

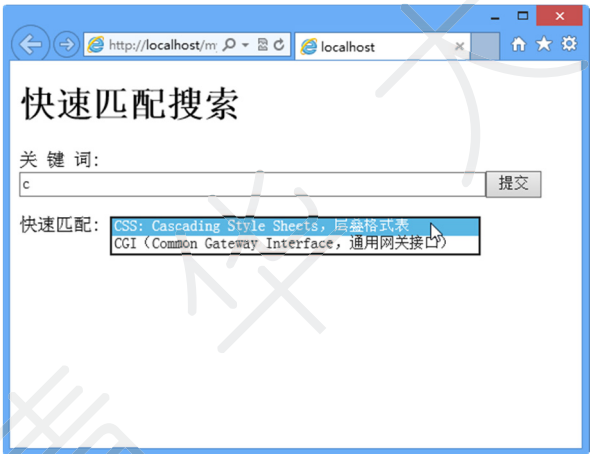
## 21.3.4 快速匹配搜索

使用百度搜索时，在搜索框中输入一个关键词，会立即弹出最热的相关关键词信息，如图 E21.4 所示。这种匹配提示能够帮助用户参考，并快速选择关键词，从技术上讲它就是使用 Ajax 技术来实现的。

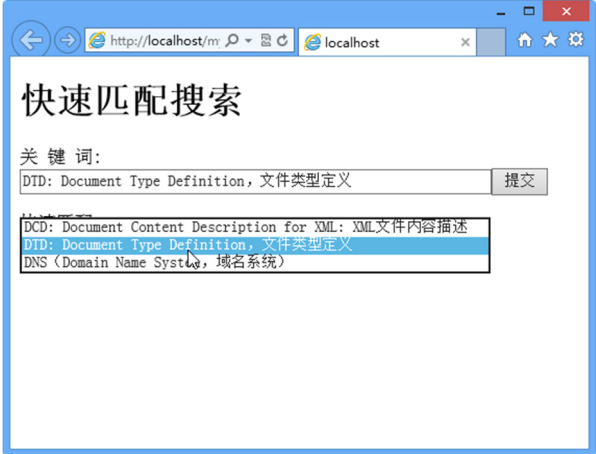


图 E21.4 百度搜索提示列表效果

设计思路：  
本示例设计当用户在文本框中输入关键字时，浏览器会自动从后台数据中查询匹配数据，并迅速显示在下面的下拉菜单中，以供输入选择，当从下面的下拉菜单中选择一项之后，选取结果会快速输入到上面文本框中，避免手动输入，演示效果如图 E21.5 所示。



输入字母 a



选择匹配的关键词

图 E21.5 快速匹配搜索列表

当输入关键词时，Ajax 快速将该词发送到后台，由后台与指定的数据进行比较，如果发现有匹配的信息，则组合成下拉列表的结构响应给客户端，最后由 JavaScript 脚本把这些文本信息插入到页面中。

**【操作步骤】**

第 1 步，构建后台数据库。由于本例仅是一个演示，这里仅使用一个数组来存储 20 条信息，代码如下（test.asp）：

```
dim a(20)
a(1)="CSS: Cascading Style Sheets, 层叠格式表"
a(2)="CGI (Common Gateway Interface, 通用网关接口)"
```

a(3)="DCD: Document Content Description for XML: XML 文件内容描述"  
a(4)="DTD: Document Type Definition, 文件类型定义"  
a(5)="HTML (HyperText Markup Language, 超文本标记语言) "  
a(6)="JVM: Java Virtual Machine, Java 虚拟机"  
a(7)="SGML: Standard Generalized Markup Language, 标准通用标记语言 "  
a(8)="XML: Extensible Markup Language (可扩展标记语言) "  
a(9)="XSL: Extensible Style Sheet Language (可扩展设计语言) "  
a(10)="DNS (Domain Name System, 域名系统) "  
a(11)="IMAP4: Internet Message Access Protocol Version 4, 第四版因特网信息存取协议 "  
a(12)="Internet (因特网) "  
a(13)="IP (Internet Protocol, 网际协议) "  
a(14)="MODEM (Modulator Demodulator, 调制解调器) "  
a(15)="POP3: Post Office Protocol Version 3, 第三版电子邮局协议"  
a(16)="RDF: Resource Description Framework, 资源描述框架"  
a(17)="SNMP (Simple Network Management Protocol, 简单网络管理协议) "  
a(18)="SMTP (Simple Mail Transfer Protocol, 简单邮件传输协议) "  
a(19)="VPN: virtual private network, 虚拟局域网"  
a(20)="WWW (World Wide Web, 万维网, 是因特网的一部分"

