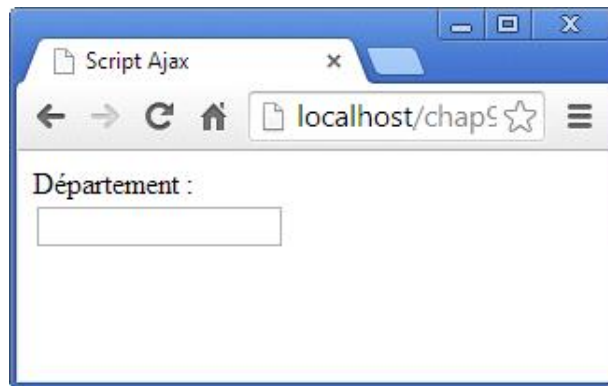


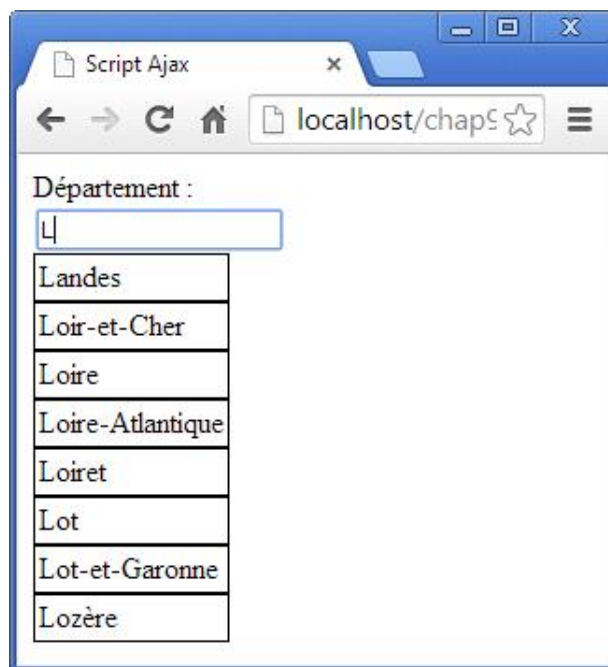
Une liste de suggestions

La philosophie de cet exemple est de présenter, lors de la saisie de données par l'utilisateur, des suggestions de valeurs.

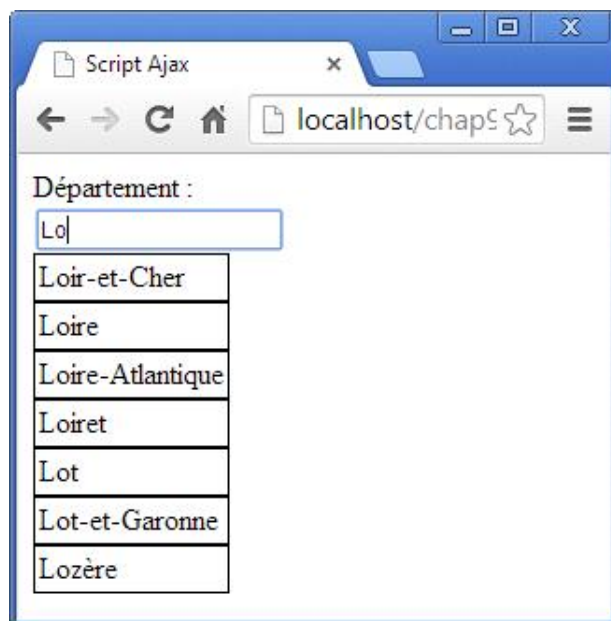
Soit une page web dans laquelle le visiteur peut saisir le nom de son département.



