Contrôle des formulaires avec JavaScript

Contrôle d'un formulaire de contact avec l'évènement *submit*

Voici un exemple où le JavaScript bloque l'action du formulaire tant que les champs ne sont pas valides. Quelques points à retenir :

- Le tableau *document.forms[0]* identifie le premier formulaire de la page
- L'action du formulaire est bloquée par la fonction *preventDefault()*
- La captation de la soumission du formulaire est réalisée par la méthode *addEventListener(« submit », ...)*

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
</head>
<body>
    <h1>Contact</h1>
    <form method="post" action="email.php">
        <label>E-mail :</label>
        <input type="text" id="email" name="email">
        <label> Votre message :</label>
        <textarea id="message" name="message"></textarea>
        <input type="submit">
    </form>
    <script>
        document.forms[0].addEventListener("submit", function(evenement) {
            if (document.getElementById("email").value == "") {
                evenement.preventDefault();
                alert("Tapez un email valable pour avoir une réponse.");
                document.getElementById("email").focus();
            }
            else if (document.getElementById("message").value == "") {
                evenement.preventDefault();
                alert("Pensez à taper un message !");
                document.getElementById("message").focus();
            }
        });
    </script>
</body>
```

Notez qu'un <input type='email'> permet aux navigateurs récents de vérifier la structure valide d'un email sans ajout de code JavaScript.

Le code d'**email.php**

```
<!doctype html>
<html>
    <head>
        <meta charset="utf-8">
    </head>
    <body>
                <h1>Contact</h1>
        <p>
        <?php
        $retour=mail(contact@monsite.fr,'Envoi depuis la page contact de monsite.fr',
$_POST['message'], 'From:'.$_POST['email']);
        if($retour)
            echo 'Votre message a bien été envoyé !';
        ?>
```

```
        </p>
    </body>
</html>
```

Si vous souhaitez comprendre la fonction *mail()* du langage PHP, n'hésitez pas à consulter **ce support ce cours**.

Variante avec l'évènement *onsubmit* dans la balise *form*

Une autre solution, plus traditionnelle, consiste à intégrer l'attribut *onsubmit* dans la balise *form*. Si la fonction *verif()* renvoie la valeur *false*, alors le formulaire n'exécutera pas l'action du formulaire et le script *email.php* ne sera pas lancé.

```
<!DOCTYPE html>
<html>
        <head>
        <meta charset="utf-8">
        <script>
        function verif() {
                if (document.getElementById("email").value=="") {
                        alert("Tapez un email valable pour avoir une réponse.");
                        document.getElementById("email").focus();
                        return false;
                }
                if (document.getElementById("message").value=="") {
                        alert("Pensez à taper un message !");
                        document.getElementById("message").focus();
                        return false;
                }
        }
        </script>
        </head>
        <body>
                <h1>Contact</h1>
                <form method="post" onsubmit="return verif()" action="email.php">
                        <label>E-mail : </label>
                        <input type="text" id="email" name="email">
                        <label> Votre message : </label>
                        <textarea id="message" name="message"></textarea>
                        <input type="submit">
                </form>
        </body>
</html>
```

Contrôle d'un champ de texte avec *change*

L'évènement *change* permet de détecter qu'un champ a fini d'être saisi. Suite à une saisie, le navigateur détectera :

- l'appui sur la touche entrée,
- un clic à l'extérieur du champ,
- un appui sur la touche tabulation (perte de focus)

Dans l'exemple ci-dessous, on vérifie la validité d'un âge :

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
</head>
<body>
    <label>Tape ton âge</label>
    <input type="number" id="age">
    <script>
```

```
        document.getElementById("age").addEventListener("change", verif);

        function verif() {
            age = document.getElementById("age").value;
            if(age<=10 || age>=120)
                alert("Tapez un âge compris entre 10 et 120 ans");
            else
                alert("Vous avez : " + age + " ans");
        }
    </script>
</body>
</html>
```

Désactivation du JavaScript et sécurité

Il faut garder à l'esprit que le JavaScript est facilement désactivable dans les paramètres des navigateurs et par conséquent, ce langage ne permet pas une réelle sécurité. C'est la même chose pour les **CAPTCHAS** codés en JavaScript. Il faudra dans ce cas privilégier **une sécurisation de formulaire avec le langage PHP**. Cependant, ces petits contrôles JavaScript permettent dans bien des cas d'éviter les problèmes de saisie qui ne sont pas forcément malveillants, mais plutôt causés par des étourderies de saisies.