

Info 16 - Html - Formulaires

OBJECTIFS : On se propose dans ce Tp d'écrire une page web qui comporte un **formulaire** afin de créer une sorte de tchat dans le groupe classe. A travers cette activité, les objectifs sont de découvrir les éléments html dédiés aux formulaires et de s'intéresser au protocole de communication http entre l'internaute et le serveur web qui héberge le site consulté (dialogue client-serveur).

DOCUMENT A RENDRE : Ce travail est évalué de 2 façons :

- Vous aurez à créer un fichier nommé *tchat.html* qui sera lié à votre espace web sur *nsibranly.fr* .
- Vous avez à répondre à des questions écrites dans un fichier texte à uploader sur *nsibranly* avec le code tp15.

1. DEMARRAGE ET CREATION DES FICHIERS HTML ET CSS :

⇒ Télécharger le zip contenant les fichiers nécessaires à la réalisation de ce Tp. Extraire et enregistrer les fichiers dans un répertoire nommé « *tchat* » par exemple.

⇒ Créer dans ce même répertoire, un nouveau fichier enregistré sous le nom *tchat.html*.

⇒ Ecrire un ! et valider par *Entrée* afin d'obtenir la structure html du fichier.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>

</body>
</html>
```

⇒ Lier le fichier *tchat.css* qui a été téléchargé, à ce fichier html, en rajoutant une balise orpheline `<link>` dans le bloc `<head>` des métas.

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Tchat</title>
  <link rel="stylesheet" href="tchat.css">
  <link rel="icon" type="img/png" href="tchat.png">
</head>
```

⇒ Modifier le titre de la page qui apparait dans l'onglet du navigateur et ajouter une autre balise `<link>` pour faire apparaître une icône à côté du titre. Le fichier image qu'on vous propose pour cette icône, est celui de l'image ci-contre (il est dans le répertoire zip téléchargé en début de Tp). Vous pourrez prendre une autre image en ayant pris soin de réduire sa largeur à environ 50 px avec le logiciel *Gimp*.




⇒ Dans l'explorateur de *Visual Studio Code*, par un clic droit sur le nom du fichier *tchat.html*, copier le chemin du fichier. Ouvrir un navigateur (utiliser **Firefox**) et coller ce lien dans la barre d'Url afin de visualiser la page qui pour l'instant est une page blanche.

2. QUE VA-T-ON FAIRE ? :

Pour être capable d'envoyer des informations vers le serveur qui héberge le site, on utilise un bloc *html* dédié aux formulaires et balisé par `<form> ... </form>`. A l'intérieur de ce bloc, on placera des éléments *html* qui pourront recueillir les informations du message que l'on postera. Le serveur se débrouillera ensuite pour rajouter en temps réel, ce message au tchat. On voit tout cela dans la suite

3. CREATION DU BLOC <FORM> - PREMIER ENVOI AVEC LA METHODE GET:

⇒ Insérer un bloc `<form>` dans le bloc `<body>` :



Un conseil, utilise l'autocomplétions de VSC : on écrit « for ... » jusqu'à ce que le nom de la balise soit reconnu On valide ensuite en tapant sur la touche Entrée

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

⇒ Visualiser le résultat en affichant la page dans le navigateur elle reste blanche.

On commence par insérer dans le bloc `<form>` uniquement le champ de saisi qui recevra le texte décrivant l'objet du message et un bouton qui permettra de créer l'évènement qui enverra le texte saisi vers le serveur :

⇒ Insérer les 2 éléments `<input>` ci-dessous, dans le bloc `<form>` :

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

Un champ de saisi recueillant un texte court pour définir l'objet du message : balise orpheline `<input>` avec l'attribut `type= 'text'`

Un bouton : balise orpheline `<input>` avec l'attribut `type= 'submit'`

⇒ Visualiser le résultat en affichant la page dans le navigateur.

⇒ Observer l'adresse dans la barre d'URL du navigateur. On y retrouve le chemin local qui conduit au fichier `tchat.html`. Cette adresse commence par « `file:/` ». Elle se termine par le nom du fichier visualisé « `tchat.html` » : `file:///..... /tchat.html`

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

⇒ Observer ce qui a changé dans la barre d'adresse URL On constate que derrière le nom du fichier « `tchat.html` », le navigateur a rajouté :

`file:///..... /tchat.html?objet=bonjour&bouton=Envoyer`

Nom fichier visualisé

?