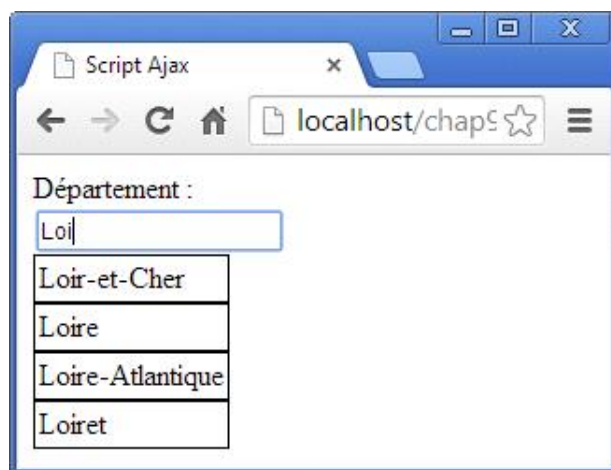
À la saisie clavier de la lettre "L", diverses propositions sont affichées dans un menu déroulant, soit tous les départements commençant par la lettre L : Landes, Loir-et-Cher, Loire, Loire-Atlantique, Loiret, Lot, Lot-et-Garonne, Lozère.



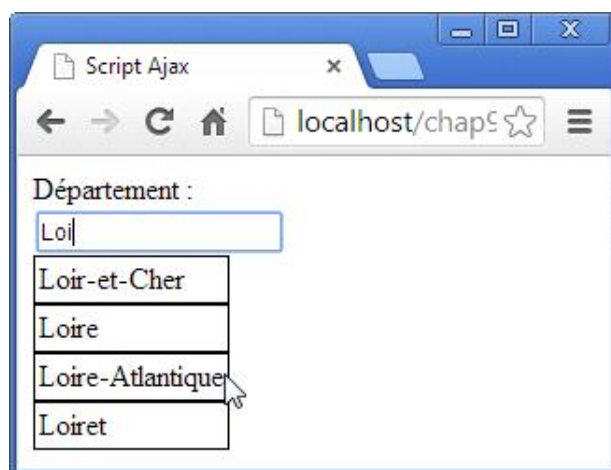
L'utilisateur poursuit la saisie et tape la lettre "o". Le menu déroulant ne présente alors que les départements commençant par "Lo", et ce instantanément, sans que la page soit rechargée.



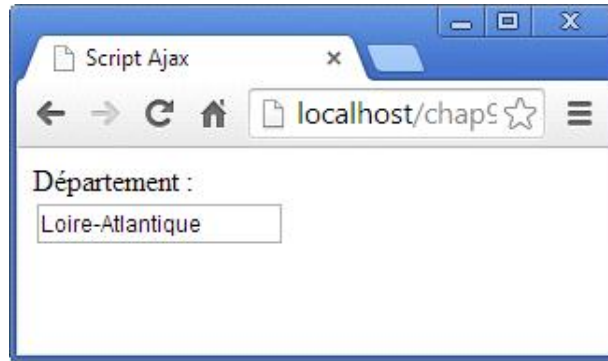
Continuons avec la saisie de la lettre "i". Une nouvelle modification du menu déroulant fait apparaître les départements commençant par les lettres "Loi" (Loir-et-Cher, Loire, Loire-Atlantique et Loiret).




Supposons qu'à ce stade le département de la Loire-Atlantique soit retenu.



Il suffit alors de cliquer sur le département de la Loire-Atlantique pour que celui-ci s'affiche automatiquement dans la zone de texte, en faisant disparaître, dans le même temps, la liste déroulante.



 Cet exemple illustre parfaitement l'apport du Web 2.0, et du concept AJAX en particulier, dans l'amélioration de la facilité et du confort d'utilisation du Web.

Au niveau du code, ce script comporte dans son ensemble :

- Un fichier XML, situé sur le serveur, qui reprend la liste des départements.
- Un fichier HTML avec, dans le cas présent, une simple ligne de texte.
- Un fichier JavaScript externe qui, après avoir récupéré la ou les lettre(s) saisie(s) dans la zone de texte, exécute une requête HTTP pour interroger le fichier XML, récupère les départements correspondants et les affiche sous la forme d'un menu pop-up déroulant.
- Un fichier de feuilles de style CSS externe prenant en charge la présentation du menu déroulant.

