# 二维数组

Var arr1=[1,2,3,4];

Var arr2=[5,6,7,8];

Var arr3=[9,2,1];

Var arr=[arr1,arr2,arr3];

Var arr=[

[1,2,3,4],

[5,6,7,8],

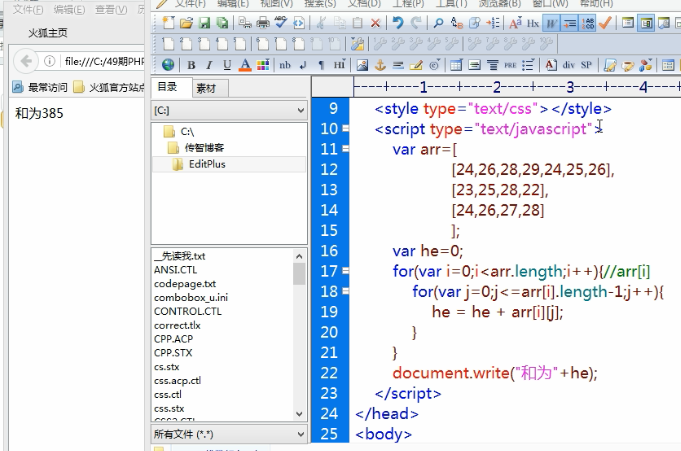
[9,2,1]

];

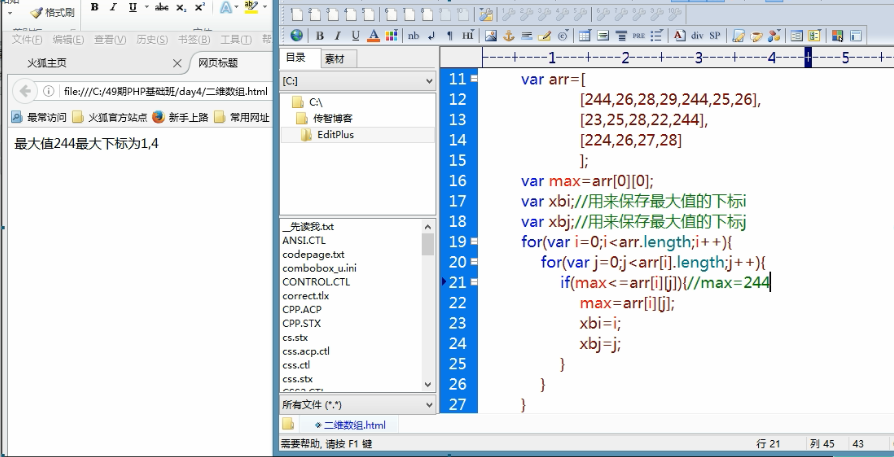
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 第一行 | 1[0][0] | 2[0][1] | 3[0][2] | 4[0][3] |
| 第二行 | 5[1][0] | 6[1][1] | 7[1][2] | 8[1][3] |
| 第三行 | 9[2][0] | 2[2][1] | 1[2][2] |  |

## 练习

求一个三行四列数组的所有项的和



求最大值及最大值的下标



# 系统内部的函数

parseInt():取整（如果字符串开头第一个是字母就会出现NaN）

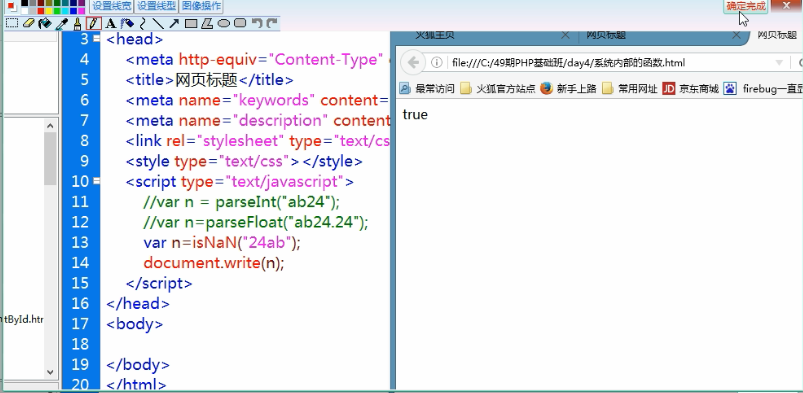
parseFloat:取浮点数

NaN:not a number 不是一个数字

isNaN:is not a number

1.如果判断的数据为数字:false

2.如果判断的数据为字符: true



# Array对象

Var arr = [22,24,26];

Var arr = new Aarray(22,24,26);

## 属性

Length

Arr.length:获取数组的长度

## 方法

Concat(): 连接内容或者数组，组成新的数组

Join(): 加入某个字符n,用n连接数组的每项组成字符串，如果以空字符串连接，就不会加入新的内容，直接把数组的每项内容连接起来

Pop(): 删除数组的最后一项并且返回

Push(): 加入新的内容返回新的长度

Reverse(): 反转数组的顺序



## 属性和方法的一般形式

属性：对象.属性名；

方法：对象.方法名();