Une variable nommée `objet` prend comme valeur 'bonjour'

& qui veut dire et

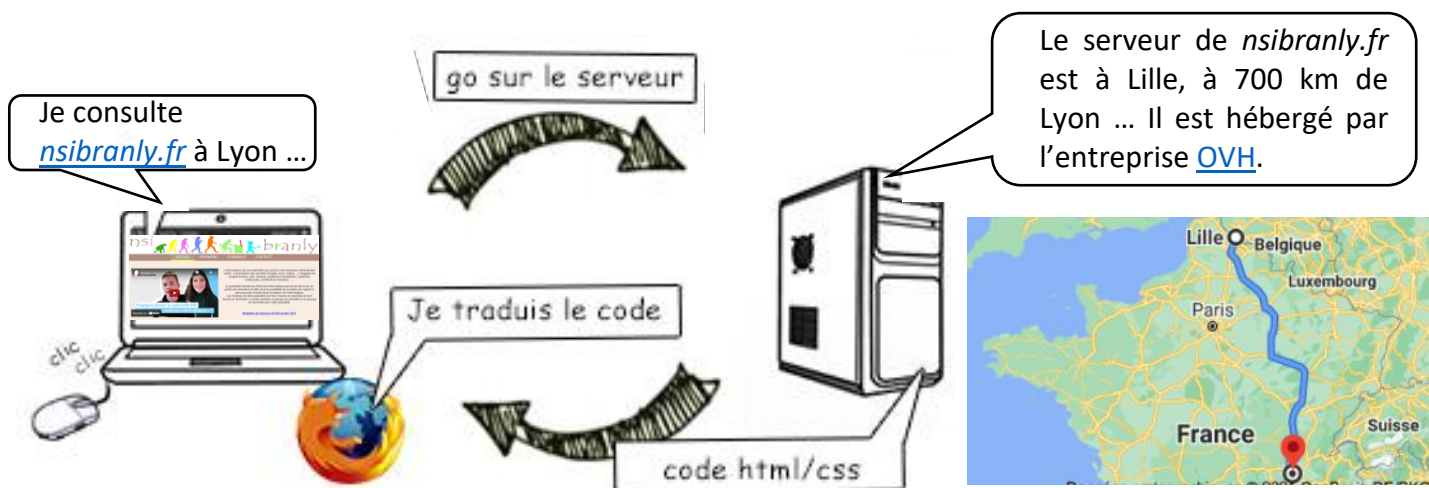
Une variable nommée `bouton` prend comme valeur 'Envoyer'

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

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 de l'attribut **name** des balises. Le tout est rajouté à la fin de l'adresse de la page web, après un point d'interrogation ? :

`file:///..... /tchat.html?objet=bonjour&bouton=Envoyer`

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



Actuellement, vous utilisez le navigateur pour lire le fichier *tchat.html* qui se trouve sur le réseau du lycée Branly. Dans la suite, on se propose d'uploader ce fichier « *tchat.html* » sur le serveur de *nsibranly.fr* afin de pouvoir cette fois-ci le lire, par l'intermédiaire d'une communication sur le réseau internet, entre Lyon et Lille (distance de 700 km).

4. INSERTION DE LA PAGE TCHAT.HTML DANS MON ESPACE WEB SUR NSIBRANLY.FR

Normalement vous avez dû conserver sur le réseau de Branly, un répertoire identique à celui de votre espace Web sur *nsibranly.fr*. Si vous ne retrouvez pas ce répertoire, pas de problème, il suffit, de copier votre fichier *index.html* qui est actuellement en ligne sur votre espace Web personnel (afficher la page dans le navigateur. Avec un clic droit, choisir « code source de la page ». Copier le code html qui s'affiche dans le presse-papier et le coller dans un nouveau fichier que vous nommerez *index.html*)

⇒ Editer ce fichier *index.html* pour ajouter un lien vers *tchat.html* . Par exemple :

```
<a href="tchat.html">Tp tchat</a>
```

⇒ Sur *nsibranly.fr*, uploader les fichiers *index.html* , *tchat.html* et *tchat.png* dans votre espace Web personnel, en utilisant le code **web**.