Commençons par le fichier XML (departements.xml).

L'arborescence de ce fichier se présente comme suit :

- L'élément racine du document est la balise `<choix> ... </choix>`.
- Pour chaque département, une balise `<item>` est prévue.
- Chaque item contient la balise `<dep> ... </dep>` pour le nom du département et la balise `<valeur> ... </valeur>` pour le numéro qui lui est associé.

```
<choix>
<item>
<dep> ... </dep>
<valeur> ... </valeur>
</item>
</choix>
```

Ce fichier, disponible en téléchargement depuis la page Informations générales, se présente comme suit :

```
<?xml version="1.0" encoding="UTF-8"?>
<choix xml:lang="FR">
<item><dep>Ain</dep><valeur>01</valeur></item>
<item><dep>Aisne</dep><valeur>02</valeur></item>
<item><dep>Allier</dep><valeur>03</valeur></item>
```

<item><dep>Alpes-de-Haute-Provence</dep><valeur>04</valeur></item>
 <item><dep>Hautes-Alpes</dep><valeur>05</valeur></item>
 <item><dep>Alpes-Maritimes</dep><valeur>06</valeur></item>
 <item><dep>Ardèche</dep><valeur>07</valeur></item>
 <item><dep>Ardennes</dep><valeur>08</valeur></item>
 <item><dep>Ariège</dep><valeur>09</valeur></item>
 <item><dep>Aube</dep><valeur>10</valeur></item>
 <item><dep>Aude</dep><valeur>11</valeur></item>
 <item><dep>Aveyron</dep><valeur>12</valeur></item>
 <item><dep>Bouches-du-Rhône</dep><valeur>13</valeur></item>
 <item><dep>Calvados</dep><valeur>14</valeur></item>
 <item><dep>Cantal</dep><valeur>15</valeur></item>
 <item><dep>Charente</dep><valeur>16</valeur></item>
 <item><dep>Charente-Maritime</dep><valeur>17</valeur></item>
 <item><dep>Cher</dep><valeur>18</valeur></item>
 <item><dep>Corrèze</dep><valeur>19</valeur></item>
 <item><dep>Côte-d'Or</dep><valeur>21</valeur></item>
 <item><dep>Côtes-d'Armor</dep><valeur>22</valeur></item>
 <item><dep>Creuse</dep><valeur>23</valeur></item>
 <item><dep>Dordogne</dep><valeur>24</valeur></item>
 <item><dep>Doubs</dep><valeur>25</valeur></item>
 <item><dep>Drôme</dep><valeur>26</valeur></item>
 <item><dep>Eure</dep><valeur>27</valeur></item>
 <item><dep>Eure-et-Loir</dep><valeur>28</valeur></item>
 <item><dep>Finistère</dep><valeur>29</valeur></item>
 <item><dep>Corse-du-Sud</dep><valeur>2A</valeur></item>
 <item><dep>Haute-Corse</dep><valeur>2B</valeur></item>
 <item><dep>Gard</dep><valeur>30</valeur></item>
 <item><dep>Gers</dep><valeur>31</valeur></item>
 <item><dep>Gironde</dep><valeur>33</valeur></item>
 <item><dep>Hérault</dep><valeur>34</valeur></item>
 <item><dep>Ille-et-Vilaine</dep><valeur>35</valeur></item>
 <item><dep>Indre</dep><valeur>36</valeur></item>
 <item><dep>Indre-et-Loire</dep><valeur>37</valeur></item>
 <item><dep>Isère</dep><valeur>38</valeur></item>
 <item><dep>Jura</dep><valeur>39</valeur></item>
 <item><dep>Landes</dep><valeur>40</valeur></item>
 <item><dep>Loir-et-Cher</dep><valeur>41</valeur></item>
 <item><dep>Loire</dep><valeur>42</valeur></item>
