

程序设计实践 2020 春

“

这是华东理工大学计算机系（计算机科学与工程、软件工程专业）的第二门实践课，在大一下开展。第一门实践课讲 HTML、CSS、JavaScript，大作业是做静态网站。可以参考清华学堂在线公开课：前端攻城狮。

程序设计实践 2020 春

课程介绍

ASP

VBScript 和 ASP

变量

过程

条件

循环

for ... to ... next

for each ... in ... next

do ... loop

#include

Cookies

Session, Application

AJAX

XMLHttpRequest

例子

Web Pages

课程介绍



程序设计综合实践

信息科学与工程学院

教师：翟洁

学分 2

学时：64h

成绩

《程序设计综合实践》是面向软件工程的必修实践课，是教学计划中理论联系实际的一个重要环节。通过本课...

内容

- HTML 表单和 ASP 编程

- 数据库设计
- 系统框架分析和实现
- 应用系统开发

参考资料

- w3schools
- 《网络程序设计——ASP》，尚俊杰
- 《数据库技术与应用基础——Access2010》，单颀 王芳

ASP

ASP 是 Active Server Pages 的缩写，是一种 Web 编程的开发框架，支持许多不同的开发模型

- 经典 ASP
- ASP.NET Web Forms
- ASP.NET MVC
- ASP.NET Web Pages
-

经典 ASP 是微软 1998 年推出的，是微软的第一个服务器端脚本语言，页面拓展名是 `.asp`，通常用 VBScript 写。

ASP.NET 是经典 ASP 的后继，在 2002 年推出，页面拓展名是 `.aspx`，通常用 C# 写。

这门课讲经典 ASP。

经典 ASP 可以用 JavaScript 写，例如

```
1  <%@ language="javascript"%>
2  <!DOCTYPE html>
3  <html>
4      <body>
5          <%
6              Response.Write("Hello World!")
7          %>
8      </body>
9  </html>
```

这门课用 VBScript 写。

VBScript 和 ASP

VBScript 可以用 `<% ... %>` 写在 `.asp` 文件里，也可以用 `<script language=vbscript runat=server>` 来写

```
1 <script language=vbscript runat=server>
2     response.write("hello")
3 </script>
```

下面用 `<%%>` 写。

• 变量

VBScript 是不区分大小写的，就算是变量名也不区分。

在 ASP 里，这样的代码

```
1 Dim val1
2 val2="Hello"
```

是不报错的，因为它自动定义了 `val2` 这个变量，我们可以用

```
1 <%@ language="vbscript"%>
2 <%Option Explicit '必须放在最前面 %>
3 <!DOCTYPE html>
4 <html>
5     <body>
6         <%
7             response.write("Hello World!")
8             Dim val1
9             val2="Hello"
10        %>
11     </body>
12 </html>
```

`Option Explicit` 就会禁止之前的自动定义，报错。

在 ASP 里可以定义一维和二维数组，索引从 `0` 开始

```
1 <%
2 Dim x(2,2)
3 x(0,0)="Volvo"
4 x(0,1)="BMW"
5 x(0,2)="Ford"
6 ...
7 for i=0 to 2
8     response.write("<p>")
```

```

9         for j=0 to 2
10             response.write(x(i,j) & "<br />")
11         next
12         response.write("</p>")
13     next
14 %>

```

在一个 ASP 页面内，变量的声明周期还是很长的

```

1 <%
2     if true then
3         Dim val1
4         val1="hello"
5     end if
6 %>
7 <%
8     response.write(val1)
9 %>

```

这样用没问题，但是如果要跨页面使用变量，就得定义成 Session 或者 Application 变量。

下面介绍了过程，你在一个过程内定义的变量，出了这个过程就没有了。Again，如果你没有使用 `Option Explicit`，虽然你引用了一个不存在的变量，也不会有任何的报错，程序会自动定义一个新的未经赋值的变量。

• 过程

可以用 `sub` 定义过程，用 `call` 调用

```

1 <%
2     sub sum(val1,val2)
3         response.write(val1+val2)
4     end sub
5 %>
6 <%
7     call sum(1,2)
8 %>

