

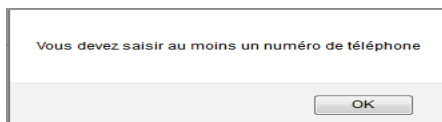
IUT LANNION 1ère Année BUT Réseaux informatiques et Télécoms	SAE23 - WEB2 : WEB Dynamique
TD7 (2H) : Javascript / TD8 (2H) : Ajax / TP6 (2H) : JS/Ajax	

1. VALIDATION D'UN FORMULAIRE (TD7 1H)

Les données transmises du client vers le serveur sont contrôlables avec les formulaires HTML5. Cependant, certains ne peuvent pas être réalisés donc elles peuvent être contrôlées par Javascript avant d'être envoyées. Rappel : Il est primordial qu'il y ait surtout un contrôle côté serveur et on peut aussi procéder à des tests côté client (Mais il faut savoir qu'en étant mal-intentionné on peut toujours passer outre l'exécution du Javascript).

1.1. CONTRÔLER SI LE FORMULAIRE CONTIENT UN NUMÉRO DE TÉLÉPHONE

Lorsqu'on clique sur le bouton de soumission du formulaire, le formulaire HTML5 vérifiera que les champs obligatoires sont remplis. De plus un script Javascript vérifiera que au moins un numéro de téléphone a été saisi. Si ce n'est pas le cas, une boîte de dialogue d'alerte doit être affichée et le formulaire ne doit pas être soumis (la même page reste à l'écran), sinon une page suite.php sera appelée.



Nom :

 Email :

 Tél Prof:

 Mobile:

Philippe DURAND [IUT Lannion](#)

1.2. INDIQUER L'ERREUR EN MODIFIANT LE STYLE CSS

Créer un nouveau formulaire qui va indiquer en rouge les champs qui sont mal remplis.

La boîte de dialogue signalant l'erreur doit être remplacée par un affichage en rouge sur la page elle-même, au-dessous du formulaire.

Nom :

 Email :

 Tél Prof:

 Mobile:

Nom :

 toto@toto
 Tél Prof:

 Mobile:

Il y a une erreur dans le second formulaire

2. FORMATAGE D'UN EMAIL EN JS (POUR LE PROJET ...)

Les formulaires HTML5 permettent de contrôler les champs adresses mails mais sans l'utilisation de pattern, le contrôle n'est pas très précis. Une alternative au pattern intégrable au formulaire est l'utilisation d'expressions régulières en Javascript.

Ajouter un contrôle : le champ mail doit être bien saisi : présence d'un '@' et d'un point derrière l'arobase, mais pas en dernière position et pas directement après l'arobase.

Pour cela nous allons utiliser le concept d'expressions régulières (nous pourrions le faire d'une manière moins élégante en analysant "manuellement" la chaîne de caractère).

Il suffira d'utiliser le code suivant:

```
function verifMail(champmail){ // reg est une expression régulière (classe Javascriptot  
existante ) et la méthode test est liée à la classe expression régulière  
var reg = /^[a-z0-9._-]+@[a-z0-9.-]{2,}[.][a-z]{2,3}$/;  
return (reg.test(champmail)); // retourne Vrai ou Faux  
}
```

Explication rapide: **reg.test(votrechaîne) retourne true ou false**

^ : depuis le début de la chaîne

[a-z0-9._-]+ : cherche au moins 1 caractère spécifié (de "a" à "z" ou "0" à "9" ou "." ou "-" ou "_")

@ : suivis de l'arobase

[a-z0-9]{2,} : suivis d'au-moins 2 caractères spécifiés (de "a" à "z" ou "0" à "9")

[.] : suivis d'un point (entre crochets car spécial dans regexp)

[a-z]{2,3} : suivis de 2 à 3 caractères spécifiés (de "a" à "z")

\$: et la chaîne doit être terminée

1. Utiliser la fonction Javascript `verifMail(champmail)` qui va vérifier **en plus** le bon formatage de l'email (cf seconde capture où email est écrit en rouge).