 <item><dep>Haute-Loire</dep><valeur>43</valeur></item>
 <item><dep>Loire-Atlantique</dep><valeur>44</valeur></item>
 <item><dep>Loiret</dep><valeur>45</valeur></item>
 <item><dep>Lot</dep><valeur>46</valeur></item>
 <item><dep>Lot-et-Garonne</dep><valeur>47</valeur></item>
 <item><dep>Lozère</dep><valeur>48</valeur></item>
 <item><dep>Maine-et-Loire</dep><valeur>49</valeur></item>
 <item><dep>Manche</dep><valeur>50</valeur></item>
 <item><dep>Marne</dep><valeur>51</valeur></item>
 <item><dep>Haute-Marne</dep><valeur>52</valeur></item>
 <item><dep>Mayenne</dep><valeur>53</valeur></item>
 <item><dep>Meurthe-et-Moselle</dep><valeur>54</valeur></item>
 <item><dep>Meuse</dep><valeur>55</valeur></item>
 <item><dep>Morbihan</dep><valeur>56</valeur></item>
 <item><dep>Moselle</dep><valeur>57</valeur></item>
 <item><dep>Nièvre</dep><valeur>58</valeur></item>

```

<item><dep>Nord</dep><valeur>59</valeur></item>
<item><dep>Oise</dep><valeur>60</valeur></item>
<item><dep>Orne</dep><valeur>61</valeur></item>
<item><dep>Pas-de-Calais</dep><valeur>62</valeur></item>
<item><dep>Puy-de-Dôme</dep><valeur>63</valeur></item>
<item><dep>Pyrénées-Atlantiques</dep><valeur>64</valeur></item>
<item><dep>Hautes-Pyrénées</dep><valeur>65</valeur></item>
<item><dep>Pyrénées-Orientales</dep><valeur>66</valeur></item>
<item><dep>Bas-Rhin</dep><valeur>67</valeur></item>
<item><dep>Haut-Rhin</dep><valeur>68</valeur></item>
<item><dep>Rhône</dep><valeur>69</valeur></item>
<item><dep>Haute-Saône</dep><valeur>70</valeur></item>
<item><dep>Saône-et-Loire</dep><valeur>71</valeur></item>
<item><dep>Sarthe</dep><valeur>72</valeur></item>
<item><dep>Savoie</dep><valeur>73</valeur></item>
<item><dep>Haute-Savoie</dep><valeur>74</valeur></item>
<item><dep>Paris</dep><valeur>75</valeur></item>
<item><dep>Seine-Maritime</dep><valeur>76</valeur></item>
<item><dep>Seine-et-Marne</dep><valeur>77</valeur></item>
<item><dep>Yvelines</dep><valeur>78</valeur></item>
<item><dep>Deux-Sèvres</dep><valeur>79</valeur></item>
<item><dep>Somme</dep><valeur>80</valeur></item>
<item><dep>Tarn</dep><valeur>81</valeur></item>
<item><dep>Tarn-et-Garonne</dep><valeur>82</valeur></item>
<item><dep>Var</dep><valeur>83</valeur></item>
<item><dep>Vaucluse</dep><valeur>84</valeur></item>
<item><dep>Vendée</dep><valeur>85</valeur></item>
<item><dep>Vienne</dep><valeur>86</valeur></item>
<item><dep>Haute-Vienne</dep><valeur>87</valeur></item>
<item><dep>Vosges</dep><valeur>88</valeur></item>
<item><dep>Yonne</dep><valeur>89</valeur></item>
<item><dep>Territoire-de-Belfort</dep><valeur>90</valeur></item>
<item><dep>Essonne</dep><valeur>91</valeur></item>
<item><dep>Hauts-de-Seine</dep><valeur>92</valeur></item>
<item><dep>Seine-Saint-Denis</dep><valeur>93</valeur></item>
<item><dep>Val-de-Marne</dep><valeur>94</valeur></item>
<item><dep>Val-d'Oise</dep><valeur>95</valeur></item>
</choix>