第2步, 设计后台脚本。后台程序是根据前台发过来的关键字为基础进行操作, 并上面的数组进行比较。然后把匹配的数组元素值返回。详细代码如下 (test.asp) :

```
Response.AddHeader "Content-Type","text/html;charset=gb2312" '定义响应信息的文本编码类型
q=request.querystring("q") '获取客户端发送过来的关键字
if len(q)>0 then '如果关键字不是空的, 则执行下面代码
    hint="" '定义变量
    for i=1 to ubound(a) '遍历数组元素
        '截取相同长度, 并把关键字全部转换为大写形式, 以方便比较
        x1=ucase(mid(q,1,len(q)))
        x2=ucase(mid(a(i),1,len(q)))
        if x1=x2 then '如果相匹配
            if hint="" then '如果是第一个匹配的元素值, 则直接赋值
                hint="<option value=""&a(i)&"">"&a(i)&"</option>"
            else '如果不是第一个匹配的元素值, 则追加
                hint=hint & "<option value=""&a(i)&"">"&a(i)&"</option>"
            end if
        end if
    next
end if
if hint="" then '最后判断如果 hin 为空, 说明没有匹配的元素
    response.write("<select><option>没有匹配对象</option></select>")
else '否则输出响应信息到客户端
    response.write("<select onBlur='ok(this)' onChange='ok(this)'>"&hint&"</select>")
end if
```

第3步, 回到前台, 设计前台结构和脚本。新建文档, 保存为 index.html。然后设计前台页面结构。

```
<h1>快速匹配搜索</h1>
<form>
    <label for="txt1">关 键 词:</label>
    <input name="txt1" type="text" id="txt1" onKeyUp="check(this.value)" size="60"><input name="" type="submit" value="
提交" />
</form>
<p>快速匹配: <span id="txtHint">
    <select>
        <option>请输入关键词</option>
```

```
</select>
</span></p>
</body>
```

#### 【提示】

这里面有两处需要着重说明：

- 为文本框绑定异步处理函数，并在函数中使用 **this** 关键字获取当前输出信息，并作为参数传递给服务器。设置事件类型为 **onKeyUp**，这样一旦输入一个字母（即当输入字符松开键时触发）就能够快速进行快速匹配。
- 使用 **innerHTML** 属性插入下拉选项时（**option** 元素），则 IE 存在一个 **Bug**，它不能够准确插入指定的选项，而是仅插入部分结构。例如，如果在 **<select>** 标签中插入 **<option>1</option>**，则 FF 浏览器能够准确插入，但是 IE 就仅插入 **"1</option"**，导致下拉列表结构无效。解决此类问题的方法是：由于 IE 在解析 **innerHTML** 存在小小的 **Bug**，如果要插入下拉选项，应该在下拉列表框（**<select>** 标签）外层包裹一个套子（如 **<span>**），然后为套子插入完整的下拉菜单结构，即同时插入 **select** 和 **option** 元素就能够有效解决这个问题。

第 4 步，定义异步处理函数，并设置参数为要传递的关键词。

```
function check(str){           //异步处理函数
    if (str.length > 0) {
//只有当传递的值字符串长度大于 0，则调用异步处理
        var url = "test.asp?q=" + str      //设置 URL 信息
        request.open("GET", url , true)    //打开连接
        request.onreadystatechange = updatePage; //绑定回调函数
        request.send(null)                 //发送异步请求
    }
}
```

第 5 步，定义回调函数。在本示例中没有再继续跟踪以请求过程。主要是在下拉菜单中不好进行显示。

```
function updatePage(){
    var info = document.getElementById("txtHint");
    if( request.readyState == 4 ){
        if( request.status == 200 ){
            info.innerHTML = request.responseText;
//把响应数据直接插入到文档中
        }else
            alert( request.status );
    }
}
```

第 6 步，定义一个小函数，当用户选择或激活匹配下拉选项，则自动把选择项的值填写到上面的文本框中。

```
function ok(o){
    var o1 = document.getElementById("ok1");
    document.getElementById("txt1").value = o.value;
}
```

第 7 步，在后台脚本中的输出下拉列表结构中绑定该函数，代码如下。这里定义了失去焦点和改动选项时都可以触发上面的函数。因为当下拉选项仅有一个选项时，**onchange** 事件类型是触发不了的。

```
if hint="" then
    response.write("<select><option>没有匹配对象</option></select>")
else
    response.write("<select onblur='ok(this)' onchange='ok(this)'>"&hint"&"</select>")
end if
```