

远方的远方~

花开堪折直须折，莫待无花空折枝。

[博客园](#) [首页](#) [博问](#) [闪存](#) [新随笔](#) [联系](#) [订阅](#) [管理](#)

100个直接可以拿来用的JavaScript实用功能代码片段

把平时网站上常用的一些实用功能代码片段通通收集起来，方便网友们学习使用，利用好的话可以加快网友们的开发速度，提高工作效率。

目录如下：

- 1、原生JavaScript实现字符串长度截取
- 2、原生JavaScript获取域名主机
- 3、原生JavaScript清除空格
- 4、原生JavaScript替换全部
- 5、原生JavaScript转义html标签
- 6、原生JavaScript还原html标签
- 7、原生JavaScript时间日期格式转换
- 8、原生JavaScript判断是否为数字类型
- 9、原生JavaScript设置cookie值
- 10、原生JavaScript获取cookie值
- 11、原生JavaScript加入收藏夹
- 12、原生JavaScript设为首页
- 13、原生JavaScript判断IE6
- 14、原生JavaScript加载样式文件
- 15、原生JavaScript返回脚本内容
- 16、原生JavaScript清除脚本内容
- 17、原生JavaScript动态加载脚本文件
- 18、原生JavaScript返回按ID检索的元素对象
- 19、原生JavaScript返回浏览器版本内容
- 20、原生JavaScript元素显示的通用方法
- 21、原生JavaScript中有insertBefore方法,可惜却没有insertAfter方法?用如下函数实现
- 22、原生JavaScript中兼容浏览器绑定元素事件
- 23、原生JavaScript光标停在文字的后面，文本框获得焦点时调用
- 24、原生JavaScript检验URL链接是否有效
- 25、原生JavaScript格式化CSS样式代码
- 26、原生JavaScript压缩CSS样式代码
- 27、原生JavaScript获取当前路径
- 28、原生JavaScriptIP转成整型
- 29、原生JavaScript整型解析为IP地址
- 30、原生JavaScript实现checkbox全选与全不选
- 31、原生JavaScript判断是否移动设备
- 32、原生JavaScript判断是否移动设备访问
- 33、原生JavaScript判断是否苹果移动设备访问
- 34、原生JavaScript判断是否安卓移动设备访问
- 35、原生JavaScript判断是否Touch屏幕
- 36、原生JavaScript判断是否在安卓上的谷歌浏览器
- 37、原生JavaScript判断是否打开视窗
- 38、原生JavaScript获取移动设备初始化大小
- 39、原生JavaScript获取移动设备最大化大小

- 40、原生JavaScript获取移动设备屏幕宽度
- 41、原生JavaScript完美判断是否为网址
- 42、原生JavaScript根据样式名称检索元素对象
- 43、原生JavaScript判断是否以某个字符串开头
- 44、原生JavaScript判断是否以某个字符串结束
- 45、原生JavaScript返回IE浏览器的版本号
- 46、原生JavaScript获取页面高度
- 47、原生JavaScript获取页面scrollLeft
- 48、原生JavaScript获取页面可视宽度
- 49、原生JavaScript获取页面宽度
- 50、原生JavaScript获取页面scrollTop
- 51、原生JavaScript获取页面可视高度
- 52、原生JavaScript跨浏览器添加事件
- 53、原生JavaScript跨浏览器删除事件
- 54、原生JavaScript去掉url前缀
- 55、原生JavaScript随机数时间戳
- 56、原生JavaScript全角半角转换,iCase: 0全到半, 1半到全, 其他不转化
- 57、原生JavaScript确认是否键盘有效输入值
- 58、原生JavaScript获取网页被卷去的位置
- 59、原生JavaScript另一种正则日期格式化函数+调用方法
- 60、原生JavaScript时间个性化输出功能
- 61、原生JavaScript解决offsetX兼容性问题
- 62、原生JavaScript常用的正则表达式
- 63、原生JavaScript实现返回顶部的通用方法
- 64、原生JavaScript获得URL中GET参数值
- 65、原生JavaScript实现全选通用方法
- 66、原生JavaScript实现全部取消选择通用方法
- 67、原生JavaScript实现打开一个窗体通用方法
- 68、原生JavaScript判断是否为客户端设备
- 69、原生JavaScript获取单选按钮的值
- 70、原生JavaScript获取复选框的值
- 71、原生JavaScript判断是否为邮箱
- 72、原生JavaScript判断是否有列表中的危险字符
- 73、原生JavaScript判断字符串是否大于规定的长度
- 74、原生JavaScript判断字符串是为网址不区分大小写
- 75、原生JavaScript判断字符串是否为小数
- 76、原生JavaScript判断字符串是否为整数
- 77、原生JavaScript判断字符串是否为浮点数
- 78、原生JavaScript判断字符是否为A-Za-z英文字母
- 79、原生JavaScript判断字符串是否邮政编码
- 80、原生JavaScript判断字符是否空NULL
- 81、原生JavaScript用正则表达式提取页面代码中所有网址
- 82、原生JavaScript用正则表达式清除相同的数组(低效率)
- 83、原生JavaScript用正则表达式清除相同的数组(高效率)
- 84、原生JavaScript用正则表达式按字母排序, 对每行进行数组排序
- 85、原生JavaScript字符串反序
- 86、原生JavaScript用正则表达式清除html代码中的脚本
- 87、原生JavaScript动态执行JavaScript脚本
- 88、原生JavaScript动态执行VBScript脚本
- 89、原生JavaScript实现金额大写转换函数
- 90、原生JavaScript常用的正则表达式大收集
- 91、原生JavaScript实现窗体改变事件resize的操作（兼容所有的浏览器）

- 92、原生JavaScript用正则清除空格分左右
- 93、原生JavaScript判断变量是否空值
- 94、原生JavaScript实现base64解码
- 95、原生JavaScript实现utf8解码
- 96、原生JavaScript获取窗体可见范围的宽与高
- 97、原生JavaScript判断IE版本号（既简洁、又向后兼容！）
- 98、原生JavaScript获取浏览器版本号
- 99、原生JavaScript半角转换为全角函数
- 100、原生JavaScript全角转换为半角函数

1、原生JavaScript实现字符串长度截取

```
1 function cutstr(str, len) {
2     var temp;
3     var icount = 0;
4     var patrn = /^[^x00-\xff]/;
5     var stre = "";
6     for (var i = 0; i < str.length; i++) {
7         if (icontains < len - 1) {
8             temp = str.substr(i, 1);
9             if (patrn.exec(temp) == null) {
10                 icount = icount + 1
11             } else {
12                 icount = icount + 2
13             }
14             stre += temp
15         } else {
16             break
17         }
18     }
19     return stre + "...";
20 }
```

2、原生JavaScript获取域名主机

```
1 function getHost(url) {
2     var host = "null";
3     if(typeof url == "undefined" || null == url) {
4         url = window.location.href;
5     }
6     var regex = /^\\w+\\.\\:\\/\\/([\\^\\/]*).*/;
7     var match = url.match(regex);
8     if(typeof match != "undefined" && null != match) {
9         host = match[1];
10    }
11    return host;
12 }
```

3、原生JavaScript清除空格

```
1 String.prototype.trim = function() {
2     var reExtraSpace = /^\\s*(.*)\\s$/;
3     return this.replace(reExtraSpace, "$1");
4 }
```

4、原生JavaScript替换全部

```
String.prototype.replaceAll = function(s1, s2) {
    return this.replace(new RegExp(s1, "gm"), s2)
}
```

5、原生JavaScript转义html标签

```
function HtmlEncode(text) {
    return text.replace(/&/g, '&amp;').replace(/\"/g, '&quot;').replace(/</g, '&lt;').replace(/>/g, '&gt;');
}
```

