

# Trois exemples Javascript

<b><u>1</u> JAVASCRIPT.....</b>	<b>1</b>
<b><u>1.1</u> Récupérer les informations d'un formulaire.....</b>	<b>1</b>
<u>1.1.1</u> Le formulaire.....	1
<u>1.1.2</u> Le code.....	1
<b><u>1.2</u> Les expressions régulières en Javascript.....</b>	<b>3</b>
<u>1.2.1</u> La page de test.....	3
<u>1.2.2</u> Le code de la page.....	4
<b><u>1.3</u> Gestion des listes en JavaScript.....</b>	<b>6</b>
<u>1.3.1</u> Le formulaire.....	6
<u>1.3.2</u> Le code.....	6

# 1 JAVASCRIPT

Nous montrons dans cette partie trois exemples d'utilisation de Javascript dans les pages WEB. Nous nous centrons sur la gestion des formulaires mais Javascript peut faire bien davantage.

## 1.1 Récupérer les informations d'un formulaire

L'exemple ci-dessous montre comment récupérer au sein du navigateur les données entrées par l'utilisateur au sein d'un formulaire. Cela permet en général de faire des pré-traitements avant de les envoyer au serveur.

### 1.1.1 Le formulaire

On a un formulaire rassemblant les composants les plus courants et d'un bouton **Afficher** permettant d'afficher les saisies faites par l'utilisateur.

**Un formulaire traité par Javascript**

Un champ textuel simple      Un champ textuel sur plusieurs lignes

Des boutons radio :  FM  GO  PO    Des choix multiples :  un  deux  trois

Un menu déroulant : 50 F    Une liste : Renault, Citroën, Peugeot

Un mot de passe : [masqué]    Un champ de contexte caché :

---

**Informations du formulaire**

Effacer    champ caché=secret  
champ textuel simple=  
champ textuel multiple=Ce texte est modifiable    Afficher

### 1.1.2 Le code

```
<html>
<head>
<title>Un formulaire traité par Javascript</title>
<script language="javascript">
  function afficher(){
    // affiche dans une liste les infos du formulaire

    // on efface d'abord
    effacerInfos();

    // on affiche la valeur des # champs
    with(document.frmExemple){
      // champ caché
      ecrire("champ caché="+cache.value);
    }
  }
</script>
</head>
<body>
  <!-- Contenu du formulaire -->
</body>
</html>
```



```

        </td>
        <td>
            <input type="checkbox" name="qcm" value="deux">deux
        </td>
        <td>
            <input type="checkbox" name="qcm" value="trois">trois
        </td>
    </tr>
</table>
<table border="0">
    <tr>
        <td>Un menu déroulant : </td>
        <td>
            <select name="menu" size="1">
                <option>50 F</option>
                <option>60 F</option>
                <option>70 F</option>
                <option>100 F</option>
            </select>
        </td>
        <td>Une liste :</td>
        <td>
            <select name="lstVoitures" multiple size="3">
                <option>Renault</option>
                <option>Citroën</option>
                <option>Peugeot</option>
                <option>Fiat</option>
                <option>Audi</option>
            </select>
        </td>
    </tr>
</table>
<table border="0">
    <tr>
        <td>Un mot de passe : </td>
        <td><input type="password" size="21" name="passwd"></td>
        <td>&nbsp;</td>
        <td>Un champ de contexte caché : </td>
    </tr>
</table>
</form>
<hr>
<h2>Informations du formulaire</h2>
<form name="frmInfos">
    <table>
        <tr>
            <td><input type="button" value="Effacer" onclick="effacerInfos()"></td>
            <td>
                <select name="lstInfos" multiple size="3">
                    </select>
            </td>
            <td>
                <input type="button" name="cmdAfficher" value="Afficher" onclick="afficher()">
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

## 1.2 Les expressions régulières en Javascript

Côté navigateur, Javascript peut être utilisé pour vérifier la validité des données entrées par l'utilisateur avant de les envoyer au serveur. Voici un programme de test de ces expressions régulières.

### 1.2.1 La page de test

## Les expressions régulières en Javascript

Expression régulière	Chaîne de test
<input type="text" value="^(\\d+)(*)(\\d+)\$"/>	<input type="text" value="45abcd67"/>
<input type="button" value="Jouer le test"/>	<input type="button" value="Ajouter aux exemples"/>

### Résultats de l'instruction champs=expression régulière.exec(chaine)

