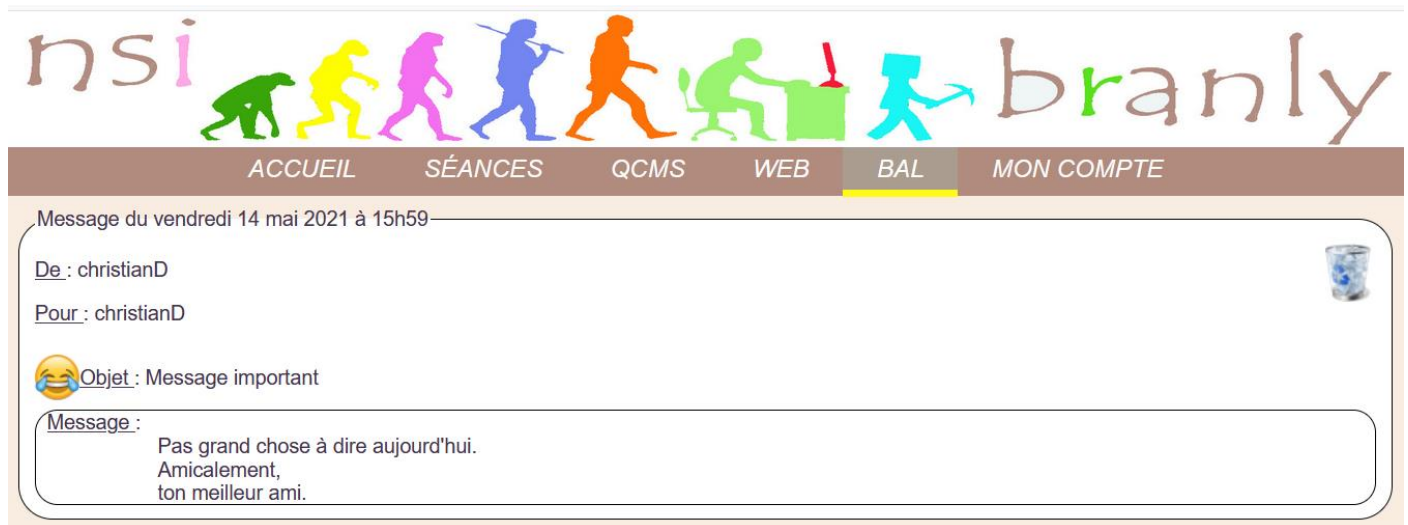


**OBJECTIFS** : L'objectif de ce TP est d'écrire une page web qui comporte un *formulaire*. Il permettra de générer et d'envoyer des messages à vos camarades de classe. Les messages pourront être lus ensuite supprimés, dans l'espace Boite Aux Lettres de chacun, sur le site <https://nsibrantly.fr/>.



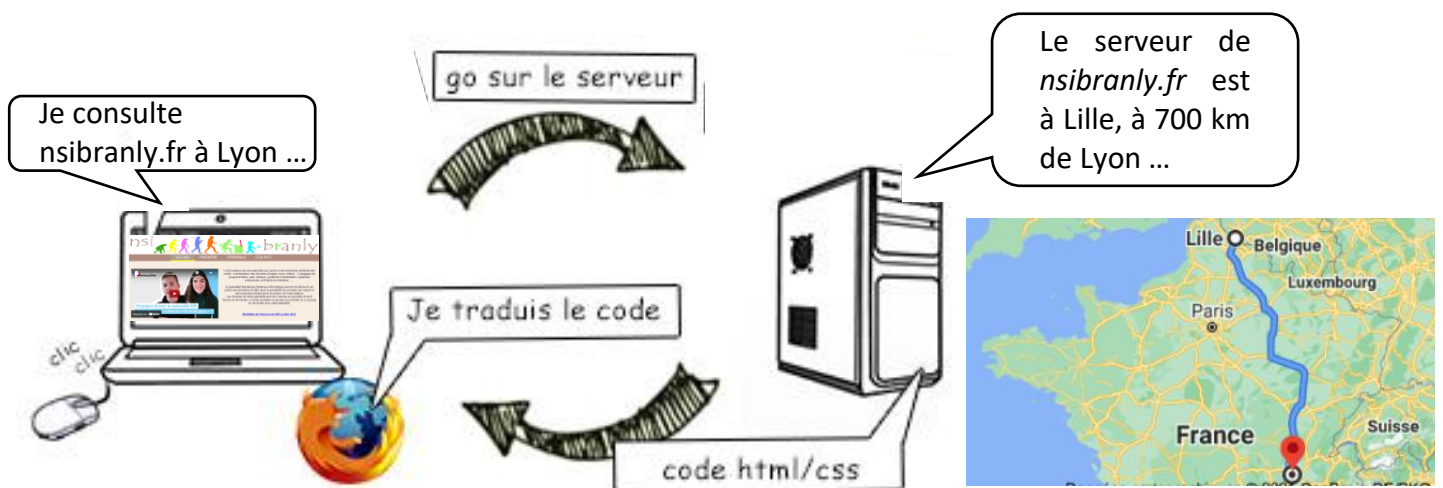
**DOCUMENT A RENDRE** : Ce travail est évalué. Vous aurez à créer un fichier nommé *formulaires.html* qui contiendra le formulaire. Vous le transférez dans votre espace de <https://nsibrantly.fr/>, avec le code web pour le transfert. Vous insérez un lien vers ce fichier *formulaires.html* dans votre page *index.html* et pourrez ainsi tester son fonctionnement.

