



**ECOLE MAROCAINE DES
SCIENCES DE L'INGENIEUR**
Membre de
HONORIS UNITED UNIVERSITIES



Développement Web: JavaScript

Professeur:

Nouhaila MOUSSAMMI

n.moussammi@emsi.ma

Gestion des événements



Exercice 3 :

Objectif: Ajoutez un écouteur d'événements pour les événements du clavier à l'élément html d'une page Web. Lorsqu'une combinaison spécifique de touches est enfoncée (par exemple, a, b, c, d), un message s'affiche pour indiquer que la combinaison a été saisie correctement.

Solution (Exercice 3) :



```
<p id="affichage"></p>
<script>
  addEventListener('keydown', keyPressed);
  addEventListener('keyup', keyReleased);
  var keys = [];
  function keyPressed(event)
  {
    var msg = document.getElementById('affichage');
    var texte = document.createTextNode('Vous avez cliqué sur les touches a et b');
    keys[event.keyCode] = true;
    if (keys[65] && keys[66] && msg.textContent === '') //touches 'a' et 'b'
    {
      msg.append(texte);
    }
  }
  function keyReleased(event)
  {
    var msg = document.getElementById('affichage');
    keys[event.keyCode] = false;
    msg.innerHTML= '';
  }
</script>
```

event.keyCode : Cette propriété de l'objet event contient le code numérique de la touche pressée. Par exemple, le code 65 correspond à la touche "a" et le code 66 correspond à la touche "b".

Jeu de Pierre-Papier-Ciseaux

Sélectionnez une option :

Pierre

Papier

Ciseaux

Vous

-

Ordinateur

-

Score :

0 - 0



Validation des formulaire



Gérer les éléments d'un formulaire

JS JavaScript

1 - Soumission d'un formulaire

La méthode **submit()** du DOM est utilisée pour soumettre les données du formulaire à l'adresse spécifiée dans l'attribut action.

Cette méthode se comporte comme le bouton de soumission du formulaire et ne prend pas de paramètres.

Syntaxe :

```
form.submit()
```

Exemple :

```
<form id="FORM1" method="post" action="/code.php">
<label>Nom <input type="text" name="nom"></label><br>
<label>Age <input type="text" name="Age"><label> <br>
<input type="submit" onclick="SoumettreForm()" value="SUBMIT">
<input type="button" onclick="ResetForm()" value="RESET">
</form>
<p id="message"></p>
<script>
    function SubmitForm() {
        document.getElementById("FORM1").submit();
    }
    function ResetForm() {
        document.getElementById("FORM1").reset();
        document.getElementById("message").innerHTML="Formulaire réinitialisé";
    }
</script>
```

Gérer les éléments d'un formulaire

2 -Interruption d'un formulaire

La fonction **preventDefault()** empêche l'exécution de l'action par défaut de l'événement.

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title></title>
5    </head>
6    <body>
7      <a href="http://www.emsi.ma">Site de l'EMSI</a>
8      <script>
9        var lien = document.querySelector("a");
10       lien.addEventListener("click", function (event) {
11         event.preventDefault();
12         console.log("Redirection interdite");
13       });
14     </script>
15   </body>
16 </html>
```

Gérer les éléments d'un formulaire

JS JavaScript

3- Validation d'un formulaire

La validation des données

La validation des données consiste à s'assurer que l'entrée de l'utilisateur est conforme aux données attendues. Parmi les types de vérifications à faire on cite :

- l'utilisateur a-t-il rempli tous les champs obligatoires ?
- l'utilisateur a-t-il saisi une date valide ?
- l'utilisateur a-t-il saisi du texte dans un champ numérique ?
-

On distingue deux types de validation :

- **La validation côté serveur** : effectuée par un serveur Web, une fois que les données sont envoyées au serveur.
- **La validation côté client** : effectuée par un navigateur Web, avant que les données ne soient envoyée à un serveur Web.

HTML5 a introduit un nouveau concept de validation HTML appelé « validation des contraintes » qui est basée sur :



Gérer les éléments d'un formulaire

3- Validation d'un formulaire

Validation des contraintes HTML

Validation des contraintes en utilisant les sélecteurs CSS

Sélecteur	Description
:disabled	Sélectionner les éléments désactivés
:invalid	Sélectionner les éléments dont la valeur est invalide
:optional	Sélectionner les éléments d'entrée sans attribut "requis" spécifié
:required	Sélectionner les éléments d'entrée avec l'attribut "requis" spécifié
:valid	Sélectionner les éléments d'entrée avec des valeurs valides

Gérer les éléments d'un formulaire

3- Validation d'un formulaire

Validation des contraintes par les attributs

Attribut	Description
disabled	L'input doit être désactivé
max	Spécifier la valeur maximale d'un élément input
min	Spécifier la valeur minimale d'un élément input
pattern	Spécifier un modèle de chaîne (Regex)
required	Saisie obligatoire
type	Spécifier le type d'un élément input

Gérer les éléments d'un formulaire

JS JavaScript

3- Validation d'un formulaire

Exemple : Si un champ de formulaire (nom) est vide, la fonction affiche un message et renvoie false pour empêcher la soumission du formulaire

```
function validerForm() {  
    let x = document.forms["myForm"]["nom"].value;  
    if (x == "") {  
        alert("Le champ "nom" doit être saisi");  
        return false;  
    }  
}
```

```
<form name="myForm" action="/code.php" onsubmit="return validerForm()" method="post">  
    Nom: <input type="text" name="nom">  
    <input type="submit" value="Submit">  
</form>
```

Gérer les éléments d'un formulaire

JS JavaScript

3- Validation d'un formulaire

Validation automatique des formulaires HTML

La validation du formulaire HTML peut être effectuée automatiquement par le navigateur :

Si un champ de formulaire (nom) est vide, l'attribut `required` empêche la soumission du formulaire

```
<form action="/code.php" method="post">  
  <input type="text" name="nom" required>  
  <input type="submit" value="Submit">  
</form>
```

Interactive User Form

Change Background Color

Enter your name

Enter your email

Enter your password

Enter your phone number

Submit

user1@emsi.ma

Remove

user2@emsi.ma

Remove

user3@emsi.ma

Remove

Hover over me!

Reset

JS JavaScript