```

页面就会显示 `3`，也可以用 `sum 1,2` 调用。

也可以定义函数，如

```

1  <%
2      function sum(val1,val2)
3          sum=val1+val2
4      end function
5  %>
6  <%
7      =sum(1,2)
8  %>

```

效果和上一个例子一样。

如果用 JavaScript 写就是

```

1  <%
2      function sum(val1,val2){
3          Response.Write(val1+val2)
4      }
5  %>
6  <%
7      sum(1,2)
8  %>

```

这时查看页面的 HTML 是

```

1  <!DOCTYPE html>
2  <html>
3  <body>
4  3
5  </body>
6  </html>

```

这说明这个过程是在服务器端执行的，在客户端我们只能得到结果，和直接在 HTML 里写 JavaScript 是不同的。

上面这个例子，你也可以写成

```

1  <script language="javascript" runat="server">
2      function jsproc(num1,num2){
3          Response.Write(num1+num2)
4      }
5  </script>

```

`runat="server"` 这个属性也使得它在服务端执行，在客户端是看不见这段代码的。

• 条件

我们有三种条件

```
1  if ... then ... end if
2  if ... then ... (elif ... then...) end if
3  select case ... (case ... ...) end select
```

• 循环

- for ... to ... next

```
1  for i=0 to 10 (step 2) ... next
```

`step` 可以是负数, 例如

```
1  for i=10 to 0 step -2 ... next
```

在 for 循环内部可以写退出, 如

```
1  for i=1 to 10
2      response.write(i&"<br/>")
3      exit for
4  next
```

在输出 `1` 之后就退出循环了, 如果用 `if i=7 then exit if` 就可以在输出 `7` 之后退出循环。

- for each ... in ... next

```
1  dim i(20),x
2  i(0)=0
3  i(1)=1
4  i(2)=2
5  for each x in i
6      response.write(x&"<br/>")
7  next
```

因为 `i(3)` 和之后的元素都未赋值, 所以输出是

```
1  0<br/>1<br/>2<br/><br/><br/><br/><br/><br/><br/><br/>
```

- do ... loop

我们有好几种 do loop

```
1  do while i<10 ... loop
2  do ... loop while i<10
3  do until i=10 ... loop
4  do ... loop until i=10
```

do loop 也有退出，如

```
1  do until i=10
2      i=i-1
3      if i<10 then exit do
4  loop
```

• #include

最后，动态页面很重要的一个需求就是，把小的 ASP 页面插入到一个大页面里面去。例如第二周作业 2，在用户登录前后导航栏是不同的，可以这样实现

`index.asp` :

```
1  <% if logged=true then %>
2      <!--#include file="logged.asp"-->
3  <% else %>
4      <!--#include file="unlogged.asp"-->
5  <% end if %>
```

`logged.asp` :

```
1  <p>Welcome</p>
```

`unlogged.asp` :

```
1  <p>Log in please.</p>
```

在这个例子里，你可以把 `logged.asp` 改成 `logged.html`，因为它是完全静态的。

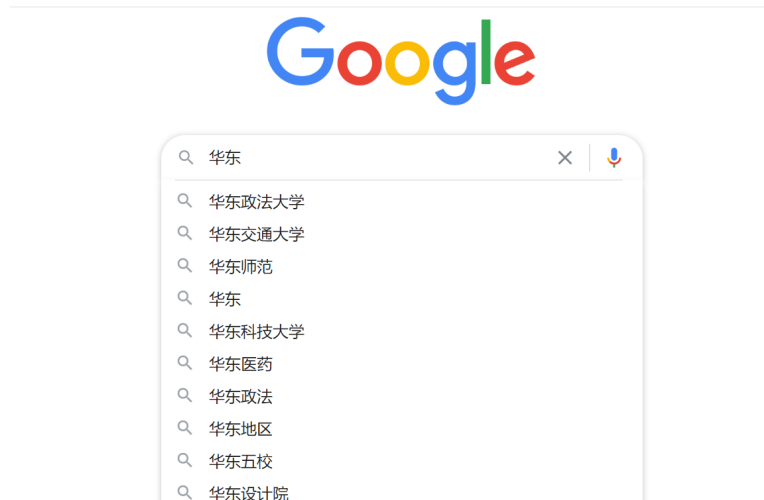
这种 #include 方法也非常安全，因为你在客户端打开调试工具也只能看到 `index.asp` 这一个文件，内容是 `<p>Welcome</p>`，看不到它具体来源于哪一个文件。

