1 判断字符串是否是这样组成的，第一个必须是字母，后面可以是字母、数字、下划线，总长度为5-20var reg = ^[a-zA-Z][a-zA-Z\_0-9]{4,19}$ ;reg test("a1a\_\_a1a\_\_a1a\_\_a1a\_\_");

1. var reg = /^[a-zA-Z][a-zA-Z\_0-9]{4,19}$/;
2. reg.test("a1a\_\_a1a\_\_a1a\_\_a1a\_\_");

复制代码

2. 截取字符串abcdefg的efg

1. var str = "abcdefg";
2. if (/efg/.test(str)) {
3. var efg = str.substr(str.indexOf("efg"), 3);
4. alert(efg);
5. }

复制代码

3. 判断一个字符串中出现次数最多的字符，统计这个次数

1. //将字符串的字符保存在一个hash table中，key是字符，value是这个字符出现的次数
2. var str = "abcdefgaddda";
3. var obj = {};
4. for (var i = 0, l = str.length; i < l; i++) {
5. var key = str[i];
6. if (!obj[key]) {
7. obj[key] = 1;
8. } else {
9. obj[key]++;
10. }
11. }
13. /\*遍历这个hash table，获取value最大的key和value\*/
14. var max = -1;
15. var max\_key = "";
16. var key;
17. for (key in obj) {
18. if (max < obj[key]) {
19. max = obj[key];
20. max\_key = key;
21. }
22. }
24. alert("max:"+max+" max\_key:"+max\_key);

复制代码

4. IE与FF脚本兼容性问题  
  
(1) window.event：  
表示当前的事件对象，IE有这个对象，FF没有，FF通过给事件处理函数传递事件对象  
  
(2) 获取事件源  
IE用srcElement获取事件源，而FF用target获取事件源  
  
(3) 添加，去除事件  
IE：element.attachEvent(“onclick”, function) element.detachEvent(“onclick”, function)  
FF：element.addEventListener(“click”, function, true) element.removeEventListener(“click”, function, true)  
  
(4) 获取标签的自定义属性  
IE：div1.value或div1[“value”]  
FF：可用div1.getAttribute(“value”)  
  
(5) document.getElementByName()和document.all[name]  
IE；document.getElementByName()和document.all[name]均不能获取div元素  
FF：可以  
  
(6) input.type的属性  
IE：input.type只读  
FF：input.type可读写  
  
(7) innerText textContent outerHTML  
IE：支持innerText, outerHTML  
FF：支持textContent  
  
(8) 是否可用id代替HTML元素  
IE：可以用id来代替HTML元素  
FF：不可以  
  
5. 规避javascript多人开发函数重名问题  
(1) 可以开发前规定命名规范，根据不同开发人员开发的功能在函数前加前缀  
(2) 将每个开发人员的函数封装到类中，调用的时候就调用类的函数，即使函数重名只要类名不重复就ok  
  
6. javascript面向对象中继承实现  
javascript面向对象中的继承实现一般都使用到了构造函数和Prototype原型链，简单的代码如下：

1. function Animal(name) {
2. this.name = name;
3. }
5. Animal.prototype.getName = function() {alert(this.name)}
6. function Dog() {};
7. Dog.prototype = new Animal("Buddy");
8. Dog.prototype.constructor = Dog;
9. var dog = new Dog();

复制代码

7. FF下面实现outerHTML  
FF不支持outerHTML，要实现outerHTML还需要特殊处理  
思路如下：  
  
在页面中添加一个新的元素A，克隆一份需要获取outerHTML的元素，将这个元素append到新的A中，然后获取A的innerHTML就可以了。

1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2. <html xmlns="http://www.w3.org/1999/xhtml">
3. <head>
4. <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
5. <title>获取outerHMTL</title>
6. <style>
7. div{ background:#0000FF;width:100px;height:100px;}
8. span{ background:#00FF00;width:100px;height:100px;}
9. p{ background:#FF0000;width:100px;height:100px;}
10. </style>
11. </head>
12. <body>
13. <div id="a"><span>SPAN</span>DIV</div>
14. <span>SPAN</span>
15. <p>P</p>
16. <script type="text/javascript">
17. function getOuterHTML(id){
18. var el = document.getElementById(id);
19. var newNode = document.createElement("div");
20. document.appendChild(newNode);
21. var clone = el.cloneNode(true);
22. newNode.appendChild(clone);
23. alert(newNode.innerHTML);
24. document.removeChild(newNode);
25. }
26. getOuterHTML("a");
27. </script>
28. </body>
29. </html>

