ?> </td>
48     </tr>
49     <tr>
50       <td>Tel : </td> <td> <?php echo $_POST["tel"]; ?> </td>
51     </tr>
52     <tr>
53       <td>Email : </td> <td> <?php echo $_POST["email"]; ?> </td>
54     </tr>
55     <tr>
56       <td>Matricule : </td> <td> <?php echo $_POST["matricule"]; ?> </td>
57     </tr>
58     <tr>
59       <td>Résidence : </td> <td> <?php echo $_POST["ville"]; ?> </td>
60     </tr>

```

```

57         <tr>
58             <td>Date de Naissance : </td> <td> <?php echo $_POST["date_naissance"
59 ]; ?> </td>
60         </tr>
61         <tr>
62             <td>Sexe : </td> <td> <?php echo $_POST["sexe"]; ?> </td>
63         </tr>
64         <tr>
65             <td>Mot de passe : </td> <td> <?php echo $_POST["motdepasse"]; ?> </td>
66         </tr>
67         <tr>
68             <td>Photo : </td>
69             <td>
70                 <?php
71                     if(!empty($fichier)){
72                         echo "<img src='$repertoire/$fichier' class='photo' />";
73                     }else{
74                         echo "<img src='$repertoire/user.png' class='photo' />";
75                     }
76                 ?>
77             </td>
78         </tr>
79     </table>
80 </center>
81
82 <?php
83
84 //Il est conseillé de mettre ces fonctions dans un fichier php et de l'inclure ici
85
86 function getFileUniqueName($nomfichier,$repertoire) {
87     while(file_exists($repertoire.$nomfichier)){
88         $stab= getFileExt($nomfichier);
89         $suffixe= rand(1,10000000);
90         $nomfichier=$stab[0].$suffixe.".". $stab[1];
91     }
92     return $nomfichier;
93 }
94 function getFileExt($nomfichier) {
95     $extensionPresumee = explode('.', $nomfichier);
96     $extension= strtolower($extensionPresumee[count($extensionPresumee)-1]);
97     $nom_sans_extension=$extensionPresumee[0];
98     for ($i=1;$i<count($extensionPresumee)-1;$i++){
99         $nom_sans_extension = $nom_sans_extension .".". $extensionPresumee[$i];
100     }
101     return array($nom_sans_extension,$extension);
102 }
103 ?>
104

```

Aperçu dans le navigateur de la page inscription.php

Les Informations renseignées sont:

Nom :	Koffi Jean
Tel :	47859652
Email :	jean@yahoo.fr
Matricule :	UV7845AB
Résidence :	Abidjan
Date de Naissance :	2017-07-20
Sexe :	M
Mot de passe :	tototata
Photo :	

Exercice : Activité d'auto-évaluation 4

VIII

Exercice

Pour sélectionner que des images, on ajoute l'attribut " " à la balise < /> de type

Exercice

Pour sélectionner que des fichiers pdf et png, on ajoute l'attribut =" , " à la balise < /> de type

Exercice

Pour récupérer les fichiers côté serveur, on utilise la variables super-globale

Exercice

Pour tester si un répertoire existe, on utilise la fonction ()

Exercice

Completez le code ci-dessous pou enregistrer le fichier de name phto :

```
<?php
if(isset( ['photo'])&&  ( ['photo']['name'])) {
    $nomfichier= ['photo'][''];
    $tmp= ['photo'][''];
    $dossier="avatars/";
    if (  ($tmp)) //permet de vérifier si le fichier a été uplodé via http
    {
        if( ($ )){ //permet de vérifier que le répertoire existe
            if( ($tmp,  .  )){
                echo "Le fichier a été enregistré avec succès";
            }
        }
    }
}
```

?>