# String对象

Var str=”abcd”;

Var str=new String(“abcd”);

## 属性

Length:

对象.length;

## 方法

charAt(n): 找到索引位置上的字符

charCodeAt(n): 找到索引位置上的字符的编码值(a:97 A:65)

indexOf(“m”): 找m在字符串中首次出现的位置，如果没有找到就返回-1

lastIndexOf(“m”): 找m在字符串中最后一次出现的位置，如果没有找到就返回-1

split(“n”): 以字符n分割字符串，返回一个数组

substr(n,m): 截取字符串—把字符串从n的位置开始截取，截取m个，如果只有一个参数n，那么就是从n开始截取，截取到最后

substring(n,m): 截取字符串—把字符串从n的位置开始截取，截取到m的位置，注意n的位置的字符可以取到，m的未知的字符取不到，如果只有一个参数n，就是从n开始截取，截取到最后

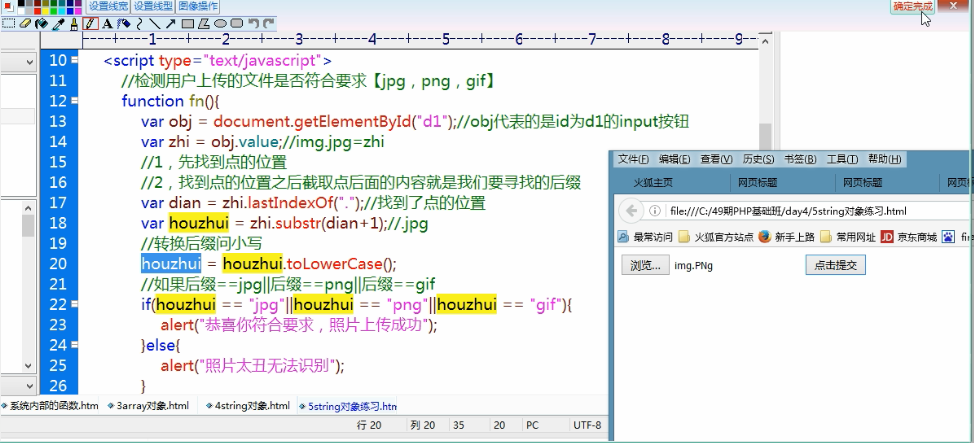
toLowerCase(): 把字符串中的字母转换成小写形式

toUpperCase(): 把字符串中的字母转换成大写形式



## 练习

检测用户上传的文件是否符合要求



# Math对象

Math.pow(n,m); 求n的m次方

Math.abs(n); 求n的绝对值

Math.round(n); 求n四舍五入的值

Math.floor(n); 求n的向下取整的值——不大于n的最大整数

Math.ceil(n); 求n的向上取整的值——不小于n的最小整数

Math.randon(); 获得一个0到1之间的随机数，注意：能取到0,但是不能取到1

1.求0到20之间的随机数：Math.random()\*20

2.求0到30之间的随机数：Math.random()\*30

求0到n之间的随机数：Math.random()\*n

a.求10到20之间的随机数： Math.random()\*10+10

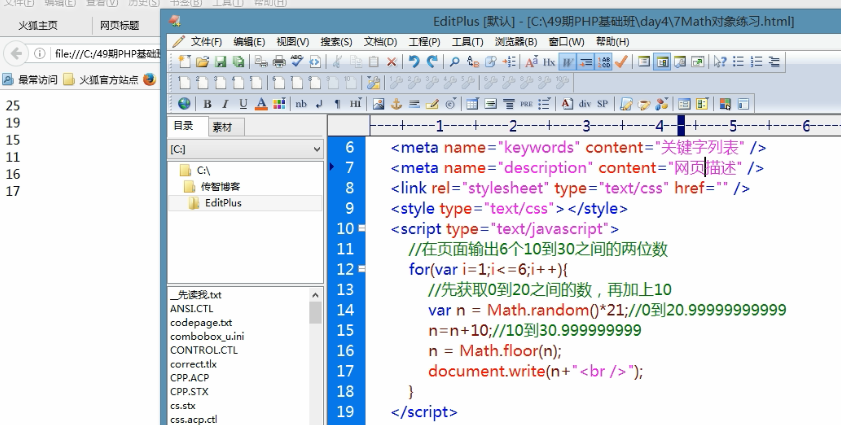
b.求30到60之间的随机数： Math.random()\*30+30

求n到m之间的随机数（n<m）：Math.random()\*(m-n)+n



## 练习

1.在页面输出6个10到30之间的随机数



## 公式

求任意两个数之间的随机数：

Math.floor(Math.random()\*(大数-小数+1)+小数)

# JS特殊字符

‘: 单引号

“: 双引号

\: 转义字符

\r: 回车

\n: 换行