6、原生JavaScript还原html标签

```
function HtmlDecode(text) {
    return text.replace(/&amp;/g, '&').replace(/&quot;/g, '\"').replace(/&lt;/g, '<').replace(/&gt;/g, '>');
}
```

7、原生JavaScript时间日期格式转换

```
Date.prototype.Format = function(formatStr) {
    var str = formatStr;
    var Week = ['日', '一', '二', '三', '四', '五', '六'];
    str = str.replace(/yyyy|YYYY/, this.getFullYear());
    str = str.replace(/yy|YY/, (this.getYear() % 100) > 9 ? (this.getYear() % 100).toString() : '0' + (this.getYear() % 100));
    str = str.replace(/MM/, (this.getMonth() + 1) > 9 ? (this.getMonth() + 1).toString() : '0' + (this.getMonth() + 1));
    str = str.replace(/M/g, (this.getMonth() + 1));
    str = str.replace(/w|W/g, Week[this.getDay()]);
    str = str.replace(/dd|DD/, this.getDate() > 9 ? this.getDate().toString() : '0' + this.getDate());
    str = str.replace(/d|D/g, this.getDate());
    str = str.replace(/hh|HH/, this.getHours() > 9 ? this.getHours().toString() : '0' + this.getHours());
    str = str.replace(/h|H/g, this.getHours());
    str = str.replace(/mm/, this.getMinutes() > 9 ? this.getMinutes().toString() : '0' + this.getMinutes());
    str = str.replace(/m/g, this.getMinutes());
    str = str.replace(/ss|SS/, this.getSeconds() > 9 ? this.getSeconds().toString() : '0' + this.getSeconds());
    str = str.replace(/s|S/g, this.getSeconds());
    return str
}
```

8、原生JavaScript判断是否为数字类型

```
function isDigit(value) {
    var patrn = /^[0-9]*$/;
    if (patrn.exec(value) == null || value == "") {
        return false
    } else {
        return true
    }
}
```

9、原生JavaScript设置cookie值

```
function setCookie(name, value, Hours) {
```

```
2         var d = new Date();
3         var offset = 8;
4         var utc = d.getTime() + (d.getTimezoneOffset() * 60000);
5         var nd = utc + (3600000 * offset);
6         var exp = new Date(nd);
7         exp.setTime(exp.getTime() + Hours * 60 * 60 * 1000);
8         document.cookie = name + "=" + escape(value) + ";path=/;expires=" + exp.toGMTString() +
";domain=360doc.com;"
9     }
```

10、原生JavaScript获取cookie值

```
1 function getCookie(name) {
2     var arr = document.cookie.match(new RegExp("(^| )" + name + "=[^;]*)(;|$)");
3     if (arr != null) return unescape(arr[2]);
4     return null
5 }
```

11、原生JavaScript加入收藏夹

```
1 function AddFavorite(sURL, sTitle) {
2     try {
3         window.external.addFavorite(sURL, sTitle)
4     } catch(e) {
5         try {
6             window.sidebar.addPanel(sTitle, sURL, "")
7         } catch(e) {
8             alert("加入收藏失败，请使用Ctrl+D进行添加")
9         }
10    }
11 }
```

12、原生JavaScript设为首页

```
1 function setHomepage() {
2     if (document.all) {
3         document.body.style.behavior = 'url(#default#homepage)';
4         document.body.setHomePage('http://www.jq-school.com')
5     } else if (window.sidebar) {
6         if (window.netscape) {
7             try {
8                 netscape.security.PrivilegeManager.enablePrivilege("UniversalXPConnect")
9             } catch(e) {
10                 alert("该操作被浏览器拒绝，如果想启用该功能，请在地址栏内输入 about:config,然后将项 signed.applets.codebase_principal_support 值该为true")
11             }
12         }
13         var prefs = Components.classes['@mozilla.org/preferences-service;1'].getService(Components.interfaces.nsIPrefBranch);
14         prefs.setCharPref('browser.startup.homepage', 'http://www.jq-school.com')
15     }
16 }
```

13、原生JavaScript判断IE6

```
1 var ua = navigator.userAgent.toLowerCase();
```

```
2   var isIE6 = ua.indexOf("msie 6") > -1;
3   if (isIE6) {
4       try {
5           document.execCommand("BackgroundImageCache", false, true)
6       } catch(e) {}
7   }
```



14、原生JavaScript加载样式文件

```
1 function LoadStyle(url) {
2     try {
3         document.createStyleSheet(url)
4     } catch(e) {
5         var cssLink = document.createElement('link');
6         cssLink.rel = 'stylesheet';
7         cssLink.type = 'text/css';
8         cssLink.href = url;
9         var head = document.getElementsByTagName('head')[0];
10        head.appendChild(cssLink)
11    }
12 }
```



15、原生JavaScript返回脚本内容

```
1 function evalscript(s) {
2     if(s.indexOf('<script') == -1) return s;
3     var p = /<script[^>]*?>([^\x00]*?)<\/script>/ig;
4     var arr = [];
5     while(arr = p.exec(s)) {
6         var p1 = /<script[^>]*?src=\"([^\>]*?)\"([^\>]*?(reload=\"1\")?(?:charset=\"([\\w\\-]+?)\"))/?>
<\/script>/i;
7         var arr1 = [];
8         arr1 = p1.exec(arr[0]);
9         if(arr1) {
10            appendscript(arr1[1], '', arr1[2], arr1[3]);
11        } else {
12            p1 = /<script(.*)>([^\x00]+?)<\/script>/i;
13            arr1 = p1.exec(arr[0]);
14            appendscript('', arr1[2], arr1[1].indexOf('reload=') != -1);
15        }
16    }
17    return s;
18 }
```



16、原生JavaScript清除脚本内容

```
1 function stripscript(s) {
2     return s.replace(/<script.*?>.*?<\/script>/ig, '');
3 }
```

17、原生JavaScript动态加载脚本文件

```
1 function appendscript(src, text, reload, charset) {
2     var id = hash(src + text);
3     if(!reload && in_array(id, evalscripts)) return;
```



```
4     if(reload && $(id)) {
5         $(id).parentNode.removeChild($(id));
6     }
7
8     evalscripts.push(id);
9     var scriptNode = document.createElement("script");
10    scriptNode.type = "text/javascript";
11    scriptNode.id = id;
12    scriptNode.charset = charset ? charset : (BROWSER.firefox ? document.characterSet : document.charset);
13    try {
14        if(src) {
15            scriptNode.src = src;
16            scriptNode.onloadDone = false;
17            scriptNode.onload = function () {
18                scriptNode.onloadDone = true;
19                JSLOADED[src] = 1;
20            };
21            scriptNode.onreadystatechange = function () {
22                if((scriptNode.readyState == 'loaded' || scriptNode.readyState == 'complete') &&
!scriptNode.onloadDone) {
23                    scriptNode.onloadDone = true;
24                    JSLOADED[src] = 1;
25                }
26            };
27        } else if(text){
28            scriptNode.text = text;
29        }
30        document.getElementsByTagName('head')[0].appendChild(scriptNode);
31    } catch(e) {}
32 }
```

18、原生JavaScript返回按ID检索的元素对象

```
1 function $(id) {
2     return !id ? null : document.getElementById(id);
3 }
```

19、原生JavaScript返回浏览器版本内容

```
1 function browserVersion(types) {
2     var other = 1;
3     for(i in types) {
4         var v = types[i] ? types[i] : i;
5         if(USERAGENT.indexOf(v) != -1){
6             var re = new RegExp(v + '(\s|\\s) ([\\d\\.]+)', 'ig');
7             var matches = re.exec(USERAGENT);
8             var ver = matches != null ? matches[2] : 0;
9             other = ver != 0 && v != 'mozilla' ? 0 : other;
10        }else {
11            var ver = 0;
12        }
13        eval('BROWSER.' + i + '= ver');
14    }
15    BROWSER.other = other;
16 }
```