5. INSERTION DE LA PAGE TCHAT.HTML DANS MON ESPACE WEB SUR NSIBRANLY.FR

⇒ Aller sur votre page web de *nsibranly.fr*, puis sur la page *tchat.html* en utilisant le lien inséré précédemment.

⇒ Observer l'adresse dans la barre d'Url `https://www.nsibranly.fr/membres/[redacted]/tchat.html`

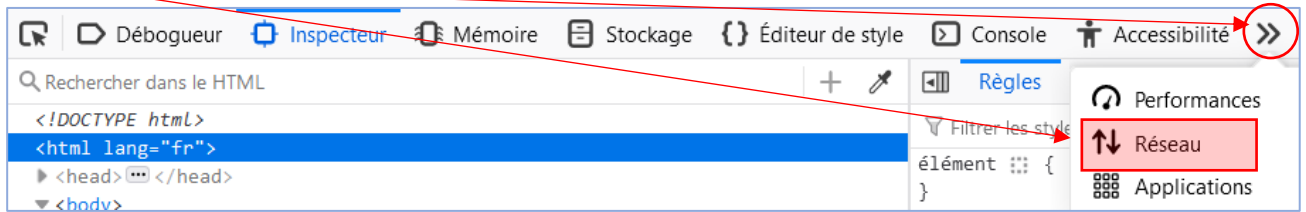
⇒ Saisir à nouveau le mot **bonjour** dans le champ de saisi, et cliquer sur le bouton et observer à nouveau l'Url :

`https://www.nsibranly.fr/membres/[redacted]/tchat.html?objet=bonjour&bouton=Envoyer`

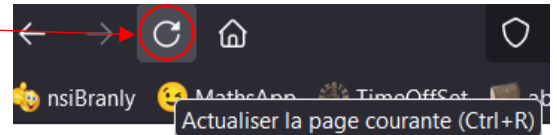
Le début de l'adresse est à présent celle d'un serveur web. Pour la fin de l'adresse, les conclusions sont identiques à celles évoquées lors de la même manipulation, réalisée en mode « local ».

On utilise les outils du développeur pour approfondir certains points :

⇒ Faire un clic droit sur la page Web et choisir « inspecter ». Parmi les onglets proposés, choisir Réseau (ou Network).



⇒ Recharger (on dit « actualiser » aussi) la page en tapant sur la touche F5 ou en utilisant l'icône :



Vous obtenez alors l'affichage suivant qui indique que pour obtenir la page tchat.html qui se trouve à 700 km de Lyon, le navigateur a envoyé 3 requêtes. On commente cela dans la suite :

État	Métho...	Domaine	Fichier	Initiateur	Type	Transfert	Tai...	0 ms	80 ms	160 ms	240 ms
200	GET	www.nsiibra...	tchat.html?objet=bonjour&bouton=Envoyer	document	html	478 o	4...	54 ms			
404	GET	www.nsiibra...	tchat.css	stylesheet	html	350 o	1...	26 ms			
200	GET	www.nsiibra...	tchat.png	FaviconLoa...	png	8,48 Ko	8...	26 ms			

2 des requêtes ont abouti. Le serveur a renvoyé un état 200.
1 requête a échoué. Le serveur a renvoyé un état 404 erreur

Le protocole utilisé pour le dialogue est nommé « GET »

Type de ressource demandée et réceptionnée

Chronologie : la 1^{ère} requête demande le fichier html. Une fois réceptionné, le navigateur lit ce fichier et s'aperçoit qu'il faut encore demander d'autres ressources ... le tout a pris ici 0,2 seconde.

Question 1. : Faire une recherche google avec les mots clés : « http 200 » . Que signifie l'état 200 affiché ci-dessus ?

Question 2. : Faire une autre recherche avec les mots clés : « http 404 » . Que signifie l'état 404 affiché ci-dessus ?