```
Il y a correspondance  
champs[0]=[45abcd67]  
champs[1]=[45]
```

### Exemples

Modèles	Chaînes	<input type="button" value="Jouer l'exemple"/>
<input type="text" value="^(\\d+)(*)(\\d+)\$"/>	<input type="text" value="45abcd67"/>	

### 1.2.2 Le code de la page

```
<html>  
  
<head>  
<title>Les expressions régulières en Javascript</title>  
<script language="javascript">  
function afficherInfos() {  
  with(document.frmRegExp) {  
    // qq chose à faire ?  
    if (! verifier()) return;  
    // c'est bon - on efface les résultats précédents  
    effacerInfos();  
    // vérification du modèle  
    modele=new RegExp(txtModele.value);  
    champs=modele.exec(txtChaine.value);  
    if(champs==null)  
      // pas de correspondance entre modèle et chaîne  
      ecrireInfos("pas de correspondance");  
    else {  
      // correspondance - on affiche les résultats obtenus  
      ecrireInfos("Il y a correspondance");  
      for(i=0;i<champs.length;i++)  
        ecrireInfos("champs["+i+"]="+champs[i]+"");  
    } //else  
  } //with  
} //function  
  
function ecrireInfos(texte) {  
  // écrit texte dans la liste des infos  
  document.frmRegExp.lstInfos.options[document.frmRegExp.lstInfos.length]=new Option(texte);  
} //écrire  
  
function effacerInfos() {  
  frmRegExp.lstInfos.length=0;  
} //effacerInfos  
  
function jouer() {  
  // teste le modèle contre la chaîne dans l'exemple choisi  
  with(document.frmRegExp) {  
    txtModele.value=lstModeles.options[lstModeles.selectedIndex].text  
    txtChaine.value=lstChaines.options[lstChaines.selectedIndex].text  
    afficherInfos();  
  }  
}
```

```

    }//with
  }//jouer

  function ajouter(){
    //ajoute le test courant aux exemples
    with(document.frmRegExp){
      // qq chose à faire ?
      if (!verifier()) return;
      // ajout
      lstModeles.options[lstModeles.length]=new Option(txtModele.value);
      lstChaines.options[lstChaines.length]=new Option(txtChaine.value);
      // raz saisies
      txtModele.value="";
      txtChaine.value="";
    }//with
  }//ajouter

  function verifier(){
    // vérifie que les champs de saisie sont non vides
    with(document.frmRegExp){
      champs=/^\s*$/ .exec(txtModele.value);
      if(champs!=null){
        alert("Vous n'avez pas indiqué de modèle");
        txtModele.focus();
        return false;
      }//if
      champs=/^\s*$/ .exec(txtChaine.value);
      if(champs!=null){
        alert("Vous n'avez pas indiqué de chaîne de test");
        txtChaine.focus();
        return false;
      }//if
      // c'est bon
      return true;
    }//with
  }//verifier
</script>

```

```
</head>
```

```
<body bgcolor="#C0C0C0">
```

```
<center>
```

```
<h2>Les expressions régulières en Javascript</h2>
```

```
<hr>
```

```
<form name="frmRegExp">
```

```
<table>
```

```
<tr>
```

```
<td>Expression régulière</td>
```

```
<td>Chaîne de test</td>
```

```
</tr>
```

```
<tr>
```

```
<td><input type="text" name="txtModele" size="20"></td>
```

```
<td><input type="text" name="txtChaine" size="20"></td>
```

```
</tr>
```

```
<tr>
```

```
<td>
```

```
<input type="button" name="cmdAfficher" value="Jouer le test" onclick="afficherInfos()">
```

```
</td>
```

```
<td>
```

```
<input type="button" name="cmdAjouter" value="Ajouter aux exemples" onclick="ajouter()">
```