20、原生JavaScript元素显示的通用方法

```
1 function $(id) {
2     return !id ? null : document.getElementById(id);
3 }
```

```
4     function display(id) {
5         var obj = $(id);
6         if(obj.style.visibility) {
7             obj.style.visibility = obj.style.visibility == 'visible' ? 'hidden' : 'visible';
8         } else {
9             obj.style.display = obj.style.display == '' ? 'none' : '';
10        }
11    }
```



21、原生JavaScript中有insertBefore方法,可惜却没有insertAfter方法?用如下函数实现

```
1 function insertAfter(newChild,refChild){
2     var parElem=refChild.parentNode;
3     if(parElem.lastChild==refChild){
4         refChild.appendChild(newChild);
5     }else{
6         parElem.insertBefore(newChild,refChild.nextSibling);
7     }
8 }
```



22、原生JavaScript中兼容浏览器绑定元素事件

```
1 function addEventSamp(obj,evt,fn){
2     if (obj.addEventListener) {
3         obj.addEventListener(evt, fn, false);
4     }else if(obj.attachEvent){
5         obj.attachEvent('on'+evt,fn);
6     }
7 }
```



23、原生JavaScript光标停在文字的后面，文本框获得焦点时调用

```
1 function focusLast(){
2     var e = event.srcElement;
3     var r =e.createTextRange();
4     r.moveStart('character',e.value.length);
5     r.collapse(true);
6     r.select();
7 }
```



24、原生JavaScript检验URL链接是否有效

```
1 function getUrlState(URL) {
2     var xmlhttp = new ActiveXObject("microsoft.xmlhttp");
3     xmlhttp.Open("GET",URL, false);
4     try{
5         xmlhttp.Send();
6     }catch(e) {
7     }finally{
8         var result = xmlhttp.responseText;
9         if(result){
10             if(xmlhttp.Status==200){
```



```
11         return(true);
12     }else{
13         return(false);
14     }
15 }else{
16     return(false);
17 }
18 }
19 }
```

[View Code](#)

25、原生JavaScript格式化CSS样式代码

```
1 function formatCss(s){//格式化代码
2     s = s.replace(/\s*([\{\}\:\;\,])\s*/g, "$1");
3     s = s.replace(/;\s*/g, ";"); //清除连续分号
4     s = s.replace(/\\, [\s\\.\\#\\d]*{/g, "{");
5     s = s.replace(/([^\s])\{([^\s])/g, "$1 {\n\t$2");
6     s = s.replace(/([^\s])\}\{([^\s])/g, "$1\n}\n$2");
7     s = s.replace(/([^\s]);([^\s])/g, "$1;\n\t$2");
8     return s;
9 }
```

[View Code](#)

26、原生JavaScript压缩CSS样式代码

```
1 function yasuoCss (s) { //压缩代码
2     s = s.replace(/\/\*(.|\n)*?\*\/g, ""); //删除注释
3     s = s.replace(/\s*([\{\}\:\;\,])\s*/g, "$1");
4     s = s.replace(/\\, [\s\\.\\#\\d]*{/g, "{"); //容错处理
5     s = s.replace(/;\s*/g, ";"); //清除连续分号
6     s = s.match(/^s*(\S+(\s+\S+)*\s*$)/); //去掉首尾空白
7     return (s == null) ? "" : s[1];
8 }
```

[View Code](#)

27、原生JavaScript获取当前路径

```
1 var currentPageUrl = "";
2 if (typeof this.href === "undefined") {
3     currentPageUrl = document.location.toString().toLowerCase();
4 }
5 else {
6     currentPageUrl = this.href.toString().toLowerCase();
7 }
```

[View Code](#)

28、原生JavaScriptIP转成整型

```
1 function _ip2int(ip){
2     var num = 0;
3     ip = ip.split(".");
4     num = Number(ip[0]) * 256 * 256 * 256 + Number(ip[1]) * 256 * 256 + Number(ip[2]) * 256 + Number(ip[3]);
5     num = num >>> 0;
6     return num;
7 }
```

[View Code](#)

29、原生JavaScript整型解析为IP地址

```
1 function _int2iP(num) {
```

```
2     var str;
3     var tt = new Array();
4     tt[0] = (num >>> 24) >>> 0;
5     tt[1] = ((num << 8) >>> 24) >>> 0;
6     tt[2] = (num << 16) >>> 24;
7     tt[3] = (num << 24) >>> 24;
8     str = String(tt[0]) + "." + String(tt[1]) + "." + String(tt[2]) + "." + String(tt[3]);
9     return str;
10 }
```

[View Code](#)

30、原生JavaScript实现checkbox全选与全不选

```
1 function checkAll() {
2     var selectall = document.getElementById("selectall");
3     var allbox = document.getElementsByName("allbox");
4     if (selectall.checked) {
5         for (var i = 0; i < allbox.length; i++) {
6             allbox[i].checked = true;
7         }
8     } else {
9         for (var i = 0; i < allbox.length; i++) {
10            allbox[i].checked = false;
11        }
12    }
13 }
```

[View Code](#)

31、原生JavaScript判断是否移动设备

```
1 function isMobile(){
2     if (typeof this._isMobile === 'boolean'){
3         return this._isMobile;
4     }
5     var screenWidth = this.getScreenWidth();
6     var fixViewPortsExperiment = rendererModel.runningExperiments.FixViewport ||
rendererModel.runningExperiments.fixviewport;
7     var fixViewPortsExperimentRunning = fixViewPortsExperiment && (fixViewPortsExperiment.toLowerCase() ===
"new");
8     if(!fixViewPortsExperiment){
9         if(!this.isAppleMobileDevice()){
10            screenWidth = screenWidth/window.devicePixelRatio;
11        }
12    }
13     var isMobileScreenSize = screenWidth < 600;
14     var isMobileUserAgent = false;
15     this._isMobile = isMobileScreenSize && this.isTouchScreen();
16     return this._isMobile;
17 }
```

[View Code](#)

32、原生JavaScript判断是否移动设备访问

```
1 function isMobileUserAgent(){
2     return
(/iphone|ipod|android.*mobile|windows.*phone|blackberry.*mobile/i.test(window.navigator.userAgent.toLowerCase()));
3 }
```

[View Code](#)

33、原生JavaScript判断是否苹果移动设备访问

```
1 function isAppleMobileDevice(){
2     return (/iphone|ipod|ipad|Macintosh/i.test(navigator.userAgent.toLowerCase()));
3 }
```

[View Code](#)

34、原生JavaScript判断是否安卓移动设备访问

```
1 function isAndroidMobileDevice() {  
2     return (/android/i.test(navigator.userAgent.toLowerCase()));  
3 }
```

[View Code](#)

35、原生JavaScript判断是否Touch屏幕

```
1 function isTouchScreen() {  
2     return (('ontouchstart' in window) || window.DocumentTouch && document instanceof DocumentTouch);  
3 }
```

[View Code](#)

36、原生JavaScript判断是否在安卓上的谷歌浏览器

```
1 function isNewChromeOnAndroid() {  
2     if(this.isAndroidMobileDevice()) {  
3         var userAgent = navigator.userAgent.toLowerCase();  
4         if(/chrome/i.test(userAgent)) {  
5             var parts = userAgent.split('chrome/');  
6  
7             var fullVersionString = parts[1].split(" ")[0];  
8             var versionString = fullVersionString.split('.')[0];  
9             var version = parseInt(versionString);  
10  
11             if(version >= 27) {  
12                 return true;  
13             }  
14         }  
15     }  
16     return false;  
17 }
```

[View Code](#)

37、原生JavaScript判断是否打开视窗

```
1 function isViewportOpen() {  
2     return !!document.getElementById('wixMobileViewport');  
3 }
```