⇒ Sélectionner par un click la 1^{ère} requête et visualiser l'onglet « En-têtes » de celle-ci :

On décode un peu ces informations :

3 requêtes | 8,85 Ko / 828 o transférés | Terminé en : 323 ms | DOMContentLo

En-têtes | Cookies | Requête | Réponse | Délais | Sécurité

En-têtes de la réponse (203 o) | Texte brut

- État: 200
- Version: HTTP/2
- Transfert: 478 o (taille 453 o)
- Politique de référent: strict-origin-when-cross-origin
- Priorité de la requête: Highest
- Résolution DNS: Système

En-têtes de la requête (655 o) | Texte brut

- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
- Accept-Encoding: gzip, deflate, br
- Accept-Language: fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3
- Cache-Control: no-cache
- Connection: keep-alive
- Cookie: PHPSESSID=114690d87a350af6176b075084e5d226
- Host: www.nsibranly.fr
- Pragma: no-cache
- Referer: https://www.nsibranly.fr/membres/christianD/
- Sec-Fetch-Dest: document
- Sec-Fetch-Mode: navigate
- Sec-Fetch-Site: same-origin
- Sec-Fetch-User: ?1
- TE: trailers
- Upgrade-Insecure-Requests: 1
- User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0

En écrivant cette Url le navigateur demande (méthode « GET ») la ressource *tchat.html* au serveur qui a comme nom de domaine *nsibranly.fr*.

Les variables écrites après le ? en fin d'Url pourront être lues et utilisées par le serveur.

Informations données en réponse par le serveur. Elles accompagnent la ressource « *tchat.html* » qui est arrivée pour la seconde fois sur le navigateur (seconde fois car la page se recharge après l'envoi du formulaire)

Informations données par le navigateur. Elles accompagnent la requête générée par le navigateur qui demande la ressource « *tchat.html* » au serveur : requête

GET https://www.nsibranly.fr/membres/christianD/tchat.html?objet=bonjour&bouton=Envoyer

⇒ En étant toujours sur la 1^{ère} requête, visualiser l'onglet « Cookies » :

En-têtes | Cookies | Requête | Réponse | Délais | Sécurité

Filtrer les cookies

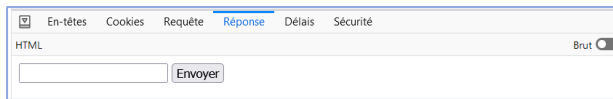
Cookies de la requête

PHPSESSID: "114690d87a350af6176b075084e5d226"

Identifiant de session composé de 32 chiffres en hexadécimal. Il est généré par le serveur au premier échange de ressource. Il est ensuite associé à toutes les requêtes pour que le serveur puisse identifier le client.

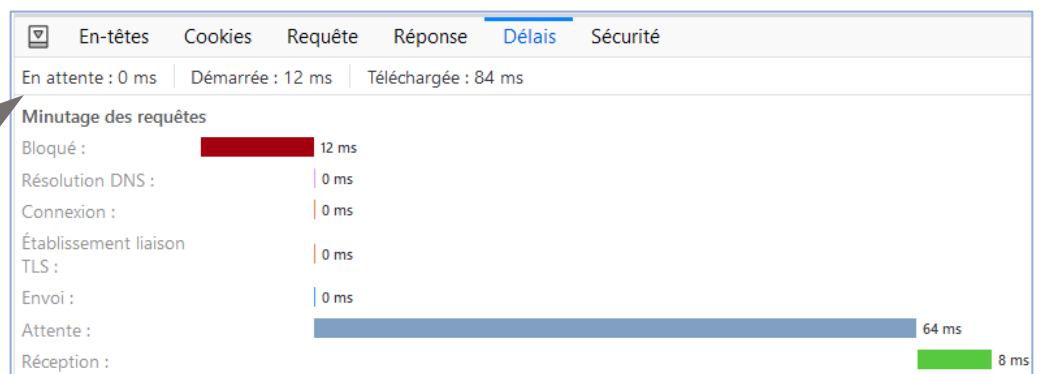
Cet identifiant est temporaire. Il est supprimé à la fermeture de la fenêtre du navigateur.