3. TEST CONNAISSANCES (TD7 1H À FINIR AU TP6 1H)

Vous devez coder une page qui interagit selon les réponses de l'utilisateur.
Récupérer le fichier testConnaissances_ETU.html

Test TP WEB2 - Exercice sur JavaScript

Testons votre culture générale

Cliquez ici pour commencer le questionnaire

Question 1 - En quelle année JavaScript a-t-il été créé ?

☐ 1985 ☐ 1990 ☐ 1995

Question 2 - En combien de jours a été créé le code JavaScript ?

Votre réponse :

Question 3 - JavaScript a été créé pour le compte de la *Netscape Communications Corporation*

☐ Vrai ☐ Faux

Source : WIKIPEDIA [Cliquez ici pour plus d'information](#)

3.1. CACHER LES QUESTIONS

Coder la fonction `cacheQuestions()`.
Appeler cette fonction au chargement de la page.

Test TP WEB2 - Exercice sur JavaScript

Testons votre culture générale

Cliquez ici pour commencer le questionnaire

3.2. AFFICHER LES QUESTIONS

Coder la fonction `afficherQuestion` en passant en paramètre le numéro de la question à afficher et tester cette fonction. Afficher la question 1 lors du clic sur « Cliquez ici pour commencer le questionnaire »

Test TP WEB2 - Exercice sur JavaScript

Testons votre culture générale

Cliquez ici pour commencer le questionnaire

Question 1 - En quelle année JavaScript a-t-il été créé ?

☐ 1985 ☐ 1990 ☐ 1995

3.3. QUESTION 1

Terminer le codage de la fonction `verifierQ1()`.
Si la réponse saisie est correcte (1995), nous affichons la question 2.

3.4. QUESTION 2

La réponse à la question 2 est 10 jours. L'utilisateur a 10 tentatives afin de trouver la réponse et à chaque réponse donnée, lui indiquer si le nombre à trouver est supérieur ou inférieur.

Coder la fonction **verifierQ2()**.

Question 2 - En combien de jours a été créé le code JavaScript ?

Votre réponse :

Tentative n°1, votre saisie : . C'est plus...

Ajouter le style pour le message à l'utilisateur (en rouge ici)

Affiner la fonction :

- lorsque la saisie est incorrecte redonner le focus au champ texte
- lorsque la saisie est correcte ou que le nombre de tentatives est dépassée, rendre inaccessible le champ texte et le mettre à jour avec la bonne valeur

(Remarque : Il existe les propriétés value, readonly, disabled et une méthode focus)

Lorsque la saisie est correcte, afficher la question 3.

Question 2 - En combien de jours a été créé le code JavaScript ?

Votre réponse :

Tentative n°2, votre saisie : 10. Vous avez trouvé ! On passe à la question 3.

Question 3 - JavaScript a été créé pour le compte de la *Netscape Communications Corporation*

☐ Vrai ☐ Faux

3.5. QUESTION 3

Coder la fonction **verifierQ3()** en vous inspirant de la fonction **verifierQ1()**.

Afficher le message final lorsque la réponse est correcte.

Question 3 - JavaScript a été créé pour le compte de la *Netscape Communications Corporation*

☒ Vrai ☐ Faux

Exact !

Source : WIKIPEDIA [Cliquez ici pour plus d'information](#)

3.6. QUESTION 3 - AJOUTER UNE FONCTION CACHERBTNOK()

Ajouter et coder une fonction nommée **cacherBtnOk()** qui devra être dans un fichier séparé, puis appeler la fonction lorsque la page est correctement saisie.

(1) Faire valider à l'enseignant en TP

4. GALERIE PHOTOS (TD8 1H15)

En utilisant, la technique de programmation (Web 2.0) AJAX, nous allons réaliser (analyser) une galerie d'images.