```

Le fichier HTML (departement.htm) est assez sommaire. Il comporte un formulaire uniquement composé d'une ligne de texte et d'un élément <div> ... </div> identifié par id="popups".

```

<!DOCTYPE html>
<html lang="fr">
<head>
<title>AJAX</title>
<meta charset="UTF-8">
<link rel="stylesheet" href="departements.css">
<script src="departements.js"></script>
</head>
<body>
<form>
Département :<br>

```

```
<input type="text" id="formulaire"><br>
<div id="popups"> </div>
</form>
</body>
</html>
```

Le fichier JavaScript est plus complexe à cause de la multiplicité des tâches qu'il doit gérer. Voici le code pas à pas de ce fichier :

```
window.onload = initAll;
var xhr = null;
var departements = new Array();
```

Comme il s'agit d'un fichier JavaScript externe, on initialise la fonction `initAll()` au chargement de la page (`onload`).

Après la déclaration de la variable `xhr`, un tableau est créé (`newArray()`) afin d'y insérer les différents départements.

```
function initAll() {
document.getElementById("formulaire").onkeyup = searchSuggest;
```

La fonction `initAll()` récupère d'abord l'action de l'utilisateur dans la ligne de texte du formulaire (`getElementById("formulaire")`). L'événement associé à celle-ci est de type `onkeyup`. Ainsi, à chaque lettre saisie, la fonction `searchSuggest()` sera appelée (voir plus loin dans le script).

Une requête HTTP peut alors être lancée sur le serveur.

```
if (window.XMLHttpRequest) {
xhr = new XMLHttpRequest();
}
else {
if (window.ActiveXObject) {
try {
xhr = new ActiveXObject("Microsoft.XMLHTTP");
}
catch (e) { }
}
}
```

Comme nous l'avons vu au chapitre relatif à l'objet `XMLHttpRequest`, ce bout de script distingue les procédures à mettre en œuvre pour les navigateurs Internet Explorer 7+, Firefox, Google Chrome (`xhr = XMLHttpRequest()`) et Internet Explorer 6, pour lequel un contrôle ActiveX est encore utilisé (`xhr = new ActiveXObject("Microsoft.XMLHTTP")`);

```
if (xhr) {
xhr.onreadystatechange = setdepartements;
xhr.open("GET", "departements.xml", true);
xhr.send(null);
}
```

```

else {
alert("Désolé, votre navigateur n'est pas compatible avec AJAX");
}
}

```

Si l'objet `xhr` a bien été créé (`if (xhr)`), la requête est initialisée, selon la méthode `GET`, pour lire le fichier *departements.xml* en mode asynchrone. La requête est ensuite effectuée par l'instruction `send(null)`. Dès qu'il y a des changements d'état de la requête (`onreadystatechange`), la fonction `setdepartements()` est appelée.

```

function setdepartements() {
if (xhr.readyState == 4) {
if (xhr.status == 200) {

```

Cette fonction `setdepartements()` teste d'abord si les données retournées sont complètement accessibles (`xhr.readyState == 4`), puis si la requête a été exécutée avec succès (`xhr.status == 200`).

```

if (xhr.responseXML) {
var tousdepartements = xhr.responseXML.getElementsByTagName("item");

```

Si l'objet `xhr.responseXML` existe, la variable `tousdepartements` contient, sous forme d'un document XML, toutes les balises `<item>` du fichier, répertoriées grâce au nom de la balise (`getElementsByTagName("item")`).

```

for (var i=0; i<tousdepartements.length; i++) {
departements[i] = tousdepartements[i].getElementsByTagName("dep")
[0].firstChild;
}
}
}
else {
alert("Il y a un problème avec la requête " + xhr.status);
}
}
}

```

Une boucle `for` va permettre de traiter chaque item un par un (`tousdepartements[i]`). La balise `<dep>` (`getElementsByTagName("dep")[0]`) est identifiée et son premier enfant est mémorisé, soit le nœud texte contenant le nom du département. Ce nom est stocké dans le tableau *departements*.