⇒ En étant toujours sur la 1^{ère} requête, visualiser l'onglet « Réponse » :



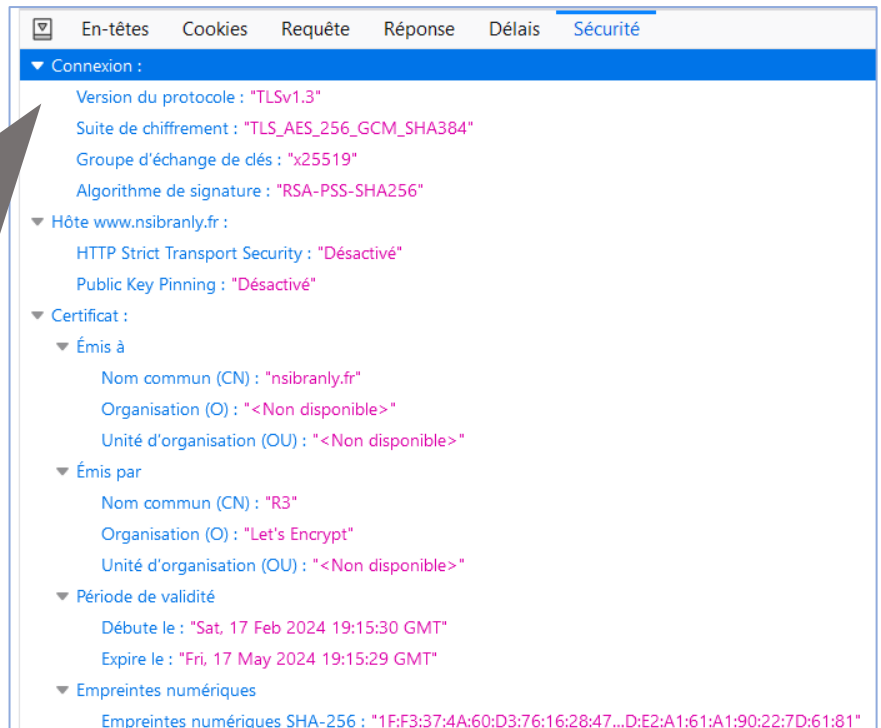
La requête affichée ici est celle qui revient du serveur. Hormis les en-têtes de demande de ressource, de retour de ressource, on trouve ici le contenu de la ressource qui a été demandée, en version BRUT (code html) ou interprétée (visuel sur le navigateur)

⇒ En étant toujours sur la 1^{ère} requête, visualiser l'onglet « Délai » :



On a le détail du timing. Intéressant pour identifier d'éventuels lenteurs sur une page à optimiser.

⇒ Enfin, visualiser l'onglet « Sécurité » :



Les données qui sont échangées entre le navigateur de votre ordinateur du lycée Branly et le serveur de nsibrany.fr sur Lille, sont cryptées car l'adresse du domaine commence par https://.

Au premier échange entre navigateur et serveur, une clé de cryptage - décryptage commune est définie par les 2 parties. Pour que cette clé ne soit connue que par ces 2 parties, une procédure complexe est mise en œuvre. Elle est basée sur des l'existence de clés privées / publics et sur des certificats d'authentification.

On trouve ici des informations sur ces certificats et sur la méthode de cryptage utilisée.

Rechercher les réponses aux questions suivantes sur Google :

Question 3. : C'est quoi *http* en informatique ?

Question 4. : Quelle différence entre *http* et *https* ?

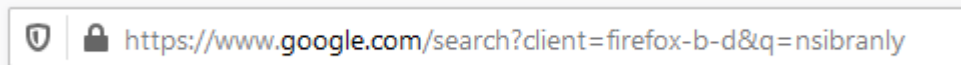
Question 5. : Quelle différence entre les méthodes GET et POST ?

6. EXEMPLE DE SITE QUI UTILISE LA METHODE GET POUR L'ENVOI DES DONNEES AU SERVEUR :

La méthode GET est utilisée par Google pour transmettre les éléments de votre recherche à leur serveur. Par exemple, en effectuant la recherche :

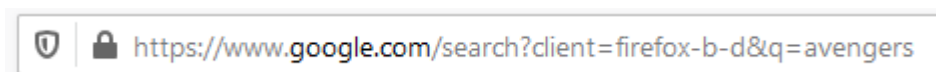


On retrouve dans la barre d'adresse :



⇒ Faire cette recherche et copier ensuite cette URL dans le presse-papier (*Ctrl C*), rafraichir la page *google.com* et coller cette adresse copiée dans la barre d'adresse vous retrouverez le résultat de la recherche précédente.

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



Oui, on en revient à notre formulaire. On avance !

