ASP.NET动态网页设计

执教：朱亚林

南通师范高等专科学校
Nantong Normal College

C#基础知识

C#基础知识

- C#简介
- C# 程序结构
- 变量与常量
- 类型的转换
- 运算符与表达式
- 字符串处理
- 流程控制
- 数组

C#字符串处理

C#字符串处理

- ASP.NET中提供了String类来进行字符串的处理。其使用方法与定义变量一致。如下例：

```
string str1=”abc”;
```

比较字符串

- 字符串内容对比是比较常见的操作。如：在设计用户登录时，获取到用户名和密码，将它们的值与数据库中的记录进行比对时，就需要使用比较字符串的方法。
- String类提供了一系列的方法用于字符串的比较，如CompareTo和Equals等。

CompareTo

- 使用CompareTo方法进行字符串比较时，一般采用以下方式：
- String1.CompareTo(String2)
- 如果该表达式返回的值为0，则表示两值相等；如果前值大于后值，则返回1，如果前值小于后值，则返回-1。

```
string str1=”abc”;
int m1= str1.CompareTo(“ab”);
int m2= str1.CompareTo(“abc”);
int m3= str1.CompareTo(“abcd”);
```

Equals

- Equals方法用于确定两个String对象是否具有相同的值。如果相同则返回true，否则返回false。

```
string str1=“abc”;
bool f1=str1.Equals(“Abc”);
bool f2=str1.Equals(“abc”);
```

定位字符及子串

- 定位字符串中某个字符或子字符第一次出现的位置，可以使用IndexOf方法。具体如：string1.IndexOf(string2)。
- string1为被检索对象，而string2为需要检索的对象。即在string1中查询string2。
- 当查询到该string2时，返回string2第一次出现的位置；当查询不到时，返回-1，当string2为空时，返回0。

```
string str1=“abc”;
int m1= str1.IndexOf(“b”);
int m2= str1.IndexOf(“d1”);
```

截取字符串

- 截取字符串是指从已知字符串中确定一个起始位置和截取长度，将其间的内容截取出来。
- 截取字符串使用的函数是SubString(start,length)

```
string str1 = ”Hello World”;
string str2 = str1.SubString(0,5);
```

分裂字符串

- 分裂字符串是指，将一个字符串对象按照某个指定分隔符分裂成一系列小的字符串的操作。
- 分裂字符串使用的函数是：Split(Char[])

```
string str1 = "Hello,World";
string[] split_result = str1.Split(new Char[]{'.','!'});
foreach (string s in split)
{
if(s.Trim()!="")
    Console.WriteLines(s);
}
```

插入字符串

- 插入字符串是指在已有字符串的指定位置加入新的字符串内容。
- 插入字符串使用的函数是：Insert(Int,String);

```
string str1 = ”This is a girl”;
string str2 = str1.Insert(10,”beautiful”);
```

填充字符串

- 填充字符串是通过添加指定的对象实现右对齐或左对齐。这个填充对象可以是空格，也可以是指定的字符。
- 字符串填充一般使用PadLeft(Int,Char)或者PadRight(Int,Char)方法来进行。参数中的Int表示填充后字符串的长度，Char表示填充的内容。

```
string str1 = ”Hello World”;
string str2 = str1.PadLeft(15,’@’);
```

删除字符串

- 删除字符串是指在一个字符串的指定位置删除指定的字符。
- 删除字符串使用Remove(Int,Int)函数，其中第1个int指定开始删除的位置，第二个int指定要删除的字符数量。

```
string str1 = "This is a beautiful girl.";
string str2 = str1.Remove(10,10);
```

清除指定字符

- 清除指定字符是指把一个字符串首尾处的一些特殊字符清除掉。
- 清除指定字符可以使用Trim(Char[])、TrimStart(Char[])、TrimEnd(Char[])等方法。

```
string str1 = "_*Hello,world*_";
string str2 = str1.Trim('*','_');
```

复制字符串

复制字符串是指将一个字符串复制到另一个字符串中。其方法较多，这里介绍的是Copy(String)方法。

```
string str1 = "This is a beautiful girl.";
string str2 = String.Copy(str1);
```