Cookies

Session, Application

在上一节的末尾，我们提到了第二周作业 2，

AJAX



AJAX 就是异步 JavaScript 和 XML 技术，在 2005 年用在了 Google 的搜索候选上从而流行起来。当你在搜索框输入到一半，还没有点击按钮或者敲击回车的时候，下方的列表里就会给出候选词。

首先我们知道，JavaScript 是可以操纵 HTML 标签的

```
1 <p id="text"></p>
2 <script>
3     document.getElementById("text").innerHTML = "hello"
4 </script>
```

我们可以用这种方法修改页面。

• XMLHttpRequest

下一个问题就是，如何在不刷新页面的情况下和服务器通信呢？为了实现这个功能，发明了 `XMLHttpRequest` 这个对象，我们可以用它发送 HTTP 请求

```
1 >> let xhttp = new XMLHttpRequest();
2 >> xhttp.open("GET", "asp2020/hw/1");
3 >> xhttp.send();
4 >> xhttp.response;
5 << "hello world"
```

查看这个请求的状态


```
1  >> xhttp.status;
2  << 200
```

200 意味着请求成功。

查看这个对象的状态

```
1  >> xhttp.readyState;
2  << 4
```

4 意味着请求已经发送，可以用 `response` 或者 `responseText` 属性查看回复信息了。

`XMLHttpRequest` 还有一个属性 `onreadystatechange`，是一个 Event Handler，当 `readyState` 发生变化的时候就会被调用。

• 例子

现在我们可以写一个 AJAX 的页面了，例如

`index.asp`：

```
1  <html>
2  <head>
3  <script>
4  function showHint(str) {
5      if (str.length == 0) {
6          document.getElementById("txtHint").innerHTML = "";
7          return;
8      } else {
9          var xmlhttp = new XMLHttpRequest();
10         xmlhttp.onreadystatechange = function() {
11             if (this.readyState == 4 && this.status == 200) {
12                 document.getElementById("txtHint").innerHTML =
13                 this.responseText;
14             }
15         };
16         xmlhttp.open("GET", "gethint.asp?q=" + str);
17         xmlhttp.send();
18     }
19 </script>
20 </head>
21 <body>
22 <form>
23 First name: <input type="text" onkeyup="showHint(this.value)">
24 </form>
25 <p>Suggestions: <span id="txtHint"></span></p>
26 </body>
```

```
27 </html>
```

gethint.asp :

```
1  <%
2  response.expires=-1
3  dim a(30)
4  a(1)="Anna"
5  a(2)="Brittany"
6  a(3)="Cinderella"
7  a(4)="Diana"
8  ...
9
10 'ucase() 这个函数把所有字母都变成大写
11 q=ucase(request.querystring("q"))
12
13 if len(q)>0 then
14     hint=""
15     for i=1 to 30
16         '对所有候选词遍历
17         if q=ucase(mid(a(i),1,len(q))) then
18             '如果 q 和该候选词的前几位相同
19             if hint="" then
20                 hint=a(i)
21             else
22                 hint=hint & ", " & a(i)
23             end if
24         end if
25     next
26 end if
27
28 if hint="" then
29     response.write("no suggestion")
30 else
31     response.write(hint)
32 end if
33 %>
```

Web Pages

“

按照顺序下面应该是 ASP 连接数据库。在涉及到数据库的 Web 编程中，推荐 MVC 设计模式，但是经典 ASP 是比较陈旧的开发模型。所以下面介绍 Web Pages。