Attention, comme on est en phase d'écriture de la page `tchat.html`, on travaille à nouveau en mode local, c'est-à-dire sur le fichier `tchat.html` qui est sur votre espace personnel du lycée Branly.

7. RAJOUT D'ATTRIBUTS A L'ÉLÉMENT `<INPUT>` DE TYPE 'TEXT':

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

Question 6. : S'agit-il d'éléments *inline* ou d'éléments *bloc* ?

Question 7. : La balise `<input>` est-elle orpheline ?

⇒ Rajouter les attributs *placeholder* et *autofocus* dans la balise `<input>` de type *text*

```
<form action="">
  <input type="text" name="objet" placeholder="objet du message" autofocus>
  <input type="submit" name="bouton" value="Envoyer">
</form>
```

Question 8. : 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.

`file:///..... /tchat.htm?objet=bonjour&bouton=Envoyer`

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

```
<input type="text" name="objet" placeholder="Objet du message" minlength="4" maxlength="20" required autofocus>
```

Mais pour notre tchat, 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.

8. RAJOUT D'UN LABEL A L'ÉLÉMENT `<INPUT>` DE TYPE 'TEXT':

L'élément `<input>` de type 'text' est généralement accompagné d'un élément `<label>`. Bien que ça ne soit pas obligatoire, on l'intègre ici à notre formulaire, car c'est ce qui se fait généralement. Pour mieux lier les éléments `<input>` et `<label>`, on les place tous les 2 dans un même bloc `<div>`.

⇒ Rajouter l'élément `<label>` comme indiqué ci-dessous :

Lier `<label>` avec `<input>` permet en cas de click sur le texte du `<label>`, d'obtenir le focus dans le champ de saisi. Pour lier `<label>` avec `<input>`, on doit ajouter un attribut « *for* » dans `<label>` et un attribut « *id* » dans `<input>` et ces 2 attributs doivent avoir LA MEME VALEUR (ci-dessous cette même valeur est "obj").

```
<form action="">
  <div>
    <label for="obj">Objet du message ; </label>
    <input type="text" name="objet" id="obj" placeholder="objet du message" autofocus>
  </div>
  <input type="submit" name="bouton" value="Envoyer">
</form>
```

⇒ Tester sur le navigateur cette histoire de focus : le focus est sur le champ de saisi au chargement de la page, car on a rajouté l'attribut « *autofocus* » dans la balise `<input>` correspondante. Cliquer en dehors du champ de saisi pour ne plus avoir le focus. Cliquer ensuite sur le texte du `<label>`.

Le focus se remet-il dans le champ de saisi ?

On continue ...



9. FINALISATION DU FORMULAIRE ET MISE EN LIGNE SUR NSIBRANLY.FR :

Lors de l'envoi du formulaire, les données saisies sont réceptionnées sur le serveur. Des codes écrits en d'autres langages que *l'html* ou le *css* y ont été installés afin de pouvoir traiter ces données. Ces codes font parties de ce qu'on appelle le « *backend* ». Dans le cadre de la 1^{ère} NSI, on ne s'intéresse qu'à la partie, dite « *frontend* », qui gère ce qui se passe sur le navigateur.

⇒ Finaliser le formulaire en rajoutant les attributs suivants dans la balise ouvrante `<form>` :

Indique l'adresse sur le serveur qui permet de retrouver le code qui va traiter les données.

Indique que l'on utilise ici non pas une méthode GET, mais une méthode POST. Cette méthode permet de transférer des données de taille plus importante.

```
<form action="/tchat" method="POST">
  <div>
    <label for="obj">Objet du message : </label>
    <input type="text" name="objet" id="obj" placeholder="objet du message" autofocus>
  </div>
  <input type="submit" name="bouton" value="Envoyer">
</form>
```

⇒ Uploader le fichier *tchat.html* dans votre espace Web sur *nsibranly.fr* (code web).

⇒ Afficher cette page dans le navigateur.

⇒ Saisir dans le champ de saisi « un texte qui vous est propre » et envoyer ... il ne se passe pas rien pour l'instant, hormis le fait qu'ici, l'Url n'a pas été modifié, car les données sont envoyées avec la méthode POST qui les insère dans le corps de la requête.