替换字符串

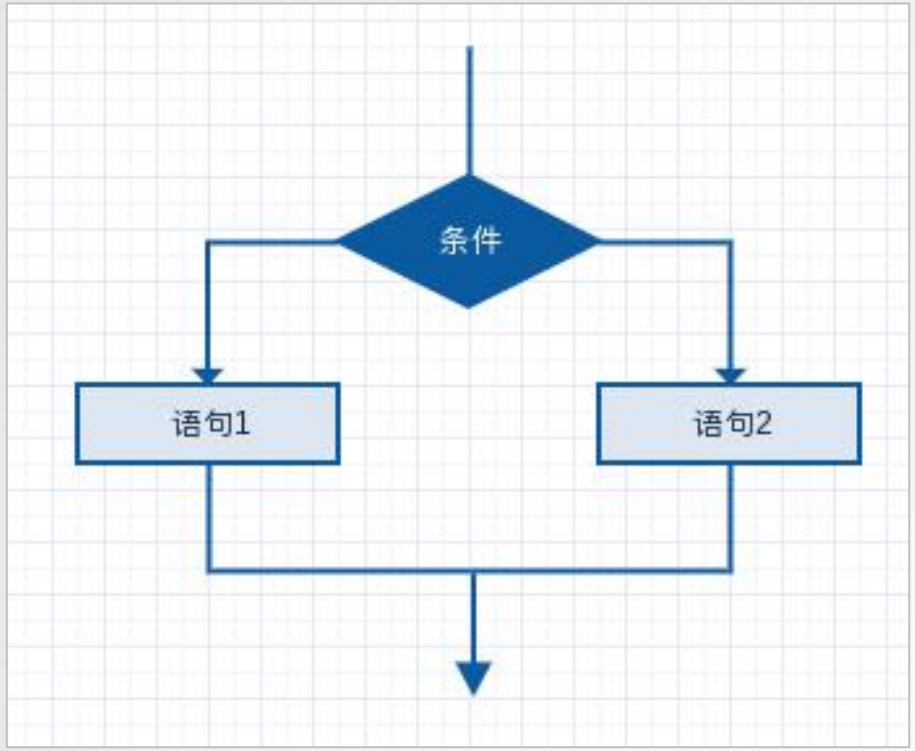
- 替换字符串是指替换掉一个字符串中的某些特定字符或者子串。
- 替换字符串使用的方法是：Replace(String String)。其中第1个参数为待替换的子串，第2个参数为替换后的新子串。即从字符串中找到第1个参数所示内容，替换成第2个参数中的内容。

```
string str = “This is a dog.”;
string str1 = str.Replace(“dog”,”pig”);
```

C#流程控制

分支语句

顺序结构的程序虽然能解决计算、输出等问题，但不能做判断再选择。对于要先做判断再选择的问题就要使用分支结构。分支结构的执行是依据一定的条件选择执行路径，而不是严格按照语句出现的物理顺序。




```
static void Main(string[] args)
{
    if (condition)
    {
        //statement 1
    }
    else
    {
        //statement 2
    }

}
```

分支语句

if...else

if...else语句是控制在某个条件下才执行某个功能，否则执行另外一个功能。

```
static void Main(string[] args)
{
    if (condition)
    {
        //statement 1
    }
    else
    {
        //statement 2
    }

}
```

如果语句块只有一条语句，则可以不使用{}，如果语句块中有多条语句，则必须使用{}。

分支语句

if...else示例

要求：

制作一个评判分数的程序，

80分及以上显示优秀，

60分及以上显示为及格，

60以下显示为不及格。

```
int score = Convert.ToSingle(Console.ReadLine());
string str = "";
if(score>=80)
    str = "优秀";
else if(score>=60)
    str = "及格";
else
    str= "不及格";
Console.WriteLine(str);
```

分支语句

if...else练习

要求：

打车起步价（3公里内）为10元，超出3公里的部分，每公里为4元，不足1公里按1公里计算。

请输入一个里程，计算打车价格。

分支语句

- switch语句
- switch语句是一种选择控制结构。一个 switch 语句允许测试一个变量等于多个值时的情况。每个值称为一个 case，且被测试的变量会对每个 switch case 进行检查。

```
switch(expression){
    case constant-expression  :
        statement(s);
        break;
    case constant-expression  :
        statement(s);
        break;
    /* 可以有任意数量的 case 语句 */
    default : /* 可选的 */
        statement(s);
        break;
}
```

