

## 一、填空题（每题1分，共10分）

1. .NET框架中包括一个庞大的类库。为了方便调用，将其中的类按照命名空间进行逻辑分区。
2. DHTML的设计思想是：浏览器从服务器端下载文档后，利用浏览器的资源，在不增加服务器负担和网上传输流量的前提下，使网页的某些元素“动”起来。
3. 传统的布局方法是利用表格，网页布局的最佳方案是 div+CSS ？  
float+position+display  
flex
4. 请将下列字符串转变为浮点类型的数据，以便进行计算

```
1 string ss = "55.52";  
2 double dd = ;  
3
```

double.Parse(ss)

5. 一个简单的存储过程分为两部分，如下，其中第一部分是过程名、待定系数及其类型；第二部分是T-SQL语句

```
1 CREATEA PROCEDURE 存储过程名  
2 (  
3 )  
4 AS  
5 (  
6 )
```

6. 要使得表格的边框不显示，应设置border的值为0
7. 在HTML网页中，使用<a>标记来实现超链接
8. 在网页中嵌入多媒体，如电影、声音等用到的的标记是<embed>

## 二、选择题

1. 静态网页文件的后缀是 (C)  
A. asp B.aspx C.html D.jsp
2. 下列JS的判断语句中 (A) 是正确的  
A. if(1==0) B.if(i=0) C. if i==0 then D.if i=0 then
3. 下列哪一个表达式将返回假？ (B)

A. !(3<=1) B.(4>=4)&&(5<=2)

C. ("a"=="a")&&("c"!="d") D.(2<3)|| (3<2)

4. 如果想在网页显示后，动态地改变网页的标题 (C)

A. 是不可以的

B. 通过document.write("新的标题内容")

C. 通过document.title= "新的标题内容"

D. 通过document.changeTitle("新的标题内容")

5. 在HTML页面上编写Javascript代码时，应编写在(B)标签中间。

A. <javascript>

B. <script>

C. <head>

D. <body>

6. 分析代码，结果是

[9,13)

```
1 var mystring = "i am a student";
2 var a = mystring.substring(9,13);
3 document.write(a);
```

uden

7. 分析代码，结果是

```
1 var s1 = parseInt("101中学234")
2 document.write(s1);
```

101

8. 现在需要验证某个TextBox控件输入的年龄是否大于18且小于65，此时应该使用控件是 (BC)

A. CompareValidator

B. RequiredFieldValidator

C. RangeValidator

D. validationSummary

9. 用户控件是后缀为 (D) 的文件

D.ascx

10. 下面哪一项是换行符标签 (C)

A. <body> B. <font> C. <br> D. <p>

11. 常用的网页图像格式有 (C)

A. gif tiff B. tiff jpg C. gif jpg D. tiff png

12. CSS样式表不可能实现 (D) 功能

A. 将格式和结构分离 B. 一个CSS文件控制多个网页

- C. 控制图片的精确位置 (position) D. 兼容所有的浏览器
13. 若要是设计网页的背景图片为bg.jpg, 以下标记中, 正确的是 (A)
- A. `<body background="bg.jpg">` background-image
- B. `<body bground="bg.jpg">`
- C. `<body image="bg.jpg">`
- D. `<body bgcolor="bg.jpg">`
14. 以下标记中, 可用来产生滚动文字或图形的是 (A)
- A. `<Scroll>`
- B. `<Marquee>`
- C. `<TextArea>`
- D. `<video>`

### 三、问答题 (每题5分, 共20分)

1. 一个项目开发团队拟在Visual Studio .NET环境下开发Web应用系统, 请描述以团队开发的形式进行开发的一般过程。

#### 团队开发过程

在团队开发过程中, 一般的方法是项目组长采用前述的个人开发过程, 先生成一个解决方案。此解决方案中, 已将各种目录建好, 并分别规定每一个目录的用途, 例如哪些目录存放共享的图片文件, 哪些目录存放用户上传的文件, 哪一个开发者管理哪一个目录等等, 然后将建好的开发框架放到源代码控制服务器上; 接着“开发者1”、“开发者2”、...、“开发者n”通过“源代码控制”客户端软件获取相同的“解决方案”, 最后互相分工协作完成Web应用系统的开发。

2. ASP.NET的数据源控件有几种类型?各用于访问什么类型的数据?

- sqlDataSource: SQL关系数据库
- AccessDataSrouce: Access数据库
- ObjectDataSource: 业务对象数据
- XMLDataSource: XML文档数据
- SitemapDataSource: 站点地图提供程序的数据