复制代码

8. 编写一个方法 求一个字符串的字节长度  
假设：一个英文字符占用一个字节，一个中文字符占用两个字节

1. function GetBytes(str){
2. var len = str.length;
3. var bytes = len;
4. for(var i=0; i<len; i++){
5. if (str.charCodeAt(i) > 255) bytes++;
6. }
7. return bytes;
8. }
9. alert(GetBytes("你好,as"));

复制代码

9. 编写一个方法 去掉一个数组的重复元素

1. var arr = [1 ,1 ,2, 3, 3, 2, 1];
2. Array.prototype.unique = function(){
3. var ret = [];
4. var o = {};
5. var len = this.length;
6. for (var i=0; i<len; i++){
7. var v = this[i];
8. if (!o[v]){
9. o[v] = 1;
10. ret.push(v);
11. }
12. }
13. return ret;
14. };
15. alert(arr.unique());

复制代码

10. 写出3个使用this的典型应用  
（1）在html元素事件属性中使用，如:

1. <input type=”button” onclick=”showInfo(this);” value=”点击一下”/>

复制代码

（2）构造函数

1. function Animal(name, color) {
2. this.name = name;
3. this.color = color;
4. }

复制代码

（3）

1. <input type="button" id="text" value="点击一下" />
2. <script type="text/<a href="http://www.bairuiw.com/tag/javascript" class="st\_tag internal\_tag" rel="tag" title="Posts tagged with Javascript">javascript</a>">
3. var btn = document.getElementById("text");
4. btn.onclick = function() {
5. alert(this.value); //此处的this是按钮元素
6. }
7. </script>

复制代码

（4）CSS expression表达式中使用this关键字

1. <table width="100px" height="100px">
2. <tr>
3. <td>
4. <div style="width:expression(this.parentNode.width);">div element</div>
5. </td>
6. </tr>
7. </table>

复制代码

12. 如何显示/隐藏一个DOM元素？

1. el.style.display = "";
2. el.style.display = "none";

复制代码

el是要操作的DOM元素  
  
  
13. JavaScript中如何检测一个变量是一个String类型？请写出函数实现  
  
String类型有两种生成方式：  
(1)Var str = “hello world”;  
(2)Var str2 = new String(“hello world”);

1. function IsString(str){
2. return (typeof str == "string" || str.constructor == String);
3. }
4. var str = "";
5. alert(IsString(1));
6. alert(IsString(str));
7. alert(IsString(new String(str)));

复制代码

14. 网页中实现一个计算当年还剩多少时间的倒数计时程序，要求网页上实时动态显示“××年还剩××天××时××分××秒”

1. <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
2. <html>
3. <head>
4. <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5. <title>倒计时</title>
6. </head>
7. <body>
8. <input type="text" value="" id="input" size="1000"/>
9. <script type="text/javascript">
10. function counter() {
11. var date = new Date();
12. var year = date.getFullYear();
13. var date2 = new Date(year, 12, 31, 23, 59, 59);
14. var time = (date2 - date)/1000;
15. var day =Math.floor(time/(24\*60\*60))
16. var hour = Math.floor(time%(24\*60\*60)/(60\*60))
17. var minute = Math.floor(time%(24\*60\*60)%(60\*60)/60);
18. var second = Math.floor(time%(24\*60\*60)%(60\*60)%60);
19. var str = year + "年还剩"+day+"天"+hour+"时"+minute+"分"+second+"秒";
20. document.getElementById("input").value = str;
21. }
22. window.setInterval("counter()", 1000);
23. </script>
24. </body>
25. </html>

复制代码

15. 补充代码，鼠标单击Button1后将Button1移动到Button2的后面

1. <div> <input type=”button” id =”button1″ value=”1″ onclick=”???”>
2. <input type=”button” id =”button2″ value=”2″ /”> </div>

复制代码

1. <div>
2. <input type="button" id ="button1" value="1" onclick="moveBtn(this);">
3. <input type="button" id ="button2" value="2" />
4. </div>
5. <script type="text/javascript">
6. function moveBtn(obj) {
7. var clone = obj.cloneNode(true);
8. var parent = obj.parentNode;
9. parent.appendChild(clone);
10. parent.removeChild(obj);
11. }
12. </script>

复制代码

16. JavaScript有哪几种数据类型  
简单：Number，Boolean，String，Null，Undefined  
复合：Object，Array，Function  
  
17. 下面css标签在JavaScript中调用应如何拼写，border-left-color，-moz-viewport  
borderLeftColor  
mozViewport  
  