分支语句

switch示例

要求：

制作一个评判分数的程序，

80分及以上显示优秀，

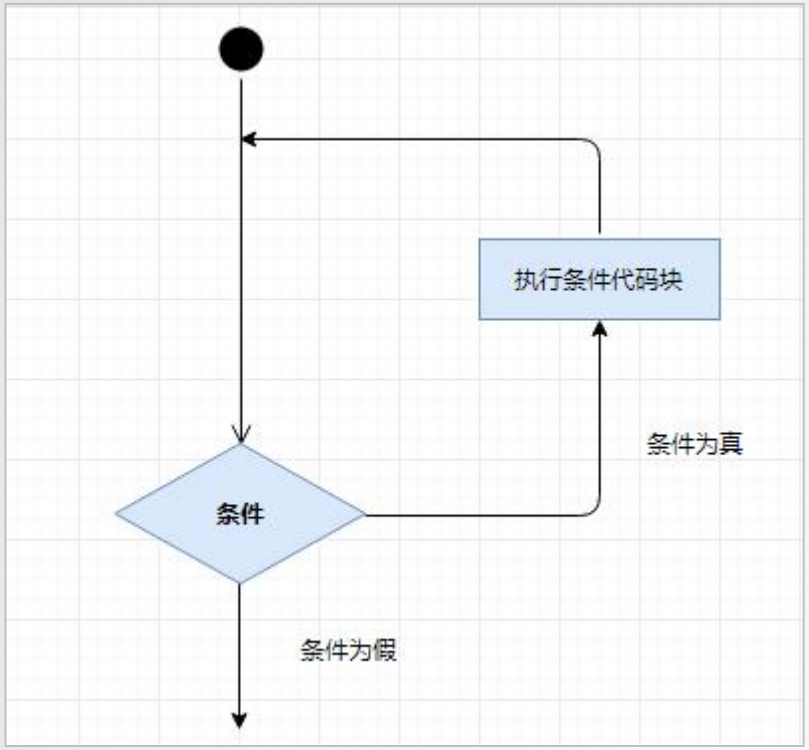
60分及以上显示为及格，

60以下显示为不及格。

```
double score = Convert.ToSingle(Console.ReadLine());
int s = (int)score/10;
string str="";
switch(s){
    case 10:
    case 9:
    case 8  :
        str="优秀";
        break;
    case 7:
    case 6:
        str="及格";
        break;
    default :
        str="不及格";
        break;
}
Console.WriteLine(str);
```

循环语句

循环结构是在一定条件下反复执行某段程序的流程结构，被反复执行的程序被称为循环体。循环语句是由循环体及循环的终止条件两部分组成的。



循环语句

for语句

for语句循环重复执行一个语句或语句块，直到指定的表达式计算为false。

```
static void Main(string[] args)
{
    int sum=0;
    for(int i=0;i<=100;i++)
        sum+=i;
    Console.WriteLine(sum);
}
```

for语句的执行顺序：

1. 计算变量的初始值(int i=0)；
2. 当布尔表达式的值为true时，执行代码段内容，并且重新计算变量的值；
3. 当布尔表达式的值为false时，结束循环向下执行

循环语句

while语句

while语句用来在指定条件内重复执行一个语句或语句块。

```
static void Main(string[] args)
{
    int sum=0;
    int i=0;
    while(i<=100)
        sum+=i++;
    Console.WriteLine(sum);
}
```

循环语句

do..while语句

do...while语句也是用来在指定条件内重复执行一个语句或语句块，它与while语句唯一不同之处在于，对于循环体内的内容至少会执行一次。

```
static void Main(string[] args)
{
    int i=1;
    do{
        i++;
    }while(i<1);
    Console.WriteLine(i);
}
```

循环语句

foreach语句

foreach语句提供一种简单、明了的方法来循环访问数组的元素。

```
static void Main(string[] args)
{
    string[] str={"Jack","Helen","Tom"};
    foreach(string s in str)
        Console.WriteLine(s);
}
```

C#数组

数组

- 数组是包含若干相同类型的变量的集合，这些变量可以通过索引进行访问。数组的索引从0开始。数组中的变量称为数组的元素。数组中的每个元素都具有唯一的索引与其相对应。数组能够容纳元素的数量称为数组的长度。
- 数组有一维、多维和交错数组。

数组的声明

- 数组可以具有多个维度。一维数组即数组的维数为1。声明一维数组的语法为：

```
type[] arraryName;
```

- 二维数组即数组的维数为2,它相当于一个表格。声明二维数组的方法为：

```
type[,] arraryName;
```

数组的初始化

- 数组的初始化有很多形式，可以通过new运算符创建数组并将数组元互初始化为它们的默认值。

```
static void Main(string[] args)
{
    int[] arr1 = new int[5]; //arr数组中的每个元素都初始化为0
    int[,] arr2 = new int[4,2];
    //初始化时也可以直接赋值
    int[] arr3 = new int[5]{1,2,3,4,5};
    int[] arr4 = new int[3,2]{{1,2},{3,4},{5,6}};
}
```

注意：数组大小必须与大括号内的元素个数相匹配，否则会产生编辑错误。

数组的赋值

- 在数组声明时可以不对其初始化，但在对数组初始化时，必须使用new运算符。

```
string[] arrStr;
arrStr = new string[7]{"Sun","Mon","Tue","Wed","Thu","Fri","Sat"};
int[,] arr;
arr = new int[,]{{1,2},{3,4},{5,6}};
```

```
static void Main(string[] args)
{
    string[] arrStr={"Sun","Mon","Tue","Wed","Thu","Fri","Sat"};
    int[,] arr = {{1,2},{3,4},{5,6}};
}
```

数组的遍历

- C#中的foreach语句可以方便实现对数组的遍历。参见循环一节的内容。