以上的数据源控件, 其都派生于DataSourceControl, 如果要实现自己的自定义数据源控件, 则应使用 DataSourceControl 基类。

3. 一个SQL Server服务器的IP地址为“202. 206. 249.56”, 使用用户名“admin”和密码“ysu123”连接到该SQL Server服务器上的StuInfo数据库, 按以上条件创建一个连接对象。

```
1 String str="Data Source=
2 202.206.249.56;Database=StuInfo;Uid=admin;Pwd =ysu123"
3 System.Data.SqlClient.SqlConnection aa;
4 aa = new System.Data.SqlClient.SqlConnection();
5 aa.ConnectionString=str;
```

4. JavaScript程序可以嵌入HTML文件，请描述其三种常用嵌入方法。

- `<script></script>`

直接用script标签

- `<script src="">`

外部引用

- `<button onclick="alert('')">`

内联

a标签用的是href，script用的是src

## 四、程序阅读题（每题6分，共30分）

1. 阅读代码

```
1 var x
2 var y = null;
3 alert(x)
4 alert(y)
5 alert(x=y)
6 alert(x==y)
7 alert(y==x)
```

结果是？

undefined

null

null

true

true

2. 阅读代码

c#

```

1 public partial class WebForm3: System.Web.UI.Page{
2     public int num = 65108995;
3     public string name = "张强";
4     public string email = "zhangqiang@126.com";
5     protected void Page_Load(object sender, EventArgs e)
6     {
7         Page.DataBind();
8     }
9 }

```

XML

```

1 <body>
2     <form>
3         <div>
4             <b>姓名: <#name%></b><br/>
5             <b>邮箱: <#email%></b><br/>
6             <b>电话: <#num%></b><br/>
7         </div>
8     </form>
9 </body>

```

<# XXX %>, 它可以引用.cs文件中的代码的字段, 但这个字段必须初始化后, 在页面的Load事件中使用Page.DataBind方法来实现。

姓名: 张强

邮箱: zhangqiang@126.com

电话: 65108995

## 五、编程题 (每题10分, 共30分)

1. 补充完成Fn(num)函数的内容和页面代码, 要求能够弹出消息框提示"参数为15.计算结果为: 56"。备注: 计算结果是通过函数调用计算所得: Fn函数实现计算0--num之间的所有偶数之和。

[0,num) +=2

```

1 <html>
2     <head>
3         <title>算法示例</title>
4     </head>
5     <script>
6         function Fn(num) {

```

```

7         var i, res = 0;
8         for (i = 0; i < num; i += 2) {
9             res += i;
10        }
11        alert('参数为15.计算结果为: ' + res)
12
13    }
14    Fn(15)
15    </script>
16    </html>

```

2. 以下页面使用frameset 框架技术编写，“LOGO”区域显示的内容为logo.htm。“左侧栏目”区域显示的内容为nav.htm。“内容显示区”显示内容为mainbody.htm。“网站信息”区域显示内容为bottom.htm.请写出该页面中的关键代码。

```

1  <HTML>
2      <HEAD>
3          <TITLE>嵌套框架</TITLE>
4      </HEAD>
5      <FRAMESET rows="100,*,100">
6          <FRAME src="logo.htm" />
7          <FRAMESET cols="150,*">
8              <FRAME src="nav.htm"/>
9              <FRAME src="mainbody.htm"/>
10         </FRAMESET>
11         <FRAME src="bottom.htm" />
12     </FRAMESET>
13     <BODY>
14         <P>太遗憾了，您的浏览器不支持框架</P>
15     </BODY>
16 </NOFRAMES>
17 </FRAMESET>
18 </HTML>

```

## 知识点

### iframe

```

1  <frameset cols="25%,*,25%">
2      <frame src="frame_a.htm">
3      <frame src="frame_b.htm">
4      <frame src="frame_c.htm">
5  </frameset>

```

**考试必考，页面分两栏，左侧导航，右侧显示主内容。**

```

1  <frameset cols="20%,*">
2      <frame src="nav.html"></frame>
3      <frame src="content.html"></frame>
4  </frameset>