[View Code](#)

38、原生JavaScript获取移动设备初始化大小

```
1 function getInitZoom() {  
2     if(!this._initZoom) {  
3         var screenWidth = Math.min(screen.height, screen.width);  
4         if(this.isAndroidMobileDevice() && !this.isNewChromeOnAndroid()) {  
5             screenWidth = screenWidth/window.devicePixelRatio;  
6         }  
7         this._initZoom = screenWidth / document.body.offsetWidth;  
8     }  
9     return this._initZoom;  
10 }
```

[View Code](#)

39、原生JavaScript获取移动设备最大化大小

```
1 function getZoom(){
2     var screenWidth = (Math.abs(window.orientation) === 90) ? Math.max(screen.height, screen.width) :
Math.min(screen.height, screen.width);
3     if(this.isAndroidMobileDevice() && !this.isNewChromeOnAndroid()){
4         screenWidth = screenWidth/window.devicePixelRatio;
5     }
6     var FixViewPortsExperiment = renderModel.runningExperiments.FixViewport ||
renderModel.runningExperiments.fixviewport;
7     var FixViewPortsExperimentRunning = FixViewPortsExperiment && (FixViewPortsExperiment === "New" ||
FixViewPortsExperiment === "new");
8     if(FixViewPortsExperimentRunning){
9         return screenWidth / window.innerWidth;
10    }else{
11        return screenWidth / document.body.offsetWidth;
12    }
13 }
```

[View Code](#)

40、原生JavaScript获取移动设备屏幕宽度

```
1 function getScreenWidth(){
2     var smallerSide = Math.min(screen.width, screen.height);
3     var fixViewPortsExperiment = renderModel.runningExperiments.FixViewport ||
renderModel.runningExperiments.fixviewport;
4     var fixViewPortsExperimentRunning = fixViewPortsExperiment && (fixViewPortsExperiment.toLowerCase() ===
"new");
5     if(fixViewPortsExperiment){
6         if(this.isAndroidMobileDevice() && !this.isNewChromeOnAndroid()){
7             smallerSide = smallerSide/window.devicePixelRatio;
8         }
9     }
10    return smallerSide;
11 }
```

[View Code](#)

41、原生JavaScript完美判断是否为网址

```
1 function IsURL(strUrl) {
2     var regular = /^b(((https?|ftp):\/\/)?[-a-z0-9]+(\.[-a-z0-9]+)*\.
(?:com|edu|gov|int|mil|net|org|biz|info|name|museum|asia|coop|aero|[-a-z][a-z]|((25[0-5])|(2[0-4]\d)|(1\d\d\d)|([1-
9]\d)\d)\d)\b(\/[-a-z0-9_:@&?+=,.\!\/~%\$]*)?)$/i
3     if (regular.test(strUrl)) {
4         return true;
5     }
6     else {
7         return false;
8     }
9 }
```

[View Code](#)

42、原生JavaScript根据样式名称检索元素对象

```
1 function getElementsByClassName(name) {
2     var tags = document.getElementsByTagName('*') || document.all;
3     var els = [];
4     for (var i = 0; i < tags.length; i++) {
5         if (tags[i].className) {
6             var cs = tags[i].className.split(' ');
7             for (var j = 0; j < cs.length; j++) {
8                 if (name == cs[j]) {
9                     els.push(tags[i]);
10                    break
11                }
12            }
13        }
14    }
```

```
11         }  
12     }  
13 }  
14 }  
15     return els  
16 }
```

[View Code](#)

43、原生JavaScript判断是否以某个字符串开头

```
⊕  
1 String.prototype.startsWith = function (s) {  
2     return this.indexOf(s) == 0  
3 }
```

[View Code](#)

44、原生JavaScript判断是否以某个字符串结束

```
⊕  
1 String.prototype.endsWith = function (s) {  
2     var d = this.length - s.length;  
3     return (d >= 0 && this.lastIndexOf(s) == d)  
4 }
```

[View Code](#)

45、原生JavaScript返回IE浏览器的版本号

```
⊕  
1 function getIE(){  
2     if (window.ActiveXObject){  
3         var v = navigator.userAgent.match(/MSIE ([^;]+)/)[1];  
4         return parseFloat(v.substring(0, v.indexOf(".")))  
5     }  
6     return false  
7 }
```

[View Code](#)

46、原生JavaScript获取页面高度

```
⊕  
1 function getPageHeight(){  
2     var g = document, a = g.body, f = g.documentElement, d = g.compatMode == "BackCompat"  
3         ? a  
4         : g.documentElement;  
5     return Math.max(f.scrollHeight, a.scrollHeight, d.clientHeight);  
6 }
```

[View Code](#)

47、原生JavaScript获取页面scrollLeft

```
⊕  
1 function getPageScrollLeft(){  
2     var a = document;  
3     return a.documentElement.scrollLeft || a.body.scrollLeft;  
4 }
```

[View Code](#)

48、原生JavaScript获取页面可视宽度

```
⊕  
1 function getPageViewWidth(){
```

```
2     var d = document, a = d.compatMode == "BackCompat"
3         ? d.body
4         : d.documentElement;
5     return a.clientWidth;
6 }
```

[View Code](#)

49、原生JavaScript获取页面宽度

```
1 function getPageWidth(){
2     var g = document, a = g.body, f = g.documentElement, d = g.compatMode == "BackCompat"
3         ? a
4         : g.documentElement;
5     return Math.max(f.scrollWidth, a.scrollWidth, d.clientWidth);
6 }
```

[View Code](#)

50、原生JavaScript获取页面scrollTop

```
1 function getPageScrollTop(){
2     var a = document;
3     return a.documentElement.scrollTop || a.body.scrollTop;
4 }
```

[View Code](#)

51、原生JavaScript获取页面可视高度

```
1 function getPageViewHeight() {
2     var d = document, a = d.compatMode == "BackCompat"
3         ? d.body
4         : d.documentElement;
5     return a.clientHeight;
6 }
```

[View Code](#)

52、原生JavaScript跨浏览器添加事件

```
1 function addEvt(oTarget,sEvtType,fnHandle) {
2     if(!oTarget){return;}
3     if(oTarget.addEventListener) {
4         oTarget.addEventListener(sEvtType,fnHandle,false);
5     }else if(oTarget.attachEvent) {
6         oTarget.attachEvent("on" + sEvtType,fnHandle);
7     }else{
8         oTarget["on" + sEvtType] = fnHandle;
9     }
10 }
```

[View Code](#)

53、原生JavaScript跨浏览器删除事件

```
1 function delEvt(oTarget,sEvtType,fnHandle) {
2     if(!oTarget){return;}
3     if(oTarget.removeEventListener) {
4         oTarget.removeEventListener(sEvtType,fnHandle,false);
5     }else if(oTarget.detachEvent) {
```

```
6      oTarget.attachEvent("on" + sEvtType,fnHandle);
7  }else{
8      oTarget["on" + sEvtType] = fnHandle;
9  }
10 }
```

[View Code](#)

54、原生JavaScript去掉url前缀

```
1 function removeUrlPrefix(a){
2     a=a.replace(/:/g,"").replace(/./g,"").replace(/ /g,"/");
3     while(trim(a).toLowerCase().indexOf("http://")==0){
4         a=trim(a.replace(/http:\\\\/\\/i,""));
5     }
6     return a;
7 }
```

[View Code](#)

55、原生JavaScript随机数时间戳

```
1 function uniqueId(){
2     var a=Math.random,b=parseInt;
3     return Number(new Date()).toString()+b(10*a())+b(10*a())+b(10*a());
4 }
```

[View Code](#)