A l'aide de deux boutons, nous pourrons faire défiler et afficher à l'écran des images provenant d'un répertoire de stockage (img) de votre serveur Web.

Pour compléter, le défilement des images sera automatique toutes les 5 secondes.



4.1.1. PARTIE HTML (GALERIE.HTML)

```
<!DOCTYPE html>
<html lang="fr">
<head>
<meta charset="utf-8" />
<link href="design/galerie.css" rel="stylesheet" type="text/css" />
<script type="text/javascript" src="galerie.js"></script>
<title>Galerie d'images</title>
</head>
<body>
<h1>Galerie d'images</h1>
<hr />
<div>
<br/>


<br />
<p id="chemin_image"></p>
<img id="image" src="" alt="" />
<br/>
</div>
<hr/>
</body>
</html>
```

1. Quels autres fichiers sont nécessaires au fonctionnement de la galerie?
2. Quels attributs sont particulièrement importants en vue de faire de l'AJAX?
3. Comment déclencher le chargement d'une image lors de l'utilisation des boutons ?
4. Comment déclencher le chargement d'une image au lancement ?

4.1.2. PARTIE SCRIPT SERVEUR PHP POUR OBTENIR UNE PHOTO

Le script suivant (lireImage.php) permet de retourner le chemin absolu d'une image présente dans un dossier du serveur. Ex **http://monserveur/diaporama/img/toto.JPG**

```
<?php
/* On vérifie que le paramètre GET est bien présent : c'est un chiffre que l'on bornera au
nombre maximum d'images dans le dossier */
if(!empty($_GET) && isset($_GET['image'])){
    $image = $_GET['image']; // on crée une variable $image
    $rep = scandir('./img');
    // retourne un tableau contenant le nom des fichiers mais aussi . et .. dans les deux
    premières cases
    $tableau=array_slice($rep, 2);//créer un nouveau tableau en enlevant les deux
    premières cases contenant . et ..
    $nbimage = count ($tableau);
    $image = ($image%($nbimage));
    $this_url = pathinfo($_SERVER['REQUEST_URI']);
    // on récupère l'url de cette page
    echo "http://".$_SERVER['SERVER_NAME'].$this_url['dirname']."/img/".
    $tableau[$image];
}
else echo "Erreur GET";
?>
```

1. Donner un exemple d'appel pour utiliser ce script correctement.
2. Faire plusieurs essais pertinents pour valider le script et donner un exemple de réponse fournie.
3. Que valent \$_SERVER['SERVER_NAME'] et \$this_url['dirname']

4.1.3. PARTIE JAVASCRIPT : LANCEMENT REQUÊTE AJAX

```
// variables globales
var lien_image =0; // initialiser au premier chargement du fichier .js
/*****
/* le paramètre sens vaudra + ou - pour charger l'image précédente ou suivante */
function Deplacement(sens){
    // créer une requête AJAX
    var req_AJAX = null;
    if (req_AJAX) { // le navigateur permet AJAX
        // on incrémente ou décrémente l'image
        // on écrit l'url du script à demander sur le serveur
        // on définit la méthode TraiteReponse(req_AJAX) qui sera exécutée lors de chaque cycle de la
        requête AJAX. On appelle ce type de fonction un callback en informatique.
        // on spécifie l'action que l'on demande au serveur avec la méthode HTTP souhaitée
    }
    else alert("Le navigateur ne traite pas AJAX");
} //fin fonction Deplacement(sens)
```

1. Quels autres fichiers sont nécessaires?
2. Quelle URL est demandée par la requête AJAX?

3. Est ce un fichier (script) local à votre serveur? Pourquoi?
4. Quel est la méthode HTTP utilisée? Donner le(s) paramètre(s) transmis.