```

## nav

```
1 | <p>这个是nav</p>
```

## 主html

```
1 | <!--index.html-->
2 | <frameset cols="25%,*">
3 |   <frame src="nav.htm">
4 |   <frame src="content.htm">
5 |   <noframes>浏览器不支持! </noframes>
6 | </frameset>
```

```
1 | <!--nav.html-->
2 | <ul>
3 |   <li>导航栏</li>
4 |   <li>导航栏</li>
5 |   <li>导航栏</li>
6 |   <li>导航栏</li>
7 | </ul>
```

```
1 | <!--content.html-->
2 | <div>
3 |   <h1>
4 |   内容
5 |   </h1>
6 | </div>
```

```
1 | <HTML>
2 | <HEAD><TITLE>嵌套框架</TITLE></HEAD>
3 | <FRAMESET rows="100,*,100">
4 |   <FRAME src="logo.htm" >
5 |   <FRAMESET cols="150,*">
6 |     <FRAME src="nav.htm">
7 |     <FRAME src="mainbody.htm">
```

```
8 </FRAMESET>
9 <FRAME src="bottom.htm" >
10 <NOFRAMES>
11 <BODY>
12 <P>太遗憾了，您的浏览器不支持框架！ /P>
13 </BODY>
14 </NOFRAMES>
15 </FRAMESET>
16 </HTML>
```

## CSS

---

- CSS可集中进行样式管理
- 把显示内容和样式定义相分离
- 一个HTML可以加载多个CSS，一个CSS也可以被多个HTML加载
- 它能够为网页上的元素精确地定位

---

CSS的嵌入方式有四种

内联

内嵌

链接

- 内联

```
1 <span style="color: red;">内联样式</span>
```

- 内嵌

```
1 <style type="text/css">
2   span{
3     color: red;
4   }
5 </style>
6 <span>内嵌样式</span>
```



- 链接

```
1 | <link rel="stylesheet" href="" type="text/css">
```

- 导入

```
1 | <style>
2 |   @import 某某某;
3 |   @import ' ';
4 |   @import "";
5 |
6 |   @import url();
7 |   @import url(' ');
8 |   @import url("");
9 | </style>
```

---

### CSS的好处

- 简化了网页的格式代码，外部的样式表还会被浏览器保存在缓存里，加快了下载显示的速度，也减少了需要上传的代码数量。
- 只要修改保存着网站格式的CSS样式表文件就可以改变整个站点的风格特色，在修改页面数量庞大的站点时，显得格外有用。避免了逐个修改网页，大大减少了重复劳动的工作量。

## JavaScript

### null与undefined

- 数字进行四则运算的时候，会隐式转换成数值类型。

```
1 | console.log("12"-1); //11
2 | console.log("12"*1); //12, 这个可以用于快速地转换成数值。
```

- `number` 和 `bigint` 可以隐式转换。

```
1 | 10 == 10n
```

- 数值和`null`相加会变成数值本身，因为`null`会隐式转换成0。而和`undefined`相加会变成NaN。

```
1 | console.log(1+null) //1
2 | console.log(1+undefined) //NaN
```

- 对一个数据使用`+`运算符，会自动转型到`number`

```
1 | var str = "string"
2 | str = +str; //转型到number
3 | console.log(str)
```

## null和undefined的区别\*

1. 作者在设计js的都是先设计的null (为什么设计了null:最初设计js的时候借鉴了java的语言)
2. null会被隐式转换成0, 很不容易发现错误。
3. 先有null后有undefined, undefined会转为NaN, 出来undefined是为了填补之前的坑。

具体区别: JavaScript的最初版本是这样区分的: null是一个表示"无"的对象(空对象指针), 转为数值时为0;  
undefined值, 转为数值时为NaN。

```
1 | var bestAge = null;
2 | var muchTooOld = 3 * bestAge;
3 | alert(bestAge);    //null
4 | alert(muchTooOld); //0
```

```
1 | var currentCount;
2 | var finalCount = 1 * currentCount;
3 | alert(currentCount);    //undefined
4 | alert(finalCount);     //NaN
```

## 隐式声明?

JavaScript支持隐式声明, 即可以不用 var 关键字声明变量, 例如:

```
1 | noStringAtAll = ""; // 隐式声明变量noStringAtAll
```

不能使用未经过声明的变量。例如:

```
1 | var volume = length * width; // 错误! length 和 width 不存在
```

原理是什么?