56、原生JavaScript全角半角转换,iCase: 0全到半, 1半到全, 其他不转化

```
1 function chgCase(sStr,iCase){
2     if(typeof sStr != "string" || sStr.length <= 0 || !(iCase === 0 || iCase == 1)){
3         return sStr;
4     }
5     var i,oRs=[],iCode;
6     if(iCase){/*半->全*/
7         for(i=0; i<sStr.length;i+=1){
8             iCode = sStr.charCodeAt(i);
9             if(iCode == 32){
10                 iCode = 12288;
11             }else if(iCode < 127){
12                 iCode += 65248;
13             }
14             oRs.push(String.fromCharCode(iCode));
15         }
16     }else{/*全->半*/
17         for(i=0; i<sStr.length;i+=1){
18             iCode = sStr.charCodeAt(i);
19             if(iCode == 12288){
20                 iCode = 32;
21             }else if(iCode > 65280 && iCode < 65375){
22                 iCode -= 65248;
23             }
24             oRs.push(String.fromCharCode(iCode));
25         }
26     }
27     return oRs.join("");
28 }
```

[View Code](#)

57、原生JavaScript确认是否键盘有效输入值

[+](#)

```
1 function checkKey(iKey){
2     if(iKey == 32 || iKey == 229){return true;}/*空格和异常*/
3     if(iKey>47 && iKey < 58){return true;}/*数字*/
4     if(iKey>64 && iKey < 91){return true;}/*字母*/
5     if(iKey>95 && iKey < 108){return true;}/*数字键盘1*/
6     if(iKey>108 && iKey < 112){return true;}/*数字键盘2*/
7     if(iKey>185 && iKey < 193){return true;}/*符号1*/
8     if(iKey>218 && iKey < 223){return true;}/*符号2*/
9     return false;
10 }
```

[View Code](#)

58、原生JavaScript获取网页被卷去的位置

```
1 function getScrollXY() {
2     return document.body.scrollTop ? {
3         x: document.body.scrollLeft,
4         y: document.body.scrollTop
5     } : {
6         x: document.documentElement.scrollLeft,
7         y: document.documentElement.scrollTop
8     }
9 }
```

[View Code](#)

59、原生JavaScript另一种正则日期格式化函数+调用方法

```
1 Date.prototype.format = function(format){ //author: meizz
2     var o = {
3         "M+" : this.getMonth()+1, //month
4         "d+" : this.getDate(),    //day
5         "h+" : this.getHours(),   //hour
6         "m+" : this.getMinutes(), //minute
7         "s+" : this.getSeconds(), //second
8         "q+" : Math.floor((this.getMonth()+3)/3), //quarter
9         "S"  : this.getMilliseconds() //millisecond
10    }
11    if(/(y+)/.test(format)) format=format.replace(RegExp.$1,
12        (this.getFullYear()+"").substr(4 - RegExp.$1.length));
13    for(var k in o)if(new RegExp("(+" + k + "+)").test(format))
14        format = format.replace(RegExp.$1,
15            RegExp.$1.length==1 ? o[k] :
16            ("00"+ o[k]).substr((""+ o[k]).length));
17    return format;
18 }
19 alert(new Date().format("yyyy-MM-dd hh:mm:ss"));
```

[View Code](#)

60、原生JavaScript时间个性化输出功能

```
1 /*
2 1、< 60s, 显示为“刚刚”
3 2、>= 1min && < 60 min, 显示与当前时间差“xx分钟前”
4 3、>= 60min && < 1day, 显示与当前时间差“今天 xx:xx”
5 4、>= 1day && < 1year, 显示日期“xx月xx日 xx:xx”
6 5、>= 1year, 显示具体日期“xxxx年xx月xx日 xx:xx”
7 */
8 function timeFormat(time){
9     var date = new Date(time)
10     , curDate = new Date()
11     , year = date.getFullYear()
12     , month = date.getMonth() + 1
13     , day = date.getDate()
```



```
14     , hour = date.getHours()
15     , minute = date.getMinutes()
16     , curYear = curDate.getFullYear()
17     , curHour = curDate.getHours()
18     , timeStr;
19
20 if(year < curYear){
21     timeStr = year +'年'+ month +'月'+ day +'日 ' + hour + ':' + minute;
22 }else{
23     var pastTime = curDate - date
24     , pastH = pastTime/3600000;
25
26     if(pastH > curHour){
27         timeStr = month +'月'+ day +'日 ' + hour + ':' + minute;
28     }else if(pastH >= 1){
29         timeStr = '今天 ' + hour + ':' + minute + '分';
30     }else{
31         var pastM = curDate.getMinutes() - minute;
32         if(pastM > 1){
33             timeStr = pastM +'分钟前';
34         }else{
35             timeStr = '刚刚';
36         }
37     }
38 }
39 return timeStr;
40 }
```

[View Code](#)

61、原生JavaScript解决offsetX兼容性问题

```
1 // 针对火狐不支持offsetX/Y
2 function getOffset(e){
3     var target = e.target, // 当前触发的目标对象
4         eventCoord,
5         pageCoord,
6         offsetCoord;
7
8     // 计算当前触发元素到文档的距离
9     pageCoord = getPageCoord(target);
10
11    // 计算光标到文档的距离
12    eventCoord = {
13        X : window.pageXOffset + e.clientX,
14        Y : window.pageYOffset + e.clientY
15    };
16
17    // 相减获取光标到第一个定位的父元素的坐标
18    offsetCoord = {
19        X : eventCoord.X - pageCoord.X,
20        Y : eventCoord.Y - pageCoord.Y
21    };
22    return offsetCoord;
23 }
24
25 function getPageCoord(element){
26     var coord = { X : 0, Y : 0 };
27     // 计算从当前触发元素到根节点为止,
28     // 各级 offsetParent 元素的 offsetLeft 或 offsetTop 值之和
29     while (element){
30         coord.X += element.offsetLeft;
31         coord.Y += element.offsetTop;
32         element = element.offsetParent;
33     }
34     return coord;
35 }
```

[View Code](#)

62、原生JavaScript常用的正则表达式

```
1 //正整数
2 /^[0-9]*[1-9][0-9]*$/;
3 //负整数
4 /^-[0-9]*[1-9][0-9]*$/;
5 //正浮点数
6 /^((([0-9]+\.[0-9]*[1-9][0-9]*)|([0-9]*[1-9][0-9]*\.[0-9]+)|([0-9]*[1-9][0-9]*)))/;
7 //负浮点数
8 /^(-((([0-9]+\.[0-9]*[1-9][0-9]*)|([0-9]*[1-9][0-9]*\.[0-9]+)|([0-9]*[1-9][0-9]*)))$/;
9 //浮点数
10 /^(-?\d+)(\.\d+)?$/;
11 //email地址
12 /^[\\w-]+(\\.\\w-+)*@[\\w-]+(\\.\\w-+)$/;
13 //url地址
14 /^[a-zA-z]+://(\\w+(-\\w+)*)(\\.\\w+(-\\w+)*))*(\\?\\S*)?$/;
15 //年/月/日 (年-月-日、年.月.日)
16 /^(19|20)\\d\\d[- /.](0[1-9]|1[012])[- /.](0[1-9]|1[2][0-9]|3[01])$/;
17 //匹配中文字符
18 /[\\u4e00-\\u9fa5]/;
19 //匹配帐号是否合法(字母开头,允许5-10字节,允许字母数字下划线)
20 /^[a-zA-Z][a-zA-Z0-9_]{4,9}$/;
21 //匹配空白行的正则表达式
22 /^\\n\\s*\\r/;
23 //匹配中国邮政编码
24 /[1-9]\\d{5}(?!\\d)/;
25 //匹配身份证
26 /^\\d{15}|\\d{18}/;
27 //匹配国内电话号码
28 /(\\d{3}-|\\d{4}-)?(\\d{8}|\\d{7})?/;
29 //匹配IP地址
30 /((2[0-4]|\\d|25[0-5]|([01]?\\d\\d?)\\.){3}(2[0-4]|\\d|25[0-5]|([01]?\\d\\d?)\\.)/;
31 //匹配首尾空白字符的正则表达式
32 /^\\s*|\\s*$/;
33 //匹配HTML标记的正则表达式
34 < (\\S*?) (^>)*>.*?|< .*? />;
```