18. JavaScript中如何对一个对象进行深度clone

1. function cloneObject(o) {
2. if(!o || 'object' !== typeof o) {
3. return o;
4. }
5. var c = 'function' === typeof o.pop ? [] : {};
6. var p, v;
7. for(p in o) {
8. if(o.hasOwnProperty(p)) {
9. v = o[p];
10. if(v && 'object' === typeof v) {
11. c[p] = Ext.ux.clone(v);
12. }
13. else {
14. c[p] = v;
15. }
16. }
17. }
18. return c;
19. };

复制代码

19. 如何控制alert中的换行

1. \n alert(“p\np”);

复制代码

20. 请实现，鼠标点击页面中的任意标签，alert该标签的名称．（注意兼容性）

1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2. <html xmlns="http://www.w3.org/1999/xhtml">
3. <head>
4. <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
5. <title>鼠标点击页面中的任意标签，alert该标签的名称</title>
6. <style>
7. div{ background:#0000FF;width:100px;height:100px;}
8. span{ background:#00FF00;width:100px;height:100px;}
9. p{ background:#FF0000;width:100px;height:100px;}
10. </style>
11. <script type="text/javascript">
12. document.onclick = function(evt){
13. var e = window.event || evt;
14. var tag = e["target"] || e["srcElement"];
15. alert(tag.tagName);
16. };
17. </script>
18. </head>
19. <body>
20. <div id="div"><span>SPAN</span>DIV</div>
21. <span>SPAN</span>
22. <p>P</p>
23. </body>
24. </html>

复制代码

21. 请编写一个JavaScript函数 parseQueryString，它的用途是把URL参数解析为一个对象，如：  
var url = “<http://witmax.cn/index.php?key0=0&key1=1&key2=2>″;

1. function parseQueryString(url){
2. var params = {};
3. var arr = url.split("?");
4. if (arr.length <= 1)
5. return params;
6. arr = arr[1].split("&");
7. for(var i=0, l=arr.length; i<l; i++){
8. var a = arr[i].split("=");
9. params[a[0]] = a[1];
10. }
11. return params;
12. }
14. var url = "http://witmax.cn/index.php?key0=0&key1=1&key2=2";
15. var ps = parseQueryString(url);
16. alert(ps["key1"]);

复制代码

22. ajax是什么? ajax的交互模型? 同步和异步的区别? 如何解决跨域问题?  
Ajax是多种技术组合起来的一种浏览器和服务器交互技术，基本思想是允许一个互联网浏览器向一个远程页面/服务做异步的http调用，并且用收到的数据来更新一个当前web页面而不必刷新整个页面。该技术能够改进客户端的体验。包含的技术：  
XHTML：对应W3C的XHTML规范，目前是XHTML1.0。  
  
CSS：对应W3C的CSS规范，目前是CSS2.0  
DOM：这里的DOM主要是指HTML DOM，XML DOM包括在下面的XML中  
JavaScript：对应于ECMA的ECMAScript规范  
XML：对应W3C的XML DOM、XSLT、XPath等等规范  
XMLHttpRequest：对应WhatWG的Web Applications1.0规范（[http://whatwg.org/specs/web-apps/current-work/](http://whatwg.org/specs/web-apps/current-work/" \t "_blank)）  
  
同步：脚本会停留并等待服务器发送回复然后再继续  
异步：脚本允许页面继续其进程并处理可能的回复  
  
跨域问题简单的理解就是因为JS同源策略的限制，a.com域名下的JS无法操作b.com或c.a.com下的对象，具体场景如下：  
  
PS：（1）如果是端口或者协议造成的跨域问题前端是无能为力的  
  
(2) 在跨域问题上，域仅仅通过URL的首部来识别而不会尝试判断相同的IP地址对应的域或者两个域是否对应一个IP  
前端对于跨域的解决办法：  
(1) document.domain+iframe  
(2) 动态创建script标签  
  
23. 什么是闭包？下面这个ul，如何点击每一列的时候alert其index?

1. <ul id=”test”>
2. <li>这是第一条</li>
3. <li>这是第二条</li>
4. <li>这是第三条</li>
5. </ul>

复制代码

内部函数被定义它的函数的外部区域调用的时候就产生了闭包。

1. (function A() {
2. var index = 0;
3. var ul = document.getElementById("test");
4. var obj = {};
5. for (var i = 0, l = ul.childNodes.length; i < l; i++) {
6. if (ul.childNodes[i].nodeName.toLowerCase() == "li") {
7. var li = ul.childNodes[i];
8. li.onclick = function() {
9. index++;
10. alert(index);
11. }
12. }
13. }
14. })();

复制代码

24. 请给出异步加载js方案，不少于两种  
默认情况javascript是同步加载的，也就是javascript的加载时阻塞的，后面的元素要等待javascript加载完毕后才能进行再加载，对于一些意义不是很大的javascript，如果放在页头会导致加载很慢的话，是会严重影响用户体验的。  
  