如果所使用的变量未被定义，有可能触发ReferenceError异常。

是否触发跟LHS和RHS有关系。

如果你直接对一个未声明的变量使用RHS，那么查询会一直抵达全局作用域，如果还没有找到，那么就会抛出异常。

```
1 | console.log(a)
```

如果你对一个未声明的变量使用LHS，那么在“非严格模式”下，引擎就会在全局作用域下创建这个变量。

```
1 | function addA(){  
2 |     a=1;  
3 | }  
4 | addA();
```

编译器会沿着作用域链一直查找a的定义，结果直到全局作用域都没有找到，所以最后在全局作用域创建了一个a。此时这个a就是全局作用域的变量了。

## 类型转换

```
1 | 55 + parseInt("45") // 100  
2 | 55 + parseInt("45AB") // 100  
3 | 55 + parseInt("A45B") // NaN  
4 | 55 + parseInt("0xFF") // 310  
5 | 55 + parseFloat("45.05") // 100.05
```

## 阶乘的代码

[1,5]

5

5\*4\*3\*2\*1

```

1 function factor(num)
2 {   var i,fact=1;
3     for (i=1;i<num+1;i++)
4         fact=i*fact;
5     return fact;
6 }
7 alert("5的阶乘="+ factor(5));

```

## DOM

HTML DOM中提供了统一访问HTML元素的六种方法

```

1 window.document.all.item("HTML元素的ID")
2 // 例如: window.document.all.item("myColor")
3
4 window.document.all.HTML元素的ID
5 // 例如: window.document.all.myColor
6
7 window.document.getElementById("HTML元素的ID")
8 // 例如: window.document.getElementById("myColor")
9
10 window.document.getElementsByName("HTML元素的Name属性值")
11 // 例如: window.document.getElementsByName("firstName")
12
13 window.document.all.namedItem("HTML元素的Id或Name属性值")
14 // 例如: window.document.all.namedItem ("myColor")
15
16 window.document.getElementsByTagName("HTML标记名称")
17 // 例如: window.document.getElementsByTagName("div")
18

```

但是，实际上开发中真正使用的是：`document.querySelector`

### 定时器

`setInterval`、`clearInterval`方法以及`setTimeout`、`clearTimeout`方法均可实现定时器功能。

`prompt`、`alert`、`confirm`方法实现对话框功能

```

1 var test=window.prompt("请输入数据:");
2 var YorN=confirm("你输入的数据是"+test+"，确定吗?");
3 if (YorN) alert("输入正确!");
4 else alert("输入不正确!");

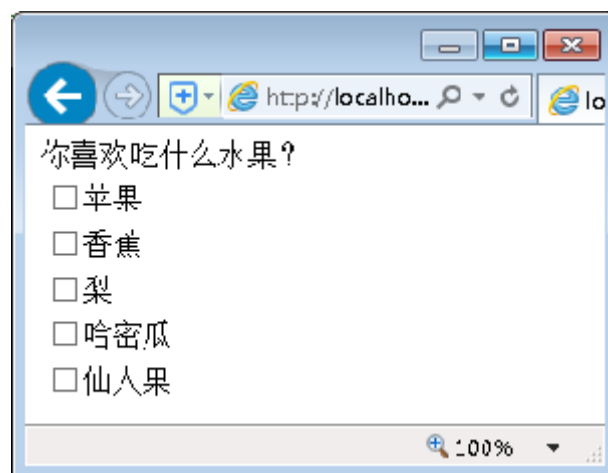
```

## Dot Net

- CheckBoxList: 复选框
- RadioButtonList: 单选框
- DropDownList: 下拉框
- ListBox: 选择框

```
var fruit = [fno,fname,fscore]
```

```
1 DataSet ds = new DataSet();
2 ds.Tables.Add("fruit");
3 ds.Tables["fruit"].Columns.Add("fNo", typeof(int));
4 ds.Tables["fruit"].Columns.Add("fName", typeof(string));
5 ds.Tables["fruit"].Columns.Add("fScore", typeof(int));
6 ds.Tables["fruit"].Rows.Add(new object[] { 1, "苹果", 100 });
7 ds.Tables["fruit"].Rows.Add(new object[] { 2, "香蕉", 100 });
8 ds.Tables["fruit"].Rows.Add(new object[] { 3, "梨", 100 });
9 ds.Tables["fruit"].Rows.Add(new object[] { 4, "哈密瓜", 100 });
10 ds.Tables["fruit"].Rows.Add(new object[] { 5, "仙人果", 100 });
11 //绑定数据到ListBox控件
12 this.CheckBoxList1.DataSource = ds.Tables["fruit"];
13 this.CheckBoxList1.DataValueField = "fNo";
14 this.CheckBoxList1.DataTextField = "fName";
15 this.CheckBoxList1.DataBind();
16
```