Les données du document XML ayant ainsi été récupérées, la fonction `searchsuggest()` va à présent les traiter.

```

function searchSuggest() {
var str = document.getElementById("formulaire").value;
document.getElementById("formulaire").className = "";

```

La variable `str` contiendra la ou les lettre(s) entrée(s) (`value`) par l'utilisateur dans la ligne de texte (`getElementById("formulaire")`).

À présent, considérons la propriété de style associée à la ligne de texte. À ce stade, rien n'est spécifié (`getElementById("formulaire").className = ""`).

```

if (str != "") {
document.getElementById("popups").innerHTML = "";

```

Si le contenu de la zone de texte n'est pas vide (`str != ""`), le contenu de la balise `<div id="popups">` du document HTML est initialisé. À cet instant, ce contenu est vide (`innerHTML = ""`).

```

for (var i=0; i<departements.length; i++) {
var ce_departement = departements[i].nodeValue;
if (ce_departement.toLowerCase().indexOf(str.toLowerCase()) == 0) {
var tempDiv = document.createElement("div");
tempDiv.innerHTML = ce_departement;
tempDiv.onclick = choix;
tempDiv.className = "suggestions";
document.getElementById("popups").appendChild(tempDiv);
}
}

```

Grâce à une boucle `for`, tous les éléments du tableau `departements` sont traités : il s'agit des données récupérées dans la fonction `setdepartements()` étudiée ci-avant (`departements[i]`). Pour chaque élément, la valeur du nœud (`nodeValue`) est retenue, soit le nom du département. Ce nom est stocké dans la variable `ce_departement`.

Puis, un test conditionnel vérifie si le contenu de la ligne de texte se trouve dans la ou les premières lettres du département en utilisant la méthode `indexOf` (`ce_departement.indexOf(str) == 0`). Par précaution, le nom du département et la variable `str` (`toLowerCase()`) sont transformés en lettres minuscules.

Si c'est le cas, un nouvel élément `<div>` (`createElement("div")`), ayant pour contenu le nom du département (`innerHTML = ce_departement`), est créé. Le cas dans lequel l'utilisateur choisit cet élément par un clic est prévu. Ce clic renvoie à la fonction `choix()` qui sera détaillée ultérieurement (`onclick = choix`). L'affichage de cet élément `<div>` est pris en charge par la propriété de style qui s'applique à la classe `suggestions` du fichier `departements.css` (`className = "suggestions"`).

Enfin, ce nouvel élément est inclus en dernière position dans la balise `<div id="popups">`, grâce à la méthode `appendChild()`.

```

var liste = document.getElementById("popups").childNodes.length;
if (liste == 0) {
document.getElementById("formulaire").className = "error";
}

```

Le script prévoit, en outre, l'éventualité d'une saisie qui n'a rien à voir avec le nom des départements, par exemple le caractère `@`. La variable `liste` contient le nombre des nœuds enfant créés dans la balise `<div id="popups">` par la boucle `for` précédente (`getElementById("popups").childNodes.length`). Dans le cas d'une saisie inadéquate (`liste == 0`), la feuille de style `error` est appliquée au formulaire.

```

if (liste == 1) {
document.getElementById("formulaire").value = document.getElementById("
popups").firstChild.innerHTML;
document.getElementById("popups").innerHTML = "";
}
}
}

```


Le script prévoit également le cas où il n'y aurait qu'un seul département possible à afficher à partir de la première lettre saisie dans la zone de texte. C'est le cas de la lettre J, le seul département ayant pour initiale J étant le Jura, de même pour la lettre F, le seul choix possible étant le Finistère. Dans cette éventualité, le script peut afficher directement le département.