4.1.4. PARTIE JAVASCRIPT : RÉPONSE REQUÊTE AJAX

```
/* le paramètre est l'objet de la requête AJAX */
function TraiteReponse(requete) {
// test si la requête est en cours de chargement et affichage "En cours"
// test si la requête est terminée (état 4) et correcte (code HTTP 200) et affichage
image + lien
} // fin TraiteReponse
```

1. Quels attributs de votre page HTML seront modifiés par la requête AJAX? Et pendant quelle phase de la requête AJAX?
2. Localiser l'endroit de la réponse de la requête AJAX?
3. Rappeler un exemple de réponse attendue.
4. Coder la fonction
5. Proposer une modification pour que les images défilent toutes les 5 secondes.

5. FILTRAGE D'UN TABLEAU (TD8 45' À FINIR AU TP6 1H)

5.1. RAPPEL EN PHP (FOURNI)

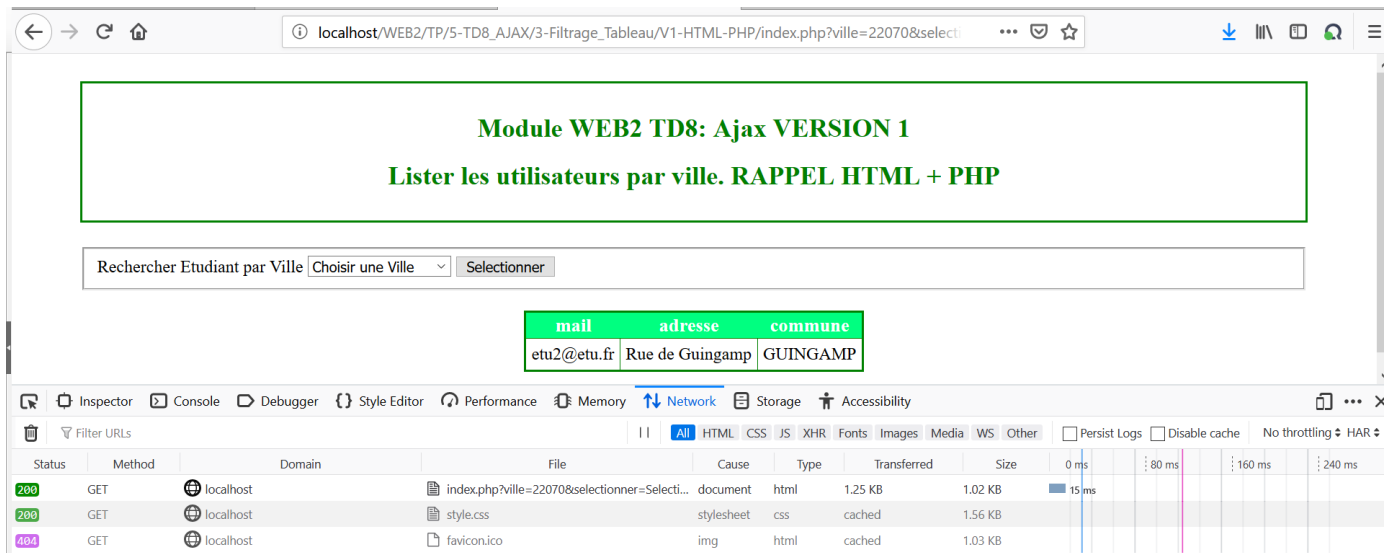
The screenshot shows a web browser at the URL `localhost/WEB2/TP/5-TD8_AJAX/3-Filtrage_Tableau/V1-HTML-PHP/index.php`. The page content is enclosed in a green border and contains the text:

Module WEB2 TD8: Ajax VERSION 1
Lister les utilisateurs par ville. RAPPEL HTML + PHP

Below this, there is a search form with the label "Rechercher Etudiant par Ville", a dropdown menu labeled "Choisir une Ville", and a "Selectionner" button.

