TP 5 : Formulaire HTML et vérification en JS (avec solutions)

Objectifs du TP:

- Découvrir le language Javascript (JS) côté client.
- Utiliser JS pour valider un formulaire.
- Comprendre le fonctionnement du traitement des formulaires.
- Comprendre le système d'envoi de données au PHP (serveur) et retour des données au HTML (client) avec la conservation de l'état de l'interface dans le query string.

1. Introduction au langage Javascript

Javascript (JS), comme PHP, est un langage objet, dynamique, interprété, avec des variables non typées. Du premier version date de 1995, il a langage TypeScript est un langage compatible avec Javascript qui apporte du typage mais qui est compilé en Javascript (ou JS avec transpiler). Dans ce cours, nous ne nous intéresserons qu'à Javascript.

1.1 Syntaxe de base

La syntaxe de JS est inspirée de Java, qu'si même est inspiré du language C++, une extension du language C.

Chaque navigateur (Firefox, Chrome, Safari) possède son interpréteur JS. Une console permet d'afficher des messages, comme les erreurs de syntaxe ou chargement d'un fichier. Il est donc important de bien ouvrir la console avant de tester.

Les variables sont définies comme en Java avec des noms formés de lettres, chiffres et le caractère soulignè ( ). Attention, le caractère $ est également un caractère possible (mais non obligatoire) pour les variables. En pratique, il est déconseillé de l'utiliser.

Par défaut les variables peut être vide (non initialisé) et non typé ( ) est donc aussi mais il est **déconseillé et déconseillé**. NB: la syntaxe de JS permet d'initialiser une variable sans la déclarer (comme en PHP).

Les mots-clés réservés pour les valeurs sont : les chaînes de caractères, les nombres (flottants), les booléens, null, undefined, les symboles et grands entiers. Il n'y a qu'une manière de structurer des données, comme des objets. Ce langage n'implémente ni classes ni interfaces, ou des objets. Il y a complètement chaîne de comme C++ ou Java.

Les instructions (if, switch, while et for) sont identiques dans le language. Seul le langage Java ne permet pas de déclarer des fonctions (mais la notion similaire de Lambda existe en Java et dans les 2 autres langages).

L'envoie de message sur les objets se fait avec la syntaxe de Java (un point). Comme dans les autres langages, les variables stockent une référence sur les objets (tab). On n'est ni seul ni des langages à manipuler explicitement des adresses et il pourrait avoir doubler un objet dans une variable.

Syntaxe	C++	Java	PHP	Javascript
Déclaration et initialisation de variable	int x = 3;	int x = 3;	$x = 3;	var x = 3;
Envoie de message sur référence	o.method(param);	o.method(param);	$o->method($param);	o.method(param); o["method"](param);
Envoie de message sur référence	o.method(param);	o.method(param);	-	-
Concaténation de chaînes	string s = "bar" + "foo"; s += " pro";	String s = "bar" + "foo"; s += " pro";	$s = "bar"; $s = $s . "foo"; $s .= " pro";	var s = "bar"; var s = "foo"; s += " pro";
Déclaration de fonction	int foo(int i, int j) { ... }	void foo(int i, int j) { ... }	function foo($i, $j) { ... }	function foo(i, j) { ... }

La variable constante est toujours utilisée comme une référence sur l'objet qui représente la constante. Cela permet des verifier sur la constante avec le méthode Object.

Attention: en JS l'opérateur + est utilisé à la fois pour l'addition (pour les nombres et l'opérateur de concaténation pour les chaînes de caractères. JS comme PHP, permet de faire automatiquement le changement de type entre nombres et chaînes mais avec l'addition, c'est la transformation vers les chaînes qui est réalisée. Donc

```
"foo"+3
```

est évalué à

```
"foo3"
```

alors que

```
3+"foo"
```

est évalué à

```
"3foo"
```

Si vous utilisez la fonction de changement de type Number() alors

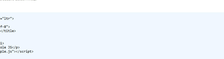
```
Number("3")
```

est évalué à

```
3
```

A faire

- Créer une page HTML simple, puis l'ouvrir avec le navigateur (Firefox).
- Ouvrir la console : «View and Search » puis « Tools Web Dev.». Dans la fenêtre : "Web Developer Tools", ou bien simplement clic droit et "Inspect", ou encore la touche F12, puis activer la console



- Dans la console, déclarer une variable et lui donner une valeur (ex: un entier). Constatez qu'on ne peut pas déclarer deux fois la même variable.