### 1. Démarrage :

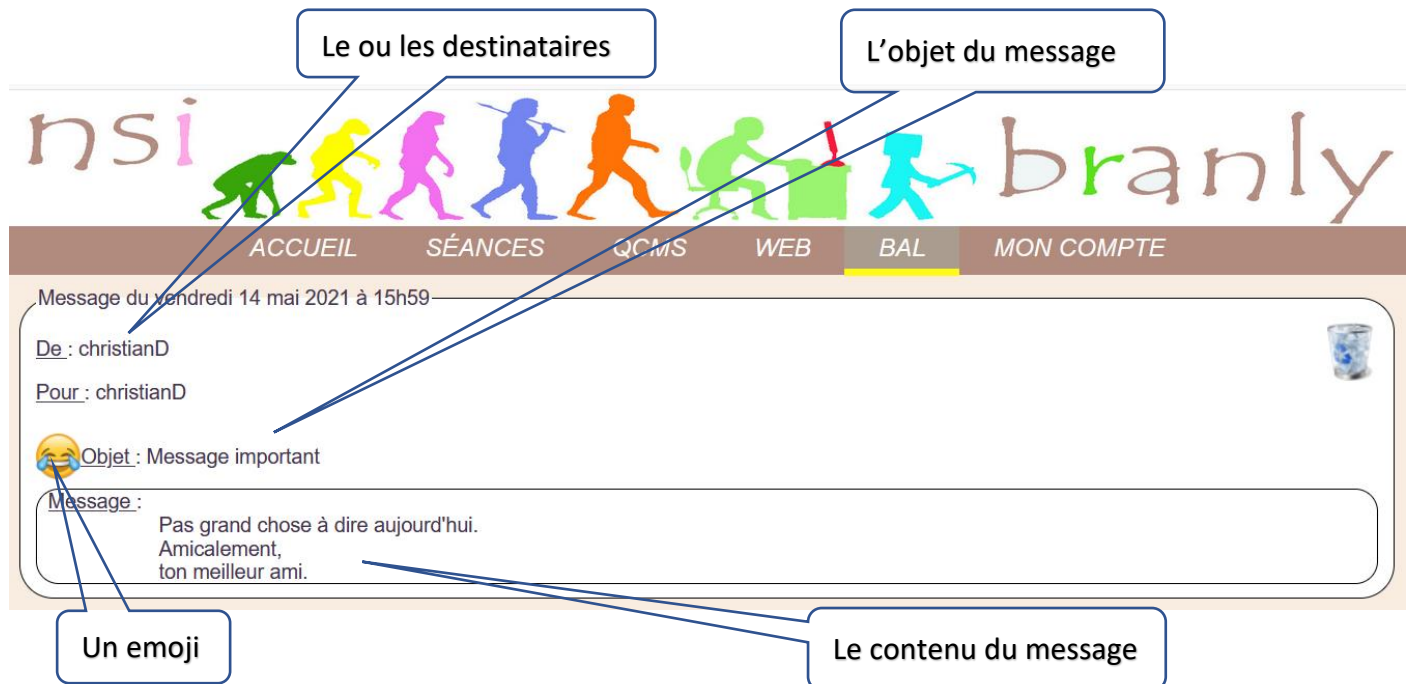
⇒ Créer un nouveau fichier enregistré sous le nom *formulaire.html* et le compléter avec le minimum requis :

Un formulaire permet d'envoyer des données sur le serveur qui héberge le site web :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Formulaires</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="formulaires.css">
  </head>
  <body>
  </body>
</html>
```



Dans ce Tp, le formulaire que l'on va créer va permettre de recueillir les informations suivantes données par le visiteur de la page :



La structure html prévue à cet effet est un block dont le nom de la balise est <form> comme ....