[View Code](#)

63、原生JavaScript实现返回顶部的通用方法

```
1 function backTop(btnId) {
2     var btn = document.getElementById(btnId);
3     var d = document.documentElement;
4     var b = document.body;
5     window.onscroll = set;
6     btn.style.display = "none";
7     btn.onclick = function() {
8         btn.style.display = "none";
9         window.onscroll = null;
10        this.timer = setInterval(function() {
11            d.scrollTop -= Math.ceil((d.scrollTop + b.scrollTop) * 0.1);
12            b.scrollTop -= Math.ceil((d.scrollTop + b.scrollTop) * 0.1);
13            if ((d.scrollTop + b.scrollTop) == 0) clearInterval(btn.timer, window.onscroll = set);
14        },
15        10);
16    };
17    function set() {
18        btn.style.display = (d.scrollTop + b.scrollTop > 100) ? 'block': "none"
19    }
20 };
21 backTop('goTop');
```

[View Code](#)

64、原生JavaScript获得URL中GET参数值

```
1 // 用法: 如果地址是 test.htm?t1=1&t2=2&t3=3, 那么能取得: GET["t1"], GET["t2"], GET["t3"]
```

```
2 function get_get(){
3     querystr = window.location.href.split("?")
4     if(querystr[1]){
5         GETs = querystr[1].split("&")
6         GET =new Array()
7         for(i=0;i<GETs.length;i++){
8             tmp_arr = GETs[i].split("=")
9             key=tmp_arr[0]
10            GET[key] = tmp_arr[1]
11        }
12    }
13    return querystr[1];
14 }
```

[View Code](#)

65、原生JavaScript实现全选通用方法

```
1 function checkall(form, prefix, checkall) {
2     var checkall = checkall ? checkall : 'chkall';
3     for(var i = 0; i < form.elements.length; i++) {
4         var e = form.elements[i];
5         if(e.type=="checkbox"){
6             e.checked = form.elements[checkall].checked;
7         }
8     }
9 }
```

[View Code](#)

66、原生JavaScript实现全部取消选择通用方法

```
1 function uncheckAll(form) {
2     for (var i=0;i<form.elements.length;i++){
3         var e = form.elements[i];
4         if (e.name != 'chkall')
5             e.checked=!e.checked;
6     }
7 }
```

[View Code](#)

67、原生JavaScript实现打开一个窗体通用方法

```
1 function openWindow(url,windowName,width,height){
2     var x = parseInt(screen.width / 2.0) - (width / 2.0);
3     var y = parseInt(screen.height / 2.0) - (height / 2.0);
4     var isMSIE= (navigator.appName == "Microsoft Internet Explorer");
5     if (isMSIE) {
6         var p = "resizable=1,location=no,scrollbars=no,width=";
7         p = p+width;
8         p = p+",height=";
9         p = p+height;
10        p = p+",left=";
11        p = p+x;
12        p = p+",top=";
13        p = p+y;
14        retval = window.open(url, windowName, p);
15    } else {
16        var win = window.open(url, "ZyiisPopup", "top=" + y + ",left=" + x + ",scrollbars=" + scrollbars +
17        ",dialog=yes,modal=yes,width=" + width + ",height=" + height + ",resizable=no" );
18        eval("try { win.resizeTo(width, height); } catch(e) { }");
19        win.focus();
20    }
```

[View Code](#)

68、原生JavaScript判断是否为客户端设备

```
1 function client(o){
2     var b = navigator.userAgent.toLowerCase();
3     var t = false;
4     if (o == 'isOP'){
5         t = b.indexOf('opera') > -1;
6     }
7     if (o == 'isIE'){
8         t = b.indexOf('msie') > -1;
9     }
10    if (o == 'isFF'){
11        t = b.indexOf('firefox') > -1;
12    }
13    return t;
14 }
```

[View Code](#)**69、原生JavaScript获取单选按钮的值**

```
1 function get_radio_value(field){
2     if(field&&field.length){
3         for(var i=0;i<field.length;i++){
4             if(field[i].checked){
5                 return field[i].value;
6             }
7         }
8     }else {
9         return ;
10    }
11 }
```

[View Code](#)**70、原生JavaScript获取复选框的值**

```
1 function get_checkbox_value(field){
2     if(field&&field.length){
3         for(var i=0;i<field.length;i++){
4             if(field[i].checked && !field[i].disabled){
5                 return field[i].value;
6             }
7         }
8     }else {
9         return;
10    }
11 }
```

[View Code](#)**71、原生JavaScript判断是否为邮箱**

```
1 function isEmail(str){
2     var re=/^\\w+((-\\w+)|(\\.\\w+))*\\@[A-Za-z0-9]+((\\.|-)[A-Za-z0-9]+)*\\.[A-Za-z0-9]+$\\/;
3     if (re.test(str) != true) {
4         return false;
5     }else{
6         return true;
7     }
8 }
```

[View Code](#)


```
6         return true;
7     }
```

[View Code](#)

77、原生JavaScript判断字符串是否为浮点数

```
⊕
1 function isFloat( str ) {
2     for(i=0;i<str.length;i++) {
3         if ((str.charAt(i)<"0" || str.charAt(i)>"9") && str.charAt(i) != '.'){
4             return false;
5         }
6     }
7     return true;
8 }
```

[View Code](#)

78、原生JavaScript判断字符是否为A-Za-z英文字母

```
⊕
1 function isLetters( str ){
2     var re=/^[A-Za-z]+$/;
3     if (str.match(re) == null)
4         return false;
5     else
6         return true;
7 }
```

[View Code](#)

79、原生JavaScript判断字符串是否邮政编码

```
⊕
1 function isValidPost( chars ) {
2     var re=/^\d{6}$/;
3     if (chars.match(re) == null)
4         return false;
5     else
6         return true;
7 }
```

[View Code](#)

80、原生JavaScript判断字符是否空NULL

```
⊕
1 function isNULL( chars ) {
2     if (chars == null)
3         return true;
4     if (jsTrim(chars).length==0)
5         return true;
6     return false;
7 }
```

[View Code](#)

81、原生JavaScript用正则表达式提取页面代码中所有网址

```
⊕
1 var aa = document.documentElement.outerHTML.match(/(url\(|src=|href=) [\\"']* *([^\\"'\(\)\<\>\\[\ ]+)([\\"'\(\)] *)|
(http:\\\\|\\w\\-\\.]+[^\\"'\(\)\<\>\\[\ ]+)(ig).join("\\r\\n").replace(/^(src=|href=|url\(|) [\\"']* *([^\\"'\(\)] *)|
)*$/igm, "");
2 alert(aa)
```

[View Code](#)

82、原生JavaScript用正则表达式清除相同的数组(低效率)

```
1 Array.prototype.unique=function(){
2     return this.reverse().join(",").match(/([^\,]+)(?!.*\1)/ig).reverse();
3 }
View Code
```

83、原生JavaScript用正则表达式清除相同的数组(高效率)

```
1 String.prototype.unique=function(){
2     var x=this.split(/\r\n+/);
3     var y='';
4     for(var i=0;i<x.length;i++){
5         if(!new RegExp("^"+x[i].replace(/([^\w])/ig,"\\$1")+"$","igm").test(y)){
6             y+=x[i]+"\r\n"
7         }
8     }
9     return y
10 }
View Code
```