> **Solution :**
>
> ```
> > var a = 18;
> 1
> > var a = 19;
> 1
> > SyntaxeError: redeclaration of let a
> ```
>
> - Javascript détecte que l'erreur
> - Javascript ne change pas la valeur
>
> ```
> > a
> 19
> ```

- Chaque instruction JS est évaluée, et rend donc une valeur, mais qui peut être **undefined**. Dans la console, il suffit alors de donner le nom d'une variable pour voir sa valeur. La méthode typeof rend le type de la variable pour afficher une trace dans la console. Testez dans la console la création d'une variable et son affichage.

> **Solution :**
>
> ```
> > var a = 18;
> 1
> > typeof(a)
> "number"
> > a
> 18
> > console.log(a)
> 18
> > undefined
> ```

- Vous pouvez exécuter des fonctions depuis la console. Lancez la fonction alert avec un message, et constatez que cela ouvre une fenêtre de dialogue.

> **Solution :**
>
> ```
> > alert("test")
> > undefined
> ```

- Utilisez la fonction reverse pour demander son nom à l'utilisateur et le stocker dans une nouvelle variable.

> **Solution :**
>
> ```
> > var name = prompt("entrez votre nom")
> > undefined
> > name
> "toto"
> > typeof(name)
> "string"
> ```

1.2 Entrée du code JS

Le code JS doit-il sur le navigateur pour les tests, soit dans fichier (ainsi suffixe .js). Il est possible d'écrire directement du JS dans le HTML, mais pour la même raison qu'on ne mélange pas le CSS dans le HTML, on ne doit pas mélanger **du JS avec du HTML**. Le fichier JS s'inclut dans le HTML avec la balise script (le préférence à la fin du fichier HTML. Un fichier JS ne contient qu'un seul fichier JavaScript, et aucun autre fichier.

```
<!DOCTYPE html>
<html lang="fr" dir="ltr">
  <head>
    <meta charset="UTF-8">
    <title>Example JS</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>Example JS</h1>
    <p>Ouvrir la console JS</p>
    <script src="example.js"></script>
  </body>
</html>
```

Contenu du fichier exemple.js :

```
console.log("Hello");
```

A faire :

- Créer un programme JS qui demande à l'utilisateur (prompt) un nombre, puis affiche la table de multiplication (1 à 10) de ce nombre. On doit voir s'afficher dans la console :

> **Solution :**
>
> Le fichier HTML
>
> ```
> <!DOCTYPE html>
> <html lang="fr" dir="ltr">
> <head>
> <meta charset="UTF-8">
> <title>Table de multiplications</title>
> </head>
> <body>
> <h1>Table de multiplications</h1>
> Ouvrir la console JS
> <script src="js/table.js"></script>
> </body>
> </html>
> ```
>
> Le fichier JS :
>
> ```
> let table = prompt("Entrez la table");
>
> console.log("Voici la table de "+table);
>
> for(let i=1;i<=10;i++) {
> console.log(i+" x "+table+" = "+(i*table));
> }
> ```

- Complétez votre code pour re-demander à l'utilisateur un autre nombre et son entrée n'est pas un nombre. Utilisez la fonction isNaN() qui détecte si son paramètre n'est pas un nombre.

> **Solution :**
>
> ```
> let table = Number(prompt("Entrez un nombre"));
>
> while (isNaN(table)){
> console.log("Ce n'est pas un nombre, entrez un nombre");
> }
>
> console.log("Voici la table de "+table);
>
> for(let i=1;i<=10;i++) {
> console.log(i+" x "+table+" = "+(i*table));
> }
> ```

Le fonctionnement doit être le suivant :

- Si les deux mots de passe sont non vides et identiques, un message le signale et on reste sur la page.
- Si le mot de passe ne respecte pas les règles, un message le signale et on reste sur la page.
- Si le mot de passe est correct et identique au mot de passe de confirmation, le mot de passe dans le fichier password.txt et affiche un message de réussite. On se contentera d'un enregistrement "en dur", mais en réalisant cette tâche il est important d'encoder le mot de passe avant de l'enregistrer, par exemple avec la fonction sequeel(mot).