Le code de calcul de la partie « backend » récupère les données et les écrits dans un fichier au format json. Ce fichier se trouve à l'adresse suivante : <https://www.nsibranly.fr/membres/tchat.json>

⇒ Suivre ce lien dans un nouvel onglet, et vérifier que votre texte saisi figure dans ce fichier dans lequel les saisies de tous les élèves de la classe sont regroupées. Ce fichier contient seulement les 10 derniers messages postés. Par exemple, pour l'utilisateur de pseudo *christianD*, qui a saisi «un tchat aime les souris », le premier article du fichier est celui-ci-contre :

```
0:
  id_chat:      "10"
  expéditeur:   "christianD"
  quand:        "2024-04-28 18:47:41"
  objet:        "un tchat aime les souris"
  texte:        null
```

Si tout est correct, c'est que votre formulaire fonctionne parfaitement.



10. AMELIORATION POUR OBTENIR VRAIMENT UN TCHAT :

Les données transmises par le formulaire, pour tous les élèves de la 1^{ère} NSI sont donc traitées par le serveur, qui maintient à jour en temps réel un fichier qui contient les 10 derniers articles postés. Ce travail fait partie de la partie « backend ».

Le navigateur de votre ordinateur peut lui, assez facilement, récupérer ce fichier et en extraire les données pour les afficher sous le formulaire. On peut également lui demander de lire toutes les secondes ce fichier afin d'actualiser en temps réel les articles écrits sur la page. Pour réaliser tout cela, on utilise un langage de programmation compatible avec les navigateurs : le **javascript**.

Un des objectifs de cette fin d'année scolaire est de s'initier à ce nouveau langage. La syntaxe est légèrement différente du langage *python*. Mais si on réinvestit tout ce que vous connaissez déjà sur *python* et *tkinter*, les choses peuvent aller assez vite. Dans ce Tp, on se concentre sur la réalisation d'un tchat. On se contente donc ici, d'utiliser le code javascript tout fait qui se trouve dans le fichier `tchat.js` qui a été téléchargé en débit de tp.

⇒ Modifier le fichier `tchat.html` afin de lier au code javascript qui est écrit dans le fichier `tchat.js` :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Tchat</title>
  <link rel="stylesheet" href="tchat.css">
  <link rel="icon" type="img/png" href="tchat.png">
  <script src="tchat.js" async</script>
</head>
<body>
  <form action="/tchat" method="POST">
    <div>
      <label for="obj">Objet du message : </label>
      <input type="text" name="objet" id="obj" placeholder="objet du message" autofocus>
    </div>
    <input type="submit" name="bouton" value="Envoyer">
  </form>
  <section>
  </section>
</body>
</html>
```

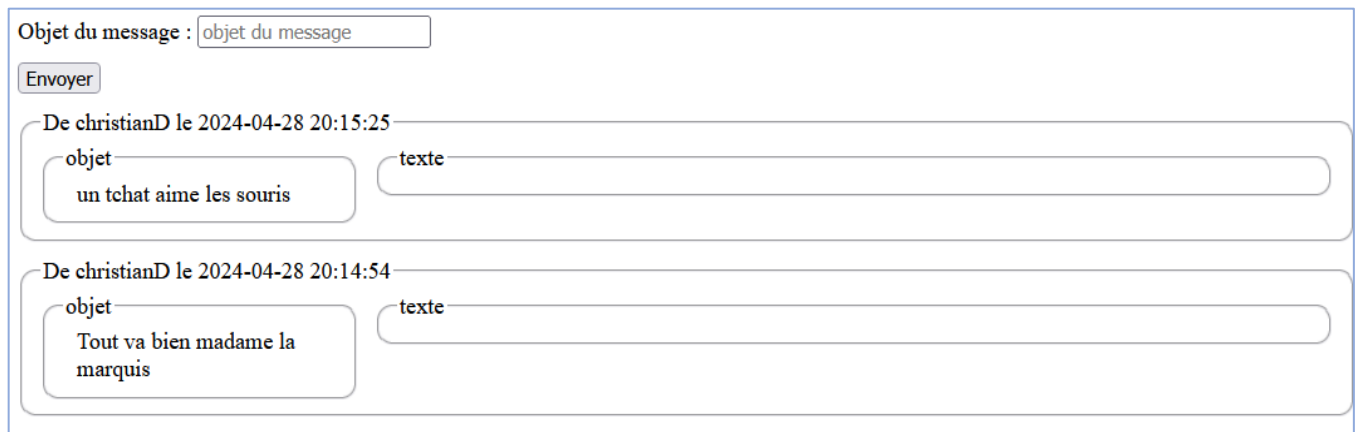
En écrivant cette ligne, lors du chargement de la page html, le fichier `tchat.js` sera lu et le code javascript qu'il contient sera exécuté

On crée un bloc `<section>` vide. C'est dans ce bloc que le code *javascript* va afficher les différents articles du chat

⇒ Uploader sur *nsibranly.fr* les fichiers *tchat.css*, *tchat.js* et *tchat.html*.

⇒ Afficher votre page *tchat.html*. Normalement les 10 derniers articles postés sont affichés, le plus récent en tête. Si vous postez un nouveau post en utilisant votre formulaire, quelques secondes plus tard votre texte saisi est affiché.

On obtient quelque chose qui ressemble à cela :



Objet du message :

Envoyer

De christianD le 2024-04-28 20:15:25

objet : texte :

De christianD le 2024-04-28 20:14:54

objet : texte :



À VOUS DE JOUER :

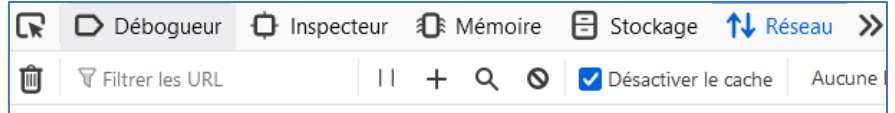
⇒ Compléter en local, le formulaire en rajoutant un bloc `<div>` contenant un élément `<textarea>` avec l'attribut `name` qui est égale à 'texte'. Cet élément `<textarea>` permet de saisir un texte de grande taille, contrairement aux éléments `<input>` :

