# Js实用技巧

1. 实现base64解码  
function base64\_decode(data){  
    var b64 = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=";  
    var o1, o2, o3, h1, h2, h3, h4, bits, i = 0,ac = 0,dec = "",tmp\_arr = [];  
    if (!data) { return data; }  
    data += '';  
    do {  
        h1 = b64.indexOf(data.charAt(i++));  
        h2 = b64.indexOf(data.charAt(i++));  
        h3 = b64.indexOf(data.charAt(i++));  
        h4 = b64.indexOf(data.charAt(i++));  
        bits = h1 << 18 | h2 << 12 | h3 << 6 | h4;  
        o1 = bits >> 16 & 0xff;  
        o2 = bits >> 8 & 0xff;  
        o3 = bits & 0xff;  
        if (h3 == 64) {  
            tmp\_arr[ac++] = String.fromCharCode(o1);  
        } else if (h4 == 64) {  
            tmp\_arr[ac++] = String.fromCharCode(o1, o2);  
        } else {  
            tmp\_arr[ac++] = String.fromCharCode(o1, o2, o3);  
        }  
    } while (i < data.length);  
    dec = tmp\_arr.join('');  
    dec = utf8\_decode(dec);  
    return dec;  
}  
2. 实现utf8解码  
  
  
function utf8\_decode(str\_data){  
    var tmp\_arr = [],i = 0,ac = 0,c1 = 0,c2 = 0,c3 = 0;str\_data += '';  
    while (i < str\_data.length) {  
        c1 = str\_data.charCodeAt(i);  
        if (c1 < 128) {  
            tmp\_arr[ac++] = String.fromCharCode(c1);  
            i++;  
        } else if (c1 > 191 && c1 < 224) {  
            c2 = str\_data.charCodeAt(i + 1);  
            tmp\_arr[ac++] = String.fromCharCode(((c1 & 31) << 6) | (c2 & 63));  
            i += 2;  
        } else {  
            c2 = str\_data.charCodeAt(i + 1);  
            c3 = str\_data.charCodeAt(i + 2);  
            tmp\_arr[ac++] = String.fromCharCode(((c1 & 15) << 12) | ((c2 & 63) << 6) | (c3 & 63));  
            i += 3;  
        }  
    }  
    return tmp\_arr.join('');  
}  
3. 半角转换为全角函数  
  
function ToDBC(str){  
    var result = '';  
    for(var i=0; i < str.length; i++){  
        code = str.charCodeAt(i);  
        if(code >= 33 && code <= 126){  
            result += String.fromCharCode(str.charCodeAt(i) + 65248);  
        }else if (code == 32){  
            result += String.fromCharCode(str.charCodeAt(i) + 12288 - 32);  
        }else{  
            result += str.charAt(i);  
        }  
    }  
    return result;  
}  
4. 全角转换为半角函数  
  
  
function ToCDB(str){  
    var result = '';  
    for(var i=0; i < str.length; i++){  
        code = str.charCodeAt(i);  
        if(code >= 65281 && code <= 65374){  
            result += String.fromCharCode(str.charCodeAt(i) - 65248);  
        }else if (code == 12288){  
            result += String.fromCharCode(str.charCodeAt(i) - 12288 + 32);  
        }else{  
            result += str.charAt(i);  
        }  
    }  
    return result;  
}  
5. 用正则表达式清除html代码中的脚本  
  
  
function clear\_script(html){  
    return html.replace(/<script.\*?>[\s\S]\*?<\/script>|\s+on[a-zA-Z]{3,16}\s?=\s?"[\s\S]\*?"|\s+on[a-zA-Z]{3,16}\s?=\s?'[\s\S]\*?'|\s+on[a-zA-Z]{3,16}\s?=[^ >]+/ig,"");  
}  
6. 获取当前元素样式  
  
function getStyle(oElm, strCssRule){  
    var strValue = "";  
    if(document.defaultView && document.defaultView.getComputedStyle){  
        strValue = document.defaultView.getComputedStyle(oElm, "").getPropertyValue(strCssRule);  
    }  
    else if(oElm.currentStyle){  
        strCssRule = strCssRule.replace(/\-(\w)/g, function (strMatch, p1){  
            return p1.toUpperCase();  
        });  
        strValue = oElm.currentStyle[strCssRule];  
    }  
    return strValue;  
}  
7. 格式化CSS样式代码  
  