异步加载方式：  
(1) defer，只支持IE  
(2) async：  
(3) 创建script，插入到DOM中，加载完毕后callBack，见代码：

1. function loadScript(url, callback){
2. var script = document.createElement("script")
3. script.type = "text/javascript";
4. if (script.readyState){ //IE
5. script.onreadystatechange = function(){
6. if (script.readyState == "loaded" ||
7. script.readyState == "complete"){
8. script.onreadystatechange = null;
9. callback();
10. }
11. };
12. } else { //Others: Firefox, Safari, Chrome, and Opera
13. script.onload = function(){
14. callback();
15. };
16. }
17. script.src = url;
18. document.body.appendChild(script);
19. }

复制代码

25. 请设计一套方案，用于确保页面中JS加载完全。

1. var n = document.createElement("script");
2. n.type = "text/javascript";
3. //以上省略部分代码
4. //ie支持script的readystatechange属性
5. if(ua.ie){
6. n.onreadystatechange = function(){
7. var rs = this.readyState;
8. if('loaded' === rs || 'complete'===rs){
9. n.onreadystatechange = null;
10. f(id,url); //回调函数
11. }
12. };
13. //省略部分代码
14. //safari 3.x supports the load event for script nodes(DOM2)
15. n.addEventListener('load',function(){
16. f(id,url);
17. });
18. //firefox and opera support onload(but not dom2 in ff) handlers for
19. //script nodes. opera, but no ff, support the onload event for link
20. //nodes.
21. }else{
22. n.onload = function(){
23. f(id,url);
24. };
25. }

复制代码

26. js中如何定义class,如何扩展prototype?  
Ele.className = “\*\*\*”; //\*\*\*在css中定义，形式如下：.\*\*\* {…}  
A.prototype.B = C;  
A是某个构造函数的名字  
B是这个构造函数的属性  
C是想要定义的属性的值  
  
27. 如何添加html元素的事件,有几种方法.  
（1） 为HTML元素的事件属性赋值  
（2） 在JS中使用ele.on\*\*\* = function() {…}  
（3） 使用DOM2的添加事件的方法 addEventListener或attachEvent  
  
28. documen.write和 innerHTML的区别  
document.write只能重绘整个页面  
innerHTML可以重绘页面的一部分  
  
29. 多浏览器检测通过什么？  
（1） navigator.userAgent  
（2） 不同浏览器的特性，如addEventListener  
  
30. js的基础对象有那些, window和document的常用的方法和属性列出来  
String,Number,Boolean  
  
Window:  
方法：setInterval,setTimeout,clearInterval,clearTimeout,alert,confirm,open  
属性：name,parent,screenLeft,screenTop,self,top,status  
  
Document  
方法：createElement,execCommand,getElementById,getElementsByName,getElementByTagName,write,writeln  
属性：cookie,doctype,domain,documentElement,readyState,URL,  
  
31. [前端开发](http://www.aseoe.com/qianduan/)的优化问题  
（1） 减少http请求次数：css spirit,data uri  
（2） JS，CSS源码压缩  
（3） [前端](http://www.aseoe.com/)模板 JS+数据，减少由于HTML标签导致的带宽浪费，[前端](http://www.aseoe.com/" \t "_blank)用变量保存AJAX请求结果，每次操作本地变量，不用请求，减少请求次数  
（4） 用innerHTML代替DOM操作，减少DOM操作次数，优化[javascript](http://www.aseoe.com/javascript/" \t "_blank)性能  
（5） 用setTimeout来避免页面失去响应  
（6） 用hash-table来优化查找  
（7） 当需要设置的样式很多时设置className而不是直接操作style  
（8） 少用全局变量  
（9） 缓存DOM节点查找的结果  
（10） 避免使用CSS Expression  
（11） 图片预载  
（12） 避免在页面的主体布局中使用table，table要等其中的内容完全下载之后才会显示出来，显示比div+css布局慢  
  
32. 如何控制网页在网络传输过程中的数据量  
启用GZIP压缩  
保持良好的编程习惯，避免重复的CSS，JavaScript代码，多余的HTML标签和属性  
  
33. Flash、Ajax各自的优缺点，在使用中如何取舍？  
Ajax的优势  
（1） 可搜索性  
（2） 开放性  
（3） 费用  
（4） 易用性  
（5） 易于开发  
  
Flash的优势  
（1） 多媒体处理  
（2） 兼容性  
（3） 矢量图形 比SVG，Canvas优势大很多  
（4） 客户端资源调度，比如麦克风，摄像头