84、原生JavaScript用正则表达式按字母排序，对每行进行数组排序

```
1 function SetSort(){
2     var text=K1.value.split(/\r\n/).sort().join("\r\n");//顺序
3     var test=K1.value.split(/\r\n/).sort().reverse().join("\r\n");//反序
4     K1.value=K1.value!=text?text:test;
5 }
View Code
```

85、原生JavaScript字符串反序

```
1 function IsReverse(text){
2     return text.split('').reverse().join('');
3 }
View Code
```

86、原生JavaScript用正则表达式清除html代码中的脚本

```
1 function clear_script(){
2     K1.value=K1.value.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,"");
3 }
View Code
```

87、原生JavaScript动态执行JavaScript脚本

```
1 function javascript(){
2     try{
3         eval(K1.value);
4     }catch(e){
5         alert(e.message);
6     }
7 }
```

```
6     }  
7 }  
View Code
```

88、原生JavaScript动态执行VBScript脚本

```
+  
1 function vbscript(){  
2     try{  
3         var script=document.getElementById("K1").value;  
4         if(script.trim()=="")return;  
5         window.execScript('On Error Resume Next \n'+script+'\n If Err.Number<>0 Then \n MsgBox "请输入正确的VBScript  
脚本!",48,"脚本错误!" \n End If',"vbscript")  
6     }catch(e){  
7         alert(e.message);  
8     }  
9 }  
View Code
```

89、原生JavaScript实现金额大写转换函数

```
+  
1 function transform(tranvalue) {  
2     try {  
3         var i = 1;  
4         var dw2 = new Array("", "万", "亿"); //大单位  
5         var dw1 = new Array("拾", "佰", "仟"); //小单位  
6         var dw = new Array("零", "壹", "贰", "叁", "肆", "伍", "陆", "柒", "捌", "玖"); //整数部分用  
7         //以下是小写转换成大写显示在合计大写的文本框中  
8         //分离整数与小数  
9         var source = splits(tranvalue);  
10        var num = source[0];  
11        var dig = source[1];  
12        //转换整数部分  
13        var k1 = 0; //计小单位  
14        var k2 = 0; //计大单位  
15        var sum = 0;  
16        var str = "";  
17        var len = source[0].length; //整数的长度  
18        for (i = 1; i <= len; i++) {  
19            var n = source[0].charAt(len - i); //取得某个位数上的数字  
20            var bn = 0;  
21            if (len - i - 1 >= 0) {  
22                bn = source[0].charAt(len - i - 1); //取得某个位数前一位上的数字  
23            }  
24            sum = sum + Number(n);  
25            if (sum != 0) {  
26                str = dw[Number(n)].concat(str); //取得该数字对应的大写数字，并插入到str字符串的前面  
27                if (n == '0') sum = 0;  
28            }  
29            if (len - i - 1 >= 0) { //在数字范围内  
30                if (k1 != 3) { //加小单位  
31                    if (bn != 0) {  
32                        str = dw1[k1].concat(str);  
33                    }  
34                    k1++;  
35                } else { //不加小单位，加大单位  
36                    k1 = 0;  
37                    var temp = str.charAt(0);  
38                    if (temp == "万" || temp == "亿") //若大单位前没有数字则舍去大单位  
39                        str = str.substr(1, str.length - 1);  
40                    str = dw2[k2].concat(str);  
41                    sum = 0;  
42                }  
43            }  
44            if (k1 == 3) //小单位到千则大单位进一  
45            {  
46                k2++;  
47            }  
48        }  
49    }  
50 }
```



```

48     }
49     //转换小数部分
50     var strdig = "";
51     if (dig != "") {
52         var n = dig.charAt(0);
53         if (n != 0) {
54             strdig += dw[Number(n)] + "角"; //加数字
55         }
56         var n = dig.charAt(1);
57         if (n != 0) {
58             strdig += dw[Number(n)] + "分"; //加数字
59         }
60     }
61     str += "元" + strdig;
62 } catch(e) {
63     return "0元";
64 }
65 return str;
66 }
67 //拆分整数与小数
68 function splits(tranvalue) {
69     var value = new Array(' ', ' ');
70     temp = tranvalue.split(".");
71     for (var i = 0; i < temp.length; i++) {
72         value[i] = temp[i];
73     }
74     return value;
75 }

```

[View Code](#)

90、原生JavaScript常用的正则表达式大收集

```

1  匹配中文字符的正则表达式: [\u4e00-\u9fa5]
2  匹配双字节字符 (包括汉字在内): [^\x00-\xff]
3  匹配空行的正则表达式: \n[\s| ]*\r
4  匹配 HTML 标记的正则表达式: <(.*?)>.*</\1>|<(.*?) \/>
5  匹配首尾空格的正则表达式: (^s*)|(\s*$)
6  匹配 IP 地址的正则表达式: /(\d+)\.(\d+)\.(\d+)\.(\d+)/g
7  匹配 Email 地址的正则表达式: \w+([-+.]w+)*@w+([-.]w+)*\w+([-.]w+)*
8  匹配网址 URL 的正则表达式: http://([w-]+\.[w-]+/([w-.]/*%&=)*)?
9  sql 语句: ^(select|drop|delete|create|update|insert).*
10 非负整数: ^\d+$
11 正整数: ^[0-9]*[1-9][0-9]*$
12 非正整数: ^((-d+)|(0+))$
13 负整数: ^-[0-9]*[1-9][0-9]*$
14 整数: ^-?\d+$
15 非负浮点数: ^\d+(\.\d+)?$
16 正浮点数: ^((0-9)+\.\d+|[0-9]*[1-9][0-9]*\.\d+)|([0-9]*[1-9][0-9]*\.\d+)|([0-9]*[1-9][0-9]*))$
17 非正浮点数: ^((-d+.\d+)?|(0+.\d+)?))$
18 英文字符串: ^[A-Za-z]+$
19 英文大写串: ^[A-Z]+$
20 英文小写串: ^[a-z]+$
21 英文字符数字串: ^[A-Za-z0-9]+$
22 英数字加下划线串: ^\w+$
23 E-mail地址: ^[\w-]+(\.[\w-]+)*@[\w-]+(\.[\w-]+)+$
24 URL: ^[a-zA-Z]+://(\w+(-\w+)*)(\.(w+(-\w+)*))*(\?s*)?$ 或: ^http://[A-Za-z0-9]+\.[A-Za-z0-9]+[\\/=]?%&-&_~@[\'\\";:~!]*([<>\"'\"])*$
25 邮政编码: ^[1-9]\d{5}$
26 电话号码: ^((\d{2,3})|(\d{3}-))?(0\d{2,3})|0\d{2,3})?[1-9]\d{6,7}(\d{1,4})?$
27 手机号码: ^((\d{2,3})|(\d{3}-))13\d{9}$
28 双字节字符 (包括汉字在内): ^\x00-\xff
29 匹配首尾空格: (^s*)|(\s*$)
30 匹配 HTML 标记: <(.*?)>.*</\1>|<(.*?) \/>
31 匹配空行: \n[\s| ]*\r
32 提取信息中的网络链接: (h|H)(r|R)(e|E)(f|F) *=(\'|")?(\w|\\|\/|\.|)+(\'|\"| *|>)?
33 提取信息中的邮件地址: \w+([-+.]w+)*@w+([-.]w+)*\w+([-.]w+)*
34 提取信息中的图片链接: (s|S)(r|R)(c|C) *=(\'|")?(\w|\\|\/|\.|)+(\'|\"| *|>)?
35 提取信息中的 IP 地址: (\d+)\.(\d+)\.(\d+)\.(\d+)
36 提取信息中的中国手机号码: (86)*0*13\d{9}
37 提取信息中的中国固定电话号码: (\d{3,4})|\d{3,4}-|\s)?\d{8}
38 提取信息中的中国电话号码 (包括移动和固定电话): (\d{3,4})|\d{3,4}-|\s)?\d{7,14}