```
</td>
```

```
</tr>
```

```
</table>
```

```
<hr>
```

```
<h2>Résultats de l'instruction champs=expression régulière.exec(chaine)</h2>
```

```
<table>
```

```
<tr>
```

```
<td>
```

```
<select name="lstInfos" size="3">
```

```
</select>
```

```
</td>
```

```
</tr>
```

```
</table>
```

```
<hr>
```

```
<h2>Exemples</h2>
```

```
<table>
```

```
<tr>
```

```
<td align="center">Modèles</td>
```

```
<td align="center">Chaînes</td>
```

```
</tr>
```

```
<tr>
```

```
<td>
```

```

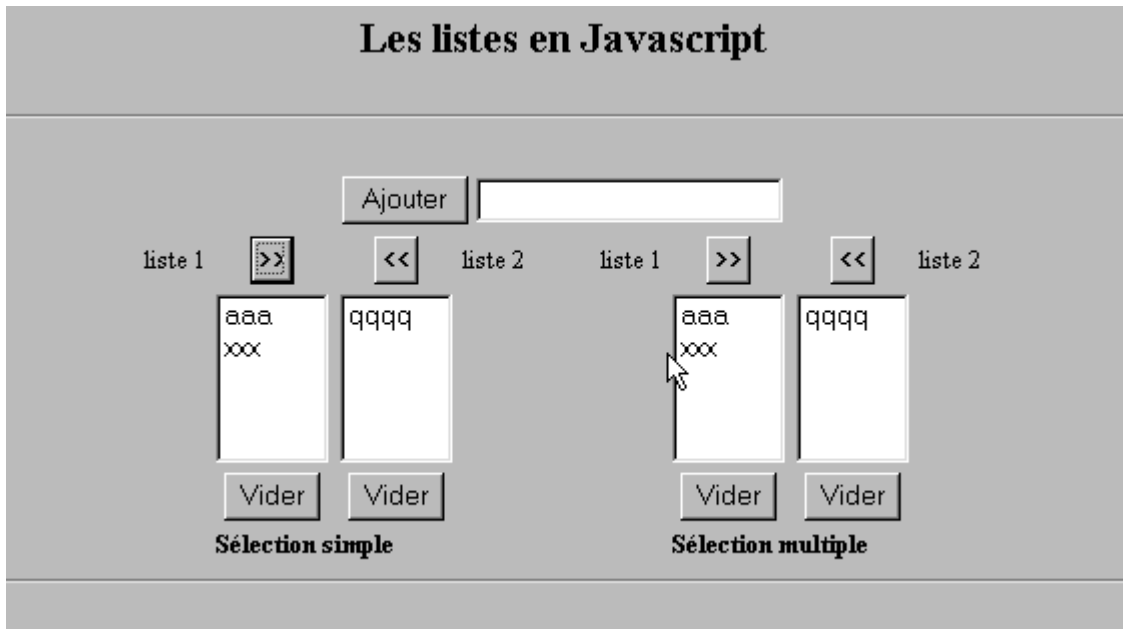
        <select name="lstModeles" size="1">
            <option>^\d+$</option>
            <option>^\d+ \d+$</option>
            <option>^\d+ (.*) \d+$</option>
            <option>^\d+ \s+ \d+$</option>
        </select>
    </td>
    <td>
        <select name="lstChaines" size="1">
            <option>67</option>
            <option>56 84</option>
            <option>45abcd67</option>
            <option>45 67</option>
        </select>
    </td>
    <td>
        <input type="button" name="cmdJouer" value="Jouer l'exemple" onclick="jouer()">
    </td>
</tr>
</form>
</body>
</html>

```

## 1.3 Gestion des listes en JavaScript

### 1.3.1 Le formulaire

# Les listes en Javascript



## 1.3.2 Le code

<html>

<head>

<title>Les listes en Javascript</title>