```

<!DOCTYPE html>
<html>
  <head>
    <title>Formulaires</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="formulaires.css">
  </head>
  <body>
    <form>
    </form>
  </body>
</html>
    
```

## 2. Démarrage et méthode GET

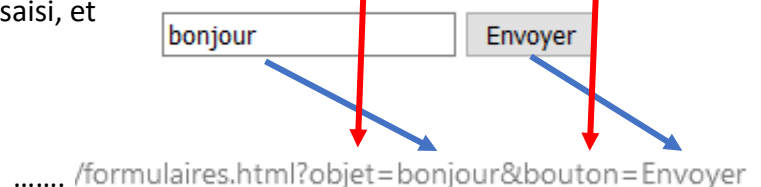
⇒ Compléter le script comme ci-dessous :

```

<body>
  <form action="" method="GET">
    <input type="text" name="objet">
    <input type="submit" value="Envoyer" name="bouton">
  </form>
</body>
    
```

⇒ Dans le navigateur saisir *bonjour* dans le champ de saisi, et cliquer sur *Envoyer*

⇒ Observer ce qui a changé dans la barre d'adresse URL du navigateur :



**Résultat :** En cliquant sur le bouton, on constate que les informations saisies (pour le bouton ce sera la valeur 'envoyer') des 2 éléments du bloc <form> sont stockées dans des variables qui reprennent le nom du **name** des balises et le tout est rajouté à la fin de l'adresse de la page web.

Effectivement, pour transmettre des informations au serveur, on utilise souvent un protocole de requête **https** (Hyper-Text-Transfer-Protocol-Secure) nommé **GET** qui écrit les données à transmettre au serveur, directement dans l'URL.



C'est ce protocole qui est par exemple utilisé par Google pour transmettre les éléments de votre recherche à leur serveur. Par exemple, en effectuant la recherche :

On retrouve dans la barre d'adresse :

⇒ Copiez cette URL dans le presse-papier (*Ctrl C*), rafraichissez la page google.com et collez cette adresse copiée dans la barre d'adresse ..... vous retrouverez le résultat de la recherche précédente.

⇒ Remplacer directement dans l'URL le mot *nsibranly* par *avengers* ..... vous aurez un résultat correspondant à la recherche liée au mot clé avengers ...

### 3. Attributs des 2 balises <input> :

Les 2 éléments input se positionnent sur la même ligne dans le navigateur :

Questions : ⇒ S'agit-il d'éléments *inline* ou d'éléments *block* ?

⇒ La balise <input> est-elle orpheline ?

⇒ Rajouter l'attribut *placeholder* dans la balise <input> de type *text*.

```
<form action="" method="GET">
  <input type="text" name="objet" placeholder="Objet">
  <input type="submit" value="Envoyer" name="bouton">
</form>
```

Question : Quel est l'effet constaté sur la page affichée ?

⇒ Modifier la valeur de l'attribut *value* de la balise input de type *submit*.

Question : Quel est l'effet constaté sur la page affichée ?

Remarque importante : Les attributs *type* et *name* sont les plus importants. Il ne s'agit pas de les modifier. La valeur des attributs *name* sera utilisée sur le serveur pour reconnaître les données reçues :

...../formulaires/formulaires.html?objet=bonjour&bouton=Envoyer

Il est possible de rajouter d'autres attributs. Par exemple :

```
<input type="text" name="objet" placeholder="Objet" minlength="4" maxlength="8" size="10" required>
```

Mais pour notre messagerie, les attributs supplémentaires (*minlength*, *required*) ne sont pas le bienvenu car un message peut bien être envoyé avec un *objet* vide. L'attribut *maxlength* serait utile pour limiter la longueur de la chaîne de caractère saisie.

#### 4. Balise <textarea> :

La balise <input type='text'> permet de saisir une phrase. Elle nous permettra de saisir l'objet du message que l'on souhaite envoyer. Au contraire, la balise <textarea> permet de saisir un texte de plusieurs lignes. Elle nous permettra de saisir le texte du message à envoyer.

⇒ Rajouter une balise de ce type dans votre script :