2. Vérification de formulaire avec du JS

2.1 Spécification

On doute dabeur une page pour réinitialiser un mot de passe. Nous allons utiliser un formulaire avec les éléments suivants :

- Un litre en gras : "Retest a password" (Retestonsnous un mot de passe).
- Un champ de saisie de mot de passe avec un libellé en gras : "Type your password" ("Saisir votre mot de passe). "Retype your password" (Retapez votre mot de passe). Ces champs de saisie ne doivent pas faire apparaître un jeté des caractères du mot de passe.
- Un bouton de validation : "Set new password" (Valider le nouveau mot de passe). Ce bouton va déclencher la vérification de la validité des mots de passe en local en Javascript (sur le navigateur client), puis va passer le mot de passe à un script PHP (sur le serveur Apache).

Les instructions sous le formulaire indiquent les règles à suivre pour le mot de passe :

- Le mot de passe doit faire au moins 8 caractères.
- Il doit inclure au moins un chiffre.
- Il doit inclure au moins une lettre majuscule et une lettre minuscule.

2.2 Aide technique

Les balises HTML peuvent déclencher un code Javascript sur un évènement. Du code javascript peut être placé dans un attribut HTML onXXX, où XXX désigne le nom de l'évènement. Ce code doit alors chaîne de caractères qui sera exécutée par le navigateur client. Pour plusieurs instructions sont séparées par des delimiteurs (point-virgules). Il est donc aussi possible de déclencher la vérification du code JS. Il faut donc ajouter l'attribut onsubmit à la balise HTML form.

Le code JS du fichier document.js est exécuté par un binding, avec l'instruction require. Ce binding indique à la formulaire web utilisé (mais, ou le nombre d'un des deux champs n'est pas vide) et le code JS qui doit être exécuté. Il est donc nécessaire de nommer le formulaire avec l'attribut name. Le code JS va ensuite utiliser le DOM pour accéder à ce formulaire et à ses champs. Il va ensuite modifier le formulaire pour y ajouter un bouton de validation. Ce bouton va déclencher le document HTML actuellement affiché dans le navigateur. L'objet référencé par le DOM va ensuite utiliser l'attribut onsubmit pour déclencher le code JS. Il faut donc ajouter du code dans form, en avant accéder à chaque champ (input) de ce formulaire en cas contraire. Il faudra ensuite ajouter du code d'erreur (input) et activer à son contenu avec l'attribut id.

Résumé : il porte d'un objet form créé par un parmet de déclencher l'attribut qui fait référence à un formulaire, alors que pour un formulaire créé l'attribut name qui permet de référencer un champ input. Finalement il est nécessaire d'accéder à la valeur du champ. Donc, si le HTML est présente ainsi :

```
<form action="response.php" id="myform">
    <input type="text" name="myfield">
</form>
```

Le code du contrôle du champ texte depuis JS crée ainsi :

```
let textInputValue = document.forms.myform.myfield.value
```

Pour résoudre les textes de présentation de caractères, vous avez vital 3 fonctions qui réalisent ces tests à partir d'une expression régulière (mot de passe). Pour les tests JS Automatiques et Langages. Pour différer et manipuler les expressions et JS avec une libraire moment, vous devrez utiliser le binding Lodash. Le langage d'un objet de type String est donc utilisé. Voici :

A faire

- Téléchargez et décompressez le fichier dabeurtest.zip.
- Complétez les fichiers pour réaliser le fonctionnement décrit.