```

```
39 提取信息中的中国邮政编码: [1-9]{1}(\d+){5}
40 提取信息中的浮点数(即小数): (-?\d*)\.\d+
41 提取信息中的任何数字: (-?\d*)(\.\d+)?
42 IP: (\d+)\.(\d+)\.(\d+)\.(\d+)
43 电话区号: ^0\d{2,3}$
44 腾讯 QQ 号: ^[1-9]*[1-9][0-9]*$
45 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}$
46 中文、英文、数字及下划线: ^[\u4e00-\u9fa5_a-zA-Z0-9_]+$
```

[View Code](#)

91、原生JavaScript实现窗体改变事件resize的操作(兼容所有的浏览器)

```
1 (function(){
2     var fn = function(){
3         var w = document.documentElement ? document.documentElement.clientWidth : document.body.clientWidth
4         ,r = 1255
5         ,b = Element.prototype
6         ,classname = b.className;
7         if(w < r){
8             //当窗体的宽度小于1255的时候执行相应的操作
9         }else{
10            //当窗体的宽度大于1255的时候执行相应的操作
11        }
12    }
13    if(window.addEventListener){
14        window.addEventListener('resize', function(){ fn(); });
15    }else if(window.attachEvent){
16        window.attachEvent('onresize', function(){ fn(); });
17    }
18    fn();
19 })();
```

[View Code](#)

92、原生JavaScript用正则清除空格分左右

```
1 function ltrim(s){ return s.replace( /^\s*/, ""); }
2 function rtrim(s){ return s.replace( /\s*/, ""); }
3 function trim(s){ return ltrim(rtrim(s)); }
```

[View Code](#)

93、原生JavaScript判断变量是否空值

```
1 /**
2  * 判断变量是否空值
3  * undefined, null, '', false, 0, [], {} 均返回true, 否则返回false
4  */
5 function empty(v){
6     switch (typeof v){
7         case 'undefined': return true;
8         case 'string': : if(trim(v).length == 0) return true; break;
9         case 'boolean' : if(!v) return true; break;
10        case 'number' : if(0 === v) return true; break;
11        case 'object' :
12            if(null === v) return true;
13            if(undefined !== v.length && v.length==0) return true;
14            for(var k in v){return false;} return true;
15            break;
16        }
17    return false;
18 }
```

[View Code](#)

94、原生JavaScript实现base64解码

```
1 function base64_decode(data){
2     var b64 = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=";
3     var o1, o2, o3, h1, h2, h3, h4, bits, i = 0,ac = 0,dec = "",tmp_arr = [];
4     if (!data) { return data; }
5     data += '=';
6     do {
7         h1 = b64.indexOf(data.charAt(i++));
8         h2 = b64.indexOf(data.charAt(i++));
9         h3 = b64.indexOf(data.charAt(i++));
10        h4 = b64.indexOf(data.charAt(i++));
11        bits = h1 << 18 | h2 << 12 | h3 << 6 | h4;
12        o1 = bits >> 16 & 0xff;
13        o2 = bits >> 8 & 0xff;
14        o3 = bits & 0xff;
15        if (h3 == 64) {
16            tmp_arr[ac++] = String.fromCharCode(o1);
17        } else if (h4 == 64) {
18            tmp_arr[ac++] = String.fromCharCode(o1, o2);
19        } else {
20            tmp_arr[ac++] = String.fromCharCode(o1, o2, o3);
21        }
22    } while (i < data.length);
23    dec = tmp_arr.join('');
24    dec = utf8_decode(dec);
25    return dec;
26 }
```

[View Code](#)

95、原生JavaScript实现utf8解码

```
1 function utf8_decode(str_data){
2     var tmp_arr = [],i = 0,ac = 0,c1 = 0,c2 = 0,c3 = 0;str_data += '';
3     while (i < str_data.length) {
4         c1 = str_data.charCodeAt(i);
5         if (c1 < 128) {
6             tmp_arr[ac++] = String.fromCharCode(c1);
7             i++;
8         } else if (c1 > 191 && c1 < 224) {
9             c2 = str_data.charCodeAt(i + 1);
10            tmp_arr[ac++] = String.fromCharCode(((c1 & 31) << 6) | (c2 & 63));
11            i += 2;
12        } else {
13            c2 = str_data.charCodeAt(i + 1);
14            c3 = str_data.charCodeAt(i + 2);
15            tmp_arr[ac++] = String.fromCharCode(((c1 & 15) << 12) | ((c2 & 63) << 6) | (c3 & 63));
16            i += 3;
17        }
18    }
19    return tmp_arr.join('');
20 }
```

[View Code](#)

96、原生JavaScript获取窗体可见范围的宽与高

```
1 function getViewSize(){
2     var de=document.documentElement;
3     var db=document.body;
4     var viewW=de.clientWidth==0 ? db.clientWidth : de.clientWidth;
5     var viewH=de.clientHeight==0 ? db.clientHeight : de.clientHeight;
6     return Array(viewW ,viewH);
7 }
```

[View Code](#)

97、原生JavaScript判断IE版本号（既简洁、又向后兼容！）

```
1 var _IE = (function(){
```

```
2   var v = 3, div = document.createElement('div'), all = div.getElementsByTagName('i');
3   while (
4       div.innerHTML = '<!--[if gt IE ' + (++v) + ']><i></i><![endif]-->',
5       all[0]
6   );
7   return v > 4 ? v : false ;
8 }());
```

[View Code](#)

98、原生JavaScript获取浏览器版本号

```
1 function browserVersion(types) {
2     var other = 1;
3     for (i in types) {
4         var v = types[i] ? types[i] : i;
5         if (USERAGENT.indexOf(v) != -1) {
6             var re = new RegExp(v + '(\\/|\\s|)([\\d\\.]+)', 'ig');
7             var matches = re.exec(USERAGENT);
8             var ver = matches != null ? matches[2] : 0;
9             other = ver != 0 && v != 'mozilla' ? 0 : other;
10        } else {
11            var ver = 0;
12        }
13        eval('BROWSER.' + i + '= ver');
14    }
15    BROWSER.other = other;
16 }
```

[View Code](#)

99、原生JavaScript半角转换为全角函数

```
1 function ToDBC(str){
2     var result = '';
3     for(var i=0; i < str.length; i++){
4         code = str.charCodeAt(i);
5         if(code >= 33 && code <= 126){
6             result += String.fromCharCode(str.charCodeAt(i) + 65248);
7         }else if (code == 32){
8             result += String.fromCharCode(str.charCodeAt(i) + 12288 - 32);
9         }else{
10            result += str.charAt(i);
11        }
12    }
13    return result;
14 }
```

[View Code](#)

100、原生JavaScript全角转换为半角函数


```
1 function ToCDB(str){
2     var result = '';
3     for(var i=0; i < str.length; i++){
4         code = str.charCodeAt(i);
5         if(code >= 65281 && code <= 65374){
6             result += String.fromCharCode(str.charCodeAt(i) - 65248);
7         }else if (code == 12288){
8             result += String.fromCharCode(str.charCodeAt(i) - 12288 + 32);
9         }else{
10            result += str.charAt(i);
11        }
12    }
13    return result;
14 }
```

[View Code](#)

分类: [JavaScript](#)

绿色通道: [好文要顶](#) [关注我](#) [收藏该文](#) [与我联系](#)





[远方的远方~](#)
[关注 - 1](#)
[粉丝 - 31](#)
[+加关注](#)

94

0

(请您对文章做出评价)

« 上一篇: [JS兼容性问题（FF与IE）](#)
» 下一篇: [JS创建Ajax的XMLHttpRequest对象的通用方法](#)

posted @ 2013-09-11 13:12 远方的远方~ 阅读(4038) 评论(81) 编辑 收藏

Copyright ©2013 远方的远方~