```
<form action="" method="GET">
  <input type="text" name="objet" placeholder="Objet">
  <input type="submit" value="Envoyer" name="bouton">
  <textarea name="message"></textarea>
</form>
```

Ce qui provoquera l'affichage suivant dans votre navigateur :

Questions : ⇒ <textarea> est-il un élément *inline* ou un élément *block* ?

⇒ La balise < textarea > est-elle orpheline ?

⇒ Saisir le texte suivant dans le champ de saisie de cette balise et cliquer sur le bouton :

Bonjour,  
ma grand-mère m'a dit que le mois de mai était le plus beau de l'année.  
A-t-elle toujours raison ?

Les 3 données envoyées qui ont comme name respectifs : *objet*, *bouton* et *message*, s'écrivent-elles dans l'URL ? :

```
/formulaires.html?objet=&bouton=Envoyer&message=Bonjour%2C%0D%0Ama+grand-mère+m'a+dit+que+le+mois+de+mai+était+le+plus+beau+de+l'année.+%0D%0AA-t-elle+toujours+raison+%3F%0D%0A
```

⇒ Sur le navigateur, pouvez-vous agrandir, la zone texte du *textarea* en agissant avec la souris sur le coin inférieur droit ?

⇒ Si le texte saisi est trop grand, une barre de défilement se met-elle en place ?

On peut rajouter des attributs supplémentaires à cette balise. Pour notre application, on limitera la saisie à 1000 caractères et on imposera la taille de la zone de saisie (modifiable ensuite avec le CSS).

```
<textarea name="message" rows="5" cols="40" maxlength="1000" placeholder="Message"></textarea>
```

5. Balise <select> :

La balise <select> permet à l'internaute de choisir un item dans une liste prédéfinie. On l'utilise ici pour pouvoir choisir le destinataire du message que l'on souhaite envoyer. Compléter votre script de la façon suivante :

```
<form action="" method="GET">
  <input type="text" name="objet" placeholder="Objet">
  <input type="submit" value="Envoyer" name="bouton">
  <textarea name="message" rows="5" cols="40" maxlength="1000" placeholder="Message"></textarea>
  <select name="destinataire[]" multiple>
    <option value="julienA">Le Roi Jeptronia</option>
    <option value="wassimM">iamthewassim</option>
    <option value="thomasT">Tototra</option>
    <option value="kaisseB">JusDeCactus</option>
    <option value="arthurL">Sirukapi</option>
    <option value="matisV">Bob l'amateur</option>
  </select>
</form>
```

Dans votre navigateur, il s'affichera la page :

Comme l'attribut *multiple* a été rajouté dans la balise <select>, il est possible de sélectionner plusieurs items en même temps. A cet effet, le *name* attribué à cette balise devient une liste : `name="destinataire[]"`

Questions : ⇒ <select> est-il un élément *inline* ou un élément *block* ?

⇒ La balise < select > est-elle orpheline ?

⇒ Saisir les 2 premiers items de la liste et cliquer sur le bouton.

Dans l'URL, sont écrites les valeurs des 4 données envoyées. Parmi celles-ci, combien ont une valeur nulle ?

```
.../formulaires.html?objet=&bouton=Envoyer&message=&destinataire[]=julienA&destinataire[]=wassimM
```

⇒ Est-ce que c'est la valeur de l'attribut *value* qui est envoyée ?

⇒ Rajouter l'attribut *size* qui limite le nombre d'items visibles et *required* qui n'accepte pas l'envoi si aucun item a été sélectionné (on peut envoyer un message vide, mais difficile d'en envoyer s'il n'y a pas de destinataire ...)

```
<select name="destinataire[]" size="3" multiple required>
```

⇒ Compléter la liste des destinataires. La valeur de l'attribut *value* de vos camarades est :

|         |          |          |          |         |
|---------|----------|----------|----------|---------|
| arthurL | gabrielD | loricA   | rayanA   | thomasT |
| claireA | irisD    | maelD    | rayanC   | timeoM  |
| djessyK | julienA  | matisV   | remiC    | wassimM |
| elisaD  | kaisseB  | mattisD  | robinC   |         |
| evanF   | leoM     | mickaelT | robinS   |         |
| faamanA | loicO    | noel     | thibaudR |         |

## 6. Balise `<input type= 'radio'>` :

La balise `<input>` de type `'radio'` permet de réaliser un choix en cochant des cases. On se sert de cette structure pour sélectionner l'image de l'emoji que l'on associera au message envoyé. Compléter votre script de la façon suivante :