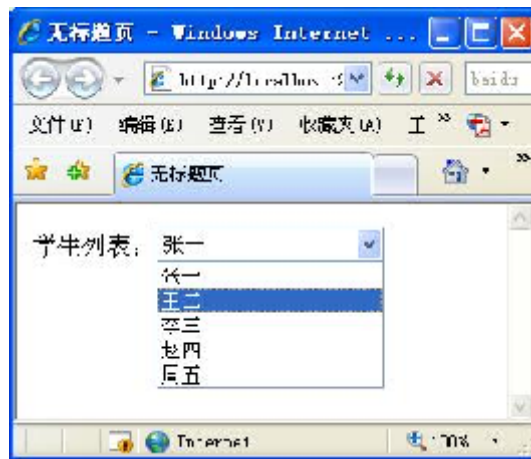


- ☐ 123
- ☐ 123
- ☐ 123
- ☐ 123
- ☐ 123
- ☐ 123123
- ☐ 123

```

2 DataSet ds = new DataSet();
3 ds.Tables.Add("stu");
4 ds.Tables["stu"].Columns.Add("stuNo", typeof(int));
5 ds.Tables["stu"].Columns.Add("stuName", typeof(string));
6 ds.Tables["stu"].Columns.Add("stuAge", typeof(int));
7 ds.Tables["stu"].Rows.Add(new object[] { 1, "张一", 20 });
8 ds.Tables["stu"].Rows.Add(new object[] { 2, "王二", 21 });
9 ds.Tables["stu"].Rows.Add(new object[] { 3, "李三", 19 });
10 ds.Tables["stu"].Rows.Add(new object[] { 4, "赵四", 19 });
11 ds.Tables["stu"].Rows.Add(new object[] { 5, "周五", 20 });
12 //绑定数据到ListBox控件
13 this.DropDownList1.DataSource = ds.Tables["stu"];
14 this.DropDownList1.DataValueField = "stuNo";
15 this.DropDownList1.DataTextField = "stuName";
16 this.DropDownList1.DataBind();
17

```



1. DataSet ds = new DataSet();
2. ds.Tables.Add("名字");
3. ds.Tables["名字"].columns.Add("自定义的字段名",类型)
4. ds.Tables["名字"].row.Add(new object[] { 1, "乒乓球", 100 })

```

1 //数据生成
2 DataSet ds = new DataSet();
3 ds.Tables.Add("ball");
4 ds.Tables["ball"].Columns.Add("bNo", typeof(int));
5 ds.Tables["ball"].Columns.Add("bName", typeof(string));
6 ds.Tables["ball"].Columns.Add("bScore", typeof(int));
7 ds.Tables["ball"].Rows.Add(new object[] { 1, "乒乓球", 100 });
8 ds.Tables["ball"].Rows.Add(new object[] { 2, "篮球", 100 });
9 ds.Tables["ball"].Rows.Add(new object[] { 3, "排球", 100 });
10 ds.Tables["ball"].Rows.Add(new object[] { 4, "羽毛球", 100 });
11 ds.Tables["ball"].Rows.Add(new object[] { 5, "足球", 100 });
12 //绑定数据到ListBox控件
13 this.RadioButtonList1.DataSource = ds.Tables["ball"];

```

```

14 this.RadioButtonList1.DataValueField = "bNo";
15 this.RadioButtonList1.DataTextField = "bName";
16 this.RadioButtonList1.DataBind();
17

```



```

1 using System.Data;
2 protected void Button1_Click(object sender, EventArgs e)
3 {
4     //数据生成
5     DataSet ds = new DataSet();
6     ds.Tables.Add("stu");
7     ds.Tables["stu"].Columns.Add("stuNo", typeof(int));
8     ds.Tables["stu"].Columns.Add("stuName",
9     typeof(string));
10    ds.Tables["stu"].Columns.Add("stuScore",
11    typeof(int));
12    ds.Tables["stu"].Rows.Add(new object[] { 1, "张一",
13    100 });
14    ds.Tables["stu"].Rows.Add(new object[] { 2, "王二",
15    100 });
16    ds.Tables["stu"].Rows.Add(new object[] { 3, "李三",
17    100 });
18    ds.Tables["stu"].Rows.Add(new object[] { 4, "赵四",
19    100 });
20    ds.Tables["stu"].Rows.Add(new object[] { 5, "周五",
21    100 });
22    //绑定数据到ListBox控件
23    this.ListBox1.DataSource = ds.Tables["stu"];
24    this.ListBox1.DataValueField = "stuNo";
25    this.ListBox1.DataTextField = "stuName";
26    this.ListBox1.DataBind();
27 }

```

张一 ▼

127.0.0.1

链接数据库Northwind

sa

docman

```
1 SqlConnection conn = new SqlConnection();  
2 Conn.ConnectionString="Data Source = 127.0.0.1; Initial Catalog =  
Northwind; Persist Security Info = True; User ID = sa; Password =  
docman ";  
3
```