```
<script language="javascript">
// ajouter
function ajouter(L1,L2,T){
// ajoute la valeur du champ T aux listes L1,L2
// qq chose à faire ?
champs=/^\s*$/ .exec(T.value);
if(champs!=null){
// le champ est vide
alert("Vous n'avez pas indiqué la valeur à ajouter");
txtElement.focus();
return;
}
// on ajoute l'élément
L1.options[L1.length]=new Option(T.value);
L2.options[L2.length]=new Option(T.value);
T.value="";
}
//vider
function vider(L){
// vide la liste L
L.length=0;
}
//transfert
function transfert(L1,L2,simple){
//transfère dans L2 les éléments sélectionnés dans la liste L1
// qq chose à faire ?
// index de l'élément sélectionné dans L1
index1=L1.selectedIndex;
if(index1==-1){
alert("Vous n'avez pas sélectionné d'élément");
return;
}
// quel est le mode de sélection des éléments des listes
if(simple){ // sélection simple
element1=L1.options[index1].text;
//ajout dans L2
L2.options[L2.length]=new Option(element1);
//suppression dans L1
L1.options[index1]=null;
}
}
```



```

    if(! simple){ //sélection multiple
        //on parcourt la liste 1 en sens inverse
        for(i=L1.length-1;i>=0;i--){
            //élément sélectionné ?
            if(L1.options[i].selected){
                //on l'ajoute à L2
                L2.options[L2.length]=new Option(L1.options[i].text);
                //on le supprime de L1
                L1.options[i]=null;
            }
        }
    }
}
</script>
</head>

<body bgcolor="#C0C0C0">
<center>
<h2>Les listes en Javascript</h2>
<hr>
<form name="frmListes">
<table>
<tr>
<td>
<input type="button" name="cmdAjouter" value="Ajouter"
onclick="ajouter(lst1A,lst1B,txtElement)">
</td>
<td>
<input type="text" name="txtElement">
</td>
</tr>
</table>
<table>
<tr>
<td align="center">liste 1</td>
<td align="center"><input type="button" value="">> onclick="transfert(lst1A,lst2A,true)"</td>
<td align="center"><input type="button" value=""><< onclick="transfert(lst2A,lst1A,true)"</td>
<td align="center">liste 2</td>
<td width="30"></td>
<td align="center">liste 1</td>
<td align="center"><input type="button" value="">> onclick="transfert(lst1B,lst2B,false)"</td>
<td align="center"><input type="button" value=""><< onclick="transfert(lst2B,lst1B,false)"</td>
<td align="center">liste 2</td>
</tr>
<tr>
<td></td>
<td align="center">
<select name="lst1A" size="5">
</select>
</td>
<td align="center">
<select name="lst2A" size="5">
</select>
</td>
<td></td>
<td></td>
<td align="center">
<select name="lst1B" size="5" multiple >
</select>
</td>
<td align="center">
<select name="lst2B" size="5" multiple>
</select>
</td>
</tr>
<tr>
<td></td>
<td align="center"><input type="button" value="Vider" onclick="vider(lst1A)"</td>
<td align="center"><input type="button" value="Vider" onclick="vider(lst2A)"</td>
<td></td>
<td></td>
<td align="center"><input type="button" value="Vider" onclick="vider(lst1B)"</td>
<td align="center"><input type="button" value="Vider" onclick="vider(lst2B)"</td>
<td></td>
</tr>
<tr>
<td></td>
<td colspan="2"><strong>Sélection simple</strong></td>

```

```
<td></td>
<td></td>
<td></td>
<td colspan="2"><strong>Sélection multiple</strong></td>
<td></td>
</tr>
</table>
<hr>
</form>
</body>
</html>
```

<b><u>1</u> JAVASCRIPT.....</b>	<b>2</b>
<b><u>1.1</u> Récupérer les informations d'un formulaire.....</b>	<b>2</b>
<u>1.1.1</u> Le formulaire.....	2
<u>1.1.2</u> Le code.....	2
<b><u>1.2</u> Les expressions régulières en Javascript.....</b>	<b>4</b>
<u>1.2.1</u> La page de test.....	4
<u>1.2.2</u> Le code de la page.....	5
<b><u>1.3</u> Gestion des listes en JavaScript.....</b>	<b>7</b>
<u>1.3.1</u> Le formulaire.....	7
<u>1.3.2</u> Le code.....	8