```
<form action="" method="GET">
  <input type="text" name="objet" placeholder="Objet">
  <input type="submit" value="Envoyer" name="bouton">
  <textarea name="message" rows="5" cols="40" maxlength="1000" placeholder="Message"></textarea>
  <select name="destinataire[]" size="3" multiple required>...
</select>
  <input type="radio" name="emoji" value="amour.png">
  <input type="radio" name="emoji" value="normal.png" >
  <input type="radio" name="emoji" value="hilare.png" >
</form>
```

Télécharger les 3 images ( <https://nsibranly.fr> ) et les enregistrer dans le répertoire de votre script. Dans votre navigateur, il s'affichera la page :

L'attribut *name* de ces 3 balises `<input>` est le même. Il est égal à `'emoji'`. Seul 1 seul emoji pourra être sélectionné. Ainsi la valeur de la donnée envoyée sera repérée par le *name* `'emoji'` et sera égale ici soit à `'amour'`, soit à `'normal'`, soit à `'hilare'`.

Questions : ⇒ `<input type='radio'>` est-il un élément *inline* ou un élément *block* ?

⇒ La balise `<input type='radio'>` est-elle orpheline ?

⇒ Sans rien cocher, rien sélectionner dans la liste, rien écrire, cliquer sur le bouton. La barre d'URL devient :

`/formulaires.html?objet=&bouton=Envoyer&message=`

Cela signifie-t-il que les données envoyées sous les *name* `objet` et `message` sont des strings vides et que celles envoyées sous les *name* `destinataire[]` et `emoji` n'ont même pas été générées ?

⇒ Cocher    , cliquer sur le bouton et observer l'URL .....

La donnée envoyée sous le *name* `emoji` a-t-elle été générée ? Quelle est sa valeur ?

`/formulaires.html?objet=&bouton=Envoyer&message=&emoji=normal`

Améliorations : On rajoute l'attribut *checked* sur le 1<sup>er</sup> `<input>` afin de sélectionner par défaut un emoji. Sur smartphone, cocher précisément avec ses doigts la bonne zone n'est pas évident. Afin d'étendre cette zone à l'image entière, on encapsule l'image dans une balise `<label>` que l'on lie avec la balise `<input type='radio'>` par un *id* qui prend une valeur commune :