> **Le formulaire HTML :**
>
> ```
> <!DOCTYPE html>
> <html lang="fr">
>
> <head>
> <meta charset="UTF-8">
> <title>Create new password</title>
> <link rel="stylesheet" href="style.css">
> <script type="text/javascript">
> </head>
>
> <body>
>
> <section>
> <h1>Select a password</h1>
> <form name="selfservicepwd" action="set_passwd.php" onsubmit="return validateform()" method="POST">
> 
> Type your password:
> <input type="password" name="password" placeholder="Password" required>
> <br><br>
> Retype your password:
> <input type="password" name="passwordcf" placeholder="Password" required>
> <br><br>
> <button type="submit">Set new password</button>
> </form>
> <br>
> Your password must follow the following rules:
> <br>
> <br>
> - Length: 8-12 characters.
> <br>
> - Contains at least 1 digit.
> <br>
> - Includes a digit.
> <br>
> - Includes upper and lower case.
> <br>
> <br>
> <br>
> </section>
>
> </body>
>
> </html>
> ```
>
> **Le code de validation en JS :**
>
> ```
> // Returns true is there is a digit in string s
> function hasDigit(s) {
> var digit = /[0-9]/;
> return !!digit.test(s);
> }
>
> // Returns true is there is a uppercase letter in string s
> function hasUpper(s) {
> var upper = /[A-Z]/;
> return !!upper.test(s);
> }
>
> // Returns true is there is a lowercase letter in string s
> function hasLowercase(s) {
> var lower = /[a-z]/;
> return !!lower.test(s);
> }
>
> // Returns true if form has valid data
> function validateform() {
> // Declaration dans une variable globale des deux mots de passes du formulaire
> let passwordv1 = document.forms.selfservicepwd.password.value;
> let passwordv2 = document.forms.selfservicepwd.passwordcf.value;
> // Test Applicatif mots de passes
> if (passwordv1 != passwordv2) {
> alert("Error: passwords differs.");
> return false;
> }
> // Test de la longueur
> var lg = passwordv1.length;
> if (lg < 8 || lg > 12) {
> alert("Error: " + lg + " char long is too small.");
> return false;
> }
> // Test chiffre
> if (!hasDigit(passwordv1)) {
> alert("Error: no digit in string.");
> return false;
> }
>
> // Test case
> if (!hasUpper(passwordv1) || !hasLowercase(passwordv1)) {
> alert("Error: upper case is missing.");
> return false;
> }
>
> return true;
> }
> ```
>
> **Le code de la réponse en PHP :**
>
> ```
> <?php
> if (isset($_POST['password'])){
> // Le mot de passe existe, on enregistre un service le script
> exit();
> }
> $passwd = $_POST['password'];
> // Encodage du mot de passe en clair (à écrire)
> file_put_contents("passwd.txt",$passwd);
>
> <!DOCTYPE html>
> <html lang="fr">
>
> <head>
> <meta charset="UTF-8">
> <title>Create new password</title>
> <link rel="stylesheet" href="style.css">
> </head>
>
> <body>
> <section>
> <h1>Password updated.</h1>
> </section>
>
> </body>
>
> </html>
> ```

3. Conversion de températures

Vous devez traiter un convertisseur de température, de Celsius à Fahrenheit. La formule de conversion est :

$$F = \frac{9}{5} \times C + 32$$

Il faut donc convertir un Celsius, et il le même température mais en Fahrenheit. Dans une première partie, on réalise un site qui fait la conversion dans un sens, et dans la deuxième partie, on le complète pour pouvoir inverser le sens de conversion.

3.1 Conversion de Celsius en Fahrenheit

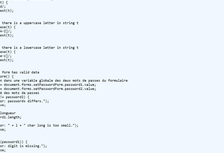
Le fichier :

```
conversion.php
```

contient l'élocution de la page HTML avec du PHP pour réaliser la conversion. Cette page doit fonctionner de la manière suivante :

- Ouvrant un adresse la page (page query string), ou appelant sur la page de valeur dans les degrés Celsius, alors la valeur "C" est affichée dans le résultat (output) en Fahrenheit.
- On entre une valeur, puis on active le bouton. Le résultat apparaît alors.

Avant la conversion



Après la conversion

Au niveau organisation, le fichier conversion.php est découpé en deux parties :

- **Contrôleur** : le fichier débute par l'URL et une partie PHP qui va récupérer les données et faire un calcul. C'est ce qu'on appelle la partie **contrôle** de la page. Cette partie ne contient que du code PHP.
- **Vue** : dans la suite du fichier, à partir de <!DOCTYPE html> la partie vue qui contient du HTML avec le minimum de PHP nécessaires pour modifier le HTML.

A faire :

1. Téléchargez le fichier zip de cet exercice : conversion.zip.
2. Examinez le contenu du fichier conversion.php. Voici l'élocution de cette page de conversion. Il est constitué d'une partie PHP pour l'instant vide, et d'une partie HTML qui est la "vue" que cette page doit afficher. Cette vue est constituée d'un titre et d'un formulaire. Le formulaire est composé d'une entrée (input) et d'une saisie (button). Chaque élément possède un id(id). Le formulaire possède un attribut action qui indique que le formulaire va appeler le même fichier (conversion.php) lorsque l'on va activer le bouton. Le formulaire possède aussi un attribut method qui indique que le formulaire va utiliser la méthode POST. Le formulaire définitive alors le nom et la valeur du champ input. Le formulaire définitive alors le nom et la valeur du champ input. Le formulaire définitive alors le nom et la valeur du champ input