```
<form action="/tchat" method="POST">
  <div>
    <label for="obj">Objet du message : </label>
    <input type="text" name="objet" id="obj" placeholder="objet du message" autofocus>
  </div>
  <div>
    <label for="txt">Texte du message : </label>
    <textarea type="text" name="texte" id="txt" placeholder="texte du message" cols="30" rows="4" ></textarea>
  </div>
  <input type="submit" name="bouton" value="Envoyer">
</form>
```

⇒ Tester en local le bon fonctionnement.

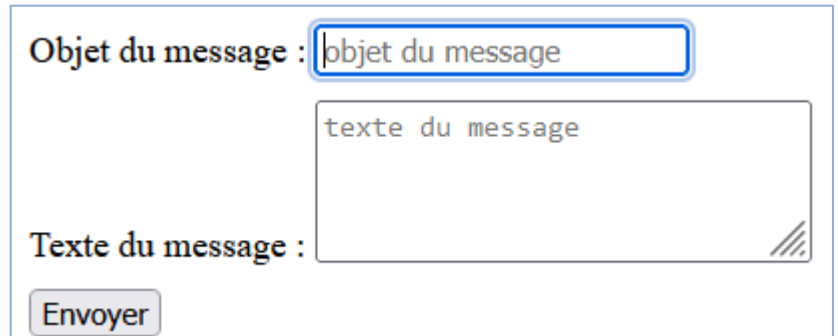
⇒ Uploader le fichier *tchat.html* dans votre espace Web sur *nsibranly.fr* (code web).

⇒ Afficher les outils du développeur. Dans l'onglet « Réseau », cocher la case « Désactiver le cache » pour demander au navigateur d'actualiser les fichiers utilisés :



⇒ Tester le bon fonctionnement en ligne. Envoyez un post avec objet et texte et vérifiez le bon affichage de l'ensemble.

Pour l'instant, le formulaire est affiché ainsi :

A screenshot of a web form. It has two main sections. The first section is labeled 'Objet du message :' and contains a text input field with the placeholder text 'objet du message'. The second section is labeled 'Texte du message :' and contains a larger text area with the placeholder text 'texte du message'. Below these sections is a button labeled 'Envoyer'.

⇒ Compléter le fichier *tchat.css* afin d'obtenir un visuel bien plus joli du formulaire et de la page entière. Jetez un coup d'œil sur google pour voir un peu les designs tendances pour les formulaires web. S'en inspirer.