```
<input type="radio" name="emoji" value="amour.png" id="emoji_1" checked ><label for="emoji_1"></label>
<input type="radio" name="emoji" value="normal.png" id="emoji_2" ><label for="emoji_2"></label>
<input type="radio" name="emoji" value="hilare.png" id="emoji_3" ><label for="emoji_3"></label>
```

⇒ Tester cette modification et l'étendre aux 2 autres emoji.

## 7. Premier test de la page :

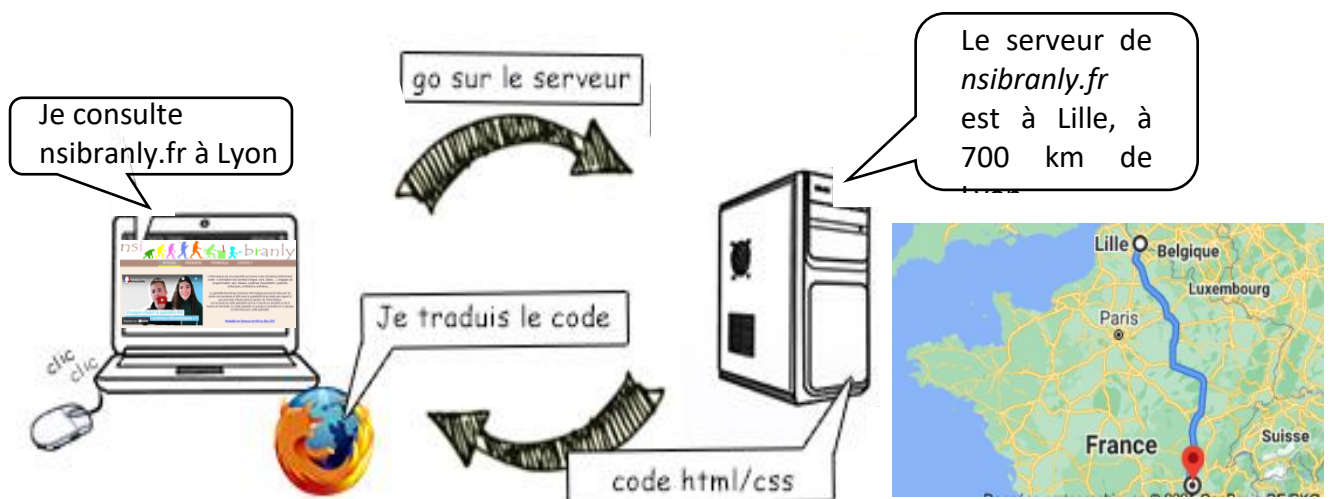
Votre script ne vous permet pas d'afficher une page sensationnelle dans votre navigateur. Il a néanmoins le mérite d'être fonctionnel. On va le tester dès à présent en le transférant ici dans votre espace sur le site <https://nsibranly.fr> .... mais pas si vite, il y a encore quelques manipulations à faire ...

### 7.1. Comment envoie-t-on réellement les données sur le serveur :

Le formulaire mis au point permet simplement de recueillir les données que l'expéditeur veut envoyer à ses camarades de classe. Ces données sont :

- L'objet du message contenu dans une variable dont le nom est : *objet*
- Le texte du message contenu dans une variable dont le nom est : *message*
- Le ou les destinataires contenus dans une liste dont le nom est : *destinataire[]*
- Le nom d'un emoji contenu dans une variable dont le nom est : *emoji*

Ces données vont être numérisées **par le navigateur** en suivant une méthode de requête **GET** ou **POST**. Ces 2 méthodes font parties d'un protocole nommé « *https* » (**H**yper **T**ext **T**ransfer **P**rotocol **S**ecure) inventé dans les années 1990. Il permet le transfert de pages web entre le serveur et le client (l'internaute).



Avec la méthode **GET**, les données saisies par l'internaute sont écrites dans l'URL et sont transférées ensuite avec cette URL, lorsque l'on clique sur le bouton `<input type='submit'>`.

C'était bien pratique jusqu'à présent dans ce Tp pour avoir une idée du contenu des données envoyées. Cette méthode GET présente néanmoins 2 inconvénients majeurs :

- Les données sont visibles dans l'URL, ce qui est problématique par exemple lorsque l'on transfère un mot de passe, ....
- Les données transmises doivent être des strings (pas de transfert possible de fichiers binaires : images, vidéos) et la taille des données est limitée.

La seconde méthode de transfert que propose le protocole **https**, s'appelle **POST** et ne présente pas ces inconvénients. On l'utilisera ici pour réaliser notre messagerie.

## 7.2. Oui, mais concrètement que fait-on ? :

1 ⇒ Modifier le nom de la méthode dans la balise <form> en remplaçant GET par **POST** ..

2 ⇒ Ajouter dans la balise <form> , le nom du fichier qui sera exécuté sur le serveur lorsque les données que l'on envoie arrivent ....

```
<form action="../formulaires.php" method="POST">
```

Après réception des données sur le serveur, ici le fichier nommé *formulaires.php* sera exécuté. Il se trouve dans le répertoire parent de celui qui vous est attribué sur le serveur, pour pouvoir publier votre page web (d'où le ../) . Ce fichier est en langage *php* que les serveurs savent interpréter (mais pas le navigateur de votre ordinateur). Il est déjà écrit et permet de lire les données envoyées et de les écrire dans une Base De Données (bdd) qui se trouve aussi sur le serveur à Lille. Cette bdd est une sorte de gros fichier qui permet de conserver tous les messages envoyés par chacun. Les données reçues sont reconnues par rapport à leur nom qui est celui de l'attribut **name** écrit dans chaque balise. C'est pour cela qu'il faut bien faire attention à cet attribut qui est très important.

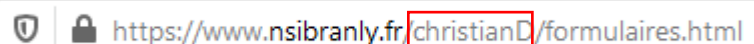
D'autres scripts, écrits aussi en php, permettent ensuite de récupérer dans cette bdd les messages qui vous sont destinés, pour les écrire en html dans la page web de votre bal, sur nsibrantly.fr . Ces choses là seront vues en classe de terminale .....

Le script complet est alors :