Voici l'explication du code : si cette variable `liste` ne contient qu'un seul élément (`liste == 1`), JavaScript peut directement encoder dans le formulaire (`getElementById("formulaire").value`) le contenu de ce seul élément. Il est alors inutile d'afficher, ni même de faire apparaître, le menu déroulant (`getElementById("popups").innerHTML = ""`);

Il ne reste plus qu'à étudier la fonction `choix()`.

```
function choix(evt) {
var thisDiv = (evt) ? evt.target : window.event.srcElement;
document.getElementById("formulaire").value = thisDiv.innerHTML;
document.getElementById("popups").innerHTML = "";
}
```

La fonction `choix()` était, rappelons-le, associée à l'événement du clic correspondant au choix de l'utilisateur dans la liste des départements suggérés. La ligne `(evt) ? evt.target : window.event.srcElement;` assure la compatibilité avec Firefox et Internet Explorer : en effet les deux navigateurs ne gèrent pas de la même façon les événements. Au clic sur un élément du menu déroulant, la valeur de celui-ci (`thisDiv.innerHTML`) est retenue pour être mise dans la ligne de texte (`getElementById("formulaire").value`). Dans ce cas, le menu déroulant peut alors être effacé (`getElementById("popups").innerHTML = ""`).

Le script complet devient donc :

```
window.onload = initAll;
var xhr = null;
var departements = new Array();
function initAll() {
document.getElementById("formulaire").onkeyup = searchSuggest;
if (window.XMLHttpRequest) {
xhr = new XMLHttpRequest();
}
else {
if (window.ActiveXObject) {
try {
xhr = new ActiveXObject("Microsoft.XMLHTTP");
}
catch (e) { }
}
}
if (xhr) {
xhr.onreadystatechange = setdepartements;
xhr.open("GET", "departements.xml", true);
xhr.send(null);
}
else {
alert("Désolé, votre navigateur n'est pas compatible avec AJAX");
}
}
```

```

function setdepartements() {
if (xhr.readyState == 4) {
if (xhr.status == 200) {
if (xhr.responseXML) {
var tousdepartements = xhr.responseXML.getElementsByTagName("item");
for (var i=0; i<tousdepartements.length; i++) {
departements[i] = tousdepartements[i].getElementsByTagName("dep")
[0].firstChild;
}
}
}
else {
alert("Il y a un problème avec la requête " + xhr.status);
}
}
}

function searchSuggest() {
var str = document.getElementById("formulaire").value;
document.getElementById("formulaire").className = "";
if (str != "") {
document.getElementById("popups").innerHTML = "";
for (var i=0; i<departements.length; i++) {
var ce_departement = departements[i].nodeValue;
if (ce_departement.toLowerCase().indexOf(str.toLowerCase()) == 0) {
var tempDiv = document.createElement("div");
tempDiv.innerHTML = ce_departement;
tempDiv.onclick = choix;
tempDiv.className = "suggestions";
document.getElementById("popups").appendChild(tempDiv);
}
}
var liste = document.getElementById("popups").childNodes.length;
if (liste == 0) {
document.getElementById("formulaire").className = "error";
}
if (liste == 1) {
document.getElementById("formulaire").value =
document.getElementById("popups").firstChild.innerHTML;
document.getElementById("popups").innerHTML = "";
}
}
}

function choix(evt) {
var thisDiv = (evt) ? evt.target : window.event.srcElement;
document.getElementById("formulaire").value = thisDiv.innerHTML;
document.getElementById("popups").innerHTML = "";
}
}

```

Il reste à construire le fichier des propriétés de feuilles de style, soit *departement.css*.

```

.suggestions { background-color: #FFF;
padding: 2px 2px;
border: 1px solid #000;}

```

```
#popups { position: absolute;}
#formulaire { font: 9pt arial, helvetica, sans-serif;}
#formulaire.error { background-color: #FFC;}
```

Commentaires :

- Les suggestions du script (`.suggestions`) seront affichées avec un arrière-plan de couleur blanche et avec une fine bordure.
- Au survol du curseur, ces suggestions (`.suggestions`) apparaissent avec un arrière-plan de couleur.
- L'encodage dans la ligne de texte (`#formulaire`) se réalise en police Arial, 9 points.
- Si l'utilisateur saisit une lettre ou un caractère qui ne correspond pas à la liste des départements (`#formulaire.error`), la ligne de texte apparaît en jaune.