The browser's developer tools are open, showing the Network tab. It lists three requests:

Status	Method	Domain	File	Cause	Type	Transferred	Size	Time
200	GET	localhost	index.php	document	html	1.11 KB	902 B	0 ms
200	GET	localhost	style.css	stylesheet	css	cached	1.56 KB	
404	GET	localhost	favicon.ico	img	html	cached	1.03 KB	



<?php

afficheFormulaireEtudiantParVille();

```

if (!empty($_GET) && isset($_GET["ville"])){
    //var_dump($_GET);
    $tab=listeEtudiantParVille($_GET["ville"]);
    if ($tab) afficheTableau($tab);
}
?>

```

Il y a deux A/R client/serveur dans ce cas.

5.2. AVEC JS (FOURNI)

The screenshot shows a web browser at the URL `http://localhost/WEB2/TP/5-TD8_AJAX/3-Filtrage_Tableau/V2-HTML-PHP -JS/index.php`. The page content is enclosed in a green border and contains the text: "Module WEB2 TD8: Ajax VERSION 2 HTML + PHP + JS" and "Lister les utilisateurs par ville. HTML + PHP + JS". Below this is a search form with the label "Rechercher Etudiant par Ville" and a dropdown menu labeled "Choisir une Ville".

Below the browser window, the Network tab of the developer tools is open, showing a list of requests:

Status	Method	Domain	File	Cause	Type	Transferred	Size	0 ms	80 ms	160 ms	240 ms
200	GET	localhost	index.php	document	html	1.14 KB	926 B	15 ms			
200	GET	localhost	index.js	script	js	cached	95 B				
200	GET	localhost	style.css	stylesheet	css	cached	1.56 KB				
404	GET	localhost	favicon.ico	img	html	cached	1.03 KB				

Dans cette version, le formulaire a été modifié :

```
<select id="id_ville" name="ville" size="1" onchange = "listerVilles(this)" >
```

pour détecter un changement dans la balise `<select>` avec le gestionnaire d'événement JS `onchange` qui appelle la fonction JS `listerVilles()`.

Cette fonction peut par exemple demander le rechargement de la page

```

function listerVilles(ville){
    window.location.assign("index.php?ville="+ville.value);
}

```

Il y a deux A/R client/serveur dans ce cas. On aurait pu aussi charger lors de la première demande du client toutes les données dans le navigateur et faire un traitement local de ces données ce qui n'aurait impliqué qu'un aller retour client/serveur mais beaucoup de données transportées inutilement.

5.3. A FAIRE AVEC AJAX (TP6 1H)

Module WEB2 TP5: Ajax
Lister les utilisateurs par ville en AJAX

Rechercher Etudiant par Ville

Module WEB2 TP5: Ajax
Lister les utilisateurs par ville en AJAX

Rechercher Etudiant par Ville

mail	adresse	commune
etu@etu.fr	Rue de Lannion	LANNION
etu4@etu.fr	Route de Trébeurden	LANNION
admin	rue Edouard Branly	LANNION

Rappel de la méthode :

1. Identifier l'événement déclencheur dans le formulaire à inclure dans la page **index.php** et la zone dans la page destinatrice de la réponse.
2. Lier cet événement à une fonction javascript **listeFiltreUtilisateurs(ville)**
3. Ecrire cette fonction javascript **listeFiltreUtilisateurs(ville)** qui va lancer une requête AJAX à destination d'un script PHP **listeUtilisateurs.php** s'exécutant sur le serveur en précisant les paramètres à passer et la méthode de passage des paramètres.
4. Ecrire le script PHP **listeUtilisateurs.php** de traitement des paramètres reçus par la requête AJAX. Ce script doit retourner une réponse à la requête AJAX
5. Ecrire la fonction javascript **TraiteListeFiltreUtilisateurs(requete)** qui va traiter la réponse fournie par le script PHP précédent et l'insérer dans la page **index.php**.

(2) Faire valider à l'enseignant en TP