```
<form action="../formulaires.php" method="POST">
  <input type="text" name="objet" placeholder="Objet">
  <input type="submit" value="Envoyer" name="bouton">
  <textarea name="message" rows="5" cols="40" maxlength="1000" placeholder="Message"></textarea>
  <select name="destinataire[]" size="3" multiple required>...
</select>
  <input type="radio" name="emoji" value="amour.png" id="emoji_1" checked ><label for="emoji_1"></label>
  <input type="radio" name="emoji" value="normal.png" id="emoji_2" ><label for="emoji_2"></label>
  <input type="radio" name="emoji" value="hilare.png" id="emoji_3" ><label for="emoji_3"></label>
</form>
```

3 ⇒ Sur le site nsibrantly.fr, se loguer et transférer avec le code *web* votre fichier *formulaires.html* et les fichiers images *amour.png*, *normal.png*, *hilare.png* .

4 ⇒ Appeler votre fichier *formulaires.html* en écrivant dans l'URL :



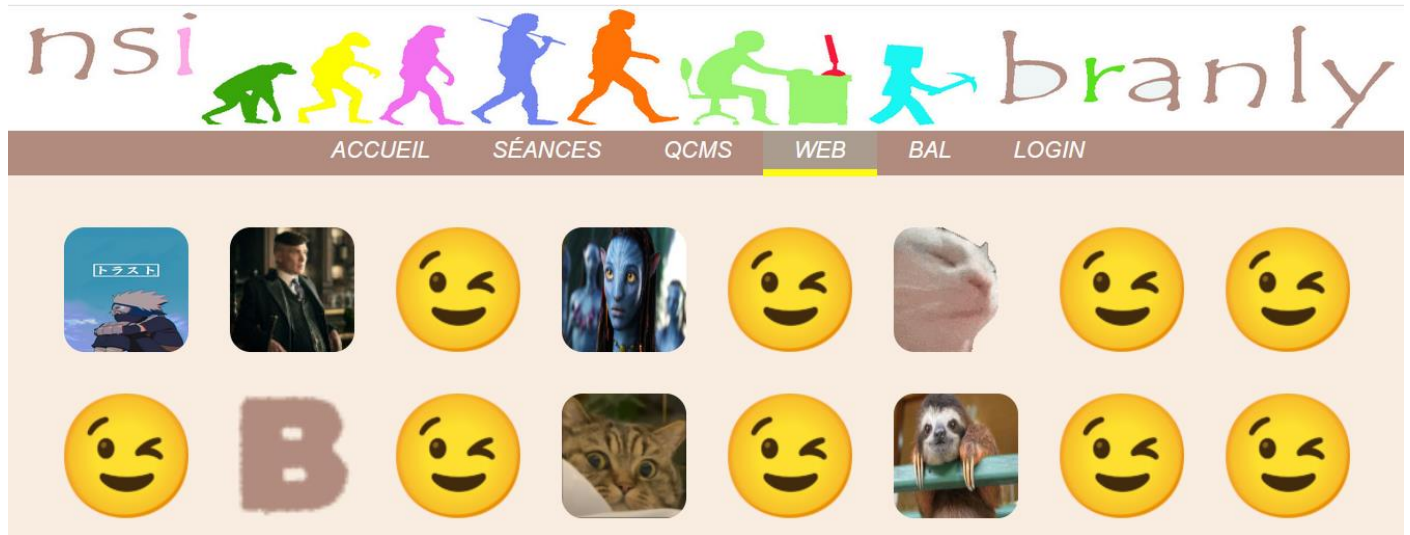
Votre répertoire s'appelle :  
prénom + 1<sup>ère</sup> lettre de votre nom de famille en Majuscule




5 ⇒ Vous pouvez envoyer un message à vous-même pour commencer. Les messages sont à consulter dans l'onglet bal de *nsibranly.fr*.

8. Intégration de votre page formulaires.html dans votre espace :

L'onglet web de *nsibranly.fr* vous permet de publier du contenu web. En cliquant sur l'icone qui vous est dédiée, le fichier *index.html* se trouvant dans votre zone est affichée par le navigateur :



Par exemple, sur ma page : , le fichier *index.html* correspond à la page web réalisée dans le dernier cours (site Avengers) :



J'ai modifié les 2 premiers items du bloc nav pour y mettre un lien vers mon fichier *formulaires.html* et vers ma bal :