function formatCss(s){//格式化代码  
    s = s.replace(/\s\*([\{\}\:\;\,])\s\*/g, "$1");  
    s = s.replace(/;\s\*;/g, ";"); //清除连续分号  
    s = s.replace(/\,[\s\.\#\d]\*{/g, "{");  
    s = s.replace(/([^\s])\{([^\s])/g, "$1 {\n\t$2");  
    s = s.replace(/([^\s])\}([^\n]\*)/g, "$1\n}\n$2");  
    s = s.replace(/([^\s]);([^\s\}])/g, "$1;\n\t$2");  
    return s;  
}  
8. 压缩CSS样式代码  
function compressCss (s) {//压缩代码  
    s = s.replace(/\/\\*(.|\n)\*?\\*\//g, ""); //删除注释  
    s = s.replace(/\s\*([\{\}\:\;\,])\s\*/g, "$1");  
    s = s.replace(/\,[\s\.\#\d]\*\{/g, "{"); //容错处理  
    s = s.replace(/;\s\*;/g, ";"); //清除连续分号  
    s = s.match(/^\s\*(\S+(\s+\S+)\*)\s\*$/); //去掉首尾空白  
    return (s == null) ? "" : s[1];  
}  
9. 常用的正则表达式  
//正整数  
/^[0-9]\*[1-9][0-9]\*$/;  
//负整数  
/^-[0-9]\*[1-9][0-9]\*$/;  
//正浮点数  
/^(([0-9]+\.[0-9]\*[1-9][0-9]\*)|([0-9]\*[1-9][0-9]\*\.[0-9]+)|([0-9]\*[1-9][0-9]\*))$/;  
//负浮点数  
/^(-(([0-9]+\.[0-9]\*[1-9][0-9]\*)|([0-9]\*[1-9][0-9]\*\.[0-9]+)|([0-9]\*[1-9][0-9]\*)))$/;  
//浮点数  
/^(-?\d+)(\.\d+)?$/;  
//email地址  
/^[\w-]+(\.[\w-]+)\*@[\w-]+(\.[\w-]+)+$/;  
//url地址  
/^[a-zA-z]+://(\w+(-\w+)\*)(\.(\w+(-\w+)\*))\*(\?\S\*)?$/;  
或：^http:\/\/[A-Za-z0-9]+\.[A-Za-z0-9]+[\/=\?%\-&\_~`@[\]\':+!]\*([^<>\"\"])\*$  
//年/月/日（年-月-日、年.月.日）  
/^(19|20)\d\d[- /.](0[1-9]|1[012])[- /.](0[1-9]|[12][0-9]|3[01])$/;  
//匹配中文字符  
/[\u4e00-\u9fa5]/;  
//匹配帐号是否合法(字母开头，允许5-10字节，允许字母数字下划线)  
/^[a-zA-Z][a-zA-Z0-9\_]{4,9}$/;  
//匹配空白行的正则表达式  
/\n\s\*\r/;  
//匹配中国邮政编码  
/[1-9]\d{5}(?!\d)/;  
//匹配身份证  
/\d{15}|\d{18}/;  
//匹配国内电话号码  
/(\d{3}-|\d{4}-)?(\d{8}|\d{7})?/;  
//匹配IP地址  
/((2[0-4]\d|25[0-5]|[01]?\d\d?)\.){3}(2[0-4]\d|25[0-5]|[01]?\d\d?)/;  
//匹配首尾空白字符的正则表达式  
/^\s\*|\s\*$/;  
//匹配HTML标记的正则表达式  
< (\S\*?)[^>]\*>.\*?|< .\*? />;  
//sql 语句  
^(select|drop|delete|create|update|insert).\*$  
//提取信息中的网络链接  
(h|H)(r|R)(e|E)(f|F) \*= \*('|")?(\w|\\|\/|\.)+('|"| \*|>)?  
//提取信息中的邮件地址  
\w+([-+.]\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*  
//提取信息中的图片链接  
(s|S)(r|R)(c|C) \*= \*('|")?(\w|\\|\/|\.)+('|"| \*|>)?  
//提取信息中的 IP 地址  
(\d+)\.(\d+)\.(\d+)\.(\d+)  
//取信息中的中国手机号码  
(86)\*0\*13\d{9}  
//提取信息中的中国邮政编码  
[1-9]{1}(\d+){5}  
//提取信息中的浮点数（即小数）  
(-?\d\*)\.?\d+  
//提取信息中的任何数字  
(-?\d\*)(\.\d+)?  
//电话区号  
^0\d{2,3}$  
//腾讯 QQ 号  
^[1-9]\*[1-9][0-9]\*$  
//帐号（字母开头，允许 5-16 字节，允许字母数字下划线）  
^[a-zA-Z][a-zA-Z0-9\_]{4,15}$  
//中文、英文、数字及下划线  
^[\u4e00-\u9fa5\_a-zA-Z0-9]+$  
10. 格式化数字、金额  
  
function number\_format(number, decimals, dec\_point, thousands\_sep) {  
    /\*  
    \* 参数说明：  
    \* number：要格式化的数字  
    \* decimals：保留几位小数  
    \* dec\_point：小数点符号  
    \* thousands\_sep：千分位符号  
    \* \*/  
    number = (number + '').replace(/[^0-9+-Ee.]/g, '');  
    var n = !isFinite(+number) ? 0 : +number,  
        prec = !isFinite(+decimals) ? 0 : Math.abs(decimals),  
        sep = (typeof thousands\_sep === 'undefined') ? ',' : thousands\_sep,  
        dec = (typeof dec\_point === 'undefined') ? '.' : dec\_point,  
        s = '',  
        toFixedFix = function (n, prec) {  
            var k = Math.pow(10, prec);  
            return '' + Math.ceil(n \* k) / k;  
        };  
  
    s = (prec ? toFixedFix(n, prec) : '' + Math.round(n)).split('.');  
    var re = /(-?\d+)(\d{3})/;  
    while (re.test(s[0])) {  
        s[0] = s[0].replace(re, "$1" + sep + "$2");  
    }  
  
    if ((s[1] || '').length < prec) {  
        s[1] = s[1] || '';  
        s[1] += new Array(prec - s[1].length + 1).join('0');  
    }  
    return s.join(dec);  
}  
var num = number\_format(1234567.089, 2, ".", ",");//1,234,567.09