```

<body>
  <header>
    <h1>Les Avengers à Koh-Lanta</h1>
    <div><img src='facebook.png'><img src='twitter.png'><img src='snap.png'></div>
  </header>
  <nav>
    <ul>
      <li><a href='formulaires.html'>Messagerie</a></li>
      <li><a href='../bal.php'>Bal</a></li>
      <li><a href=''>Hulk</a></li>
      <li><a href=''>C.America</a></li>
    </ul>
  </nav>
  <section>
    <article>
      <p>
        Avengers ou Les Avengers : Le film au Québec et au Nouveau-Brunswick (M
      </p>
    </article>
  </section>

```

⇒ Réaliser cette modification afin de pouvoir accéder facilement à votre formulaire.

### 9. Mise en forme de votre formulaire en ajoutant du CSS :

L'objectif de cette partie est d'obtenir grâce au Css, un formulaire digne de ce nom.

**On retravaillera à présent en mode local à partir du fichier *formulaires.html* qui se trouve sur votre ordinateur. Ce n'est qu'à la fin de ce travail que l'on transfèrera les fichiers modifiés sur le serveur.**

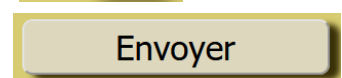
L'objectif est de créer le fichier *formulaires.css* qui permette d'arriver à obtenir un formulaire qui part de :

A screenshot of a basic HTML form. It features a text input field labeled "Message", a dropdown menu with options "Le Roi Jepronia", "iamthewassim", and "Tototra", and three emoji buttons: a smiling face with hearts, a tongue sticking out, and a laughing face with tears.

pour arriver **par exemple** à :

A screenshot of a styled form titled "Ma messagerie". The form has a yellow background and rounded corners. It includes a "Destinataire" dropdown menu with options "iamthewassim", "Tototra", and "JusDeCactus". Below it is an "Objet" input field. The "Humeur" section contains three emoji buttons with a blue highlight on the first one. The "Message" section has a text input field. At the bottom is a large "Envoyer" button.

avec un effet hover d'ombrage si on passe sur les emoji ou le bouton :



Pour cela la première chose à faire est d'enfermer les éléments html du formulaire qui sont tous *inline*, dans des blocs <div>. On vous propose de partir du code html qui suit, dans lequel cette opération a été réalisée.

⇒ Créer un fichier nommé *formulaires.css* qui comporte la partie css de la page du formulaire. Afin d'être organisé dans la construction de ce fichier, on propose d'y rajouter d'entrée les commentaires donnés ci-après :

```
<h1>Ma messagerie</h1>
<form action="../../formulaires.php" method="POST">
  <div class="destinataire">
    <p>Destinataire : </p>
    <div>
      <select name="destinataire[]" size="3" multiple required>
        <option value="julienA">Le Roi Jeptonia</option>
        <option value="wassimM">iamthewassim</option>
        <option value="thomasT">Tototra</option>
        <option value="kaisseB">JusDeCactus</option>
        <option value="arthurL">Sirukapi</option>
        <option value="matisV">Bob l'amateur</option>
        <option value="christianD">Prof</option>
      </select>
    </div>
  </div>
  <div class="objet">
    <p>Objet : </p>
    <input type="text" name="objet" placeholder="Objet : ">
  </div>
  <div class="humeur">
    <p>Humeur : </p>
    <div class="emoji">
      <div>
        <input type="radio" name="emoji" value="amour.png" id="emoji_1" checked ><label for="emoji_1"></label>
      </div>
      <div>
        <input type="radio" name="emoji" value="normal.png" id="emoji_2" ><label for="emoji_2"></label>
      </div>
      <div>
        <input type="radio" name="emoji" value="hilare.png" id="emoji_3" ><label for="emoji_3"></label>
      </div>
    </div>
  </div>
  <div class="message">
    <p>Message : </p>
    <textarea name="message" rows="5" cols="40" maxlength="1000" placeholder="Message : "></textarea>
  </div>
  <input type="submit" value="Envoyer" name="bouton">
</form>
```

```
body {
}
h1{
}
/***** Commun à tous les blocs *****/
.destinataire , .objet , .humeur , .message {
}
/***** Gestion bloc destinataire *****/
/***** Gestion bloc objet *****/
/***** Gestion bloc emoji *****/
/***** Gestion bloc message *****/
/***** Gestion bouton type submit *****/
input[type="submit"]{
}
input[type="submit"]:hover{
}
}
```

### Infos Css :

- Les balises `<input>` sont nombreuses ... Pour ne sélectionner que celles d'un certain type, on utilise la syntaxe :  
*input[type= 'submit'] {}* .... par exemple pour la sélection des boutons.
- Pour créer un ombrage :  
*box-shadow: 10px 5px 5px red* par exemple
- Votre page doit se comporter correctement pour tout type d'écran, y compris sur smartphone. Penser à vérifier le bon fonctionnement du script, en réduisant la largeur de la fenêtre du navigateur.

⇒ Lorsque le rendu de la page *formulaires.css* vous semble correct, vous transfèrerez les 2 fichiers : *formulaires.html* et *formulaires.css* sur [nsibrantly.fr](https://nsibrantly.fr) afin de mettre en ligne votre travail.