答案：

1.C

(1)、“+”是左关联，表达式相当于("2"+3)+4  
(2)、根据ES5规范11,6,1节，如果某个操作数是字符串或者能够通过以下步骤转换为字符串的话，+将进行拼接操作。（ToPrimitive、DefaultValue。。。）

2.C

 js禁用

document.getElementByIdx\_x\_x("btnShow").disabled=true;（这样就把按钮禁用了）

 如果要重新开启按钮，则可以让disabled=false;即可

3.A

只要 协议 、 域名 、 端口 有任何一个 不同, 都被当作是 不同 的域。

4.D

最后一个只是对块级元素的背景做修饰，并不能改变大小

5.A

A是语法错误

B是创建一个数组对象

C是创建一个对象

D是一个创建正则对象；若为var obj=/ /； 即赋值被注释掉，及运行被结束；

6.A

第一个字符必须是一个字母、下划线（\_）或一个美元符号（$）；其他字符可以是字母、下划线、美元符号或数字。

7.C

XMLHttpRequest返回值

0 初始化状态。XMLHttpRequest 对象已创建或已被 abort() 方法重置

1 open()方法已调用，但是 send() 方法未调用。请求还没有被发送

2 send()方法已调用，HTTP 请求已发送到 Web 服务器。未接收到响应

3 所有响应头部都已经接收到。响应体开始接收但未完成

4 HTTP 响应已经完全接收

常用的就以上这几种

8.ABCD

$.ajax()函数依赖服务器提供的信息来处理返回的数据。如果服务器报告说返回的数据是XML，那么返回的结果就可以用普通的XML方法或者jQuery的选择器来遍历。如果见得到其他类型，比如HTML，则数据就以文本形式来对待。

通过dataType选项还可以指定其他不同数据处理方式。除了单纯的XML，还可以指定 html、json、jsonp、script或者text。

9.A

10.B

简答

1.使用闭包主要是为了设计私有的方法和变量。闭包的优点是可以避免全局变量的污染，缺点是闭包会常驻内存，会增大内存使用量，使用不当很容易造成内存泄露。

闭包有三个特性：a.函数嵌套函数 b.函数内部可以引用外部的参数和变量 c.参数和变量不会被垃圾回收机制回收

2.

absolute :生成绝对定位的元素， 相对于最近一级的 定位不是 static 的父元素来进行定位。

fixed （老IE不支持）生成绝对定位的元素，通常相对于浏览器窗口或 frame 进行定位。

relative 生成相对定位的元素，相对于其在普通流中的位置进行定位。

static 默认值。没有定位，元素出现在正常的流中

3.HTML5 不基于 SGML，因此不需要对DTD进行引用，但是需要doctype来规范浏览器的行为（让浏览器按照它们应该的方式来运行）；而HTML4.01基于SGML,所以需要对DTD进行引用，才能告知浏览器文档所使用的文档类型。

4.

为了准确无误地把数据送达目标处，TCP协议采用了三次握手策略。用TCP协议把数据包送出去后，TCP不会对传送 后的情况置之不理，它一定会向对方确认是否成功送达。握手过程中使用了TCP的标志：SYN和ACK。

发送端首先发送一个带SYN标志的数据包给对方。接收端收到后，回传一个带有SYN/ACK标志的数据包以示传达确认信息。最后，发送端再回传一个带ACK标志的数据包，代表“握手”结束

若在握手过程中某个阶段莫名中断，TCP协议会再次以相同的顺序发送相同的数据包。

5.

标记清除（mark and sweep）

这是JavaScript最常见的垃圾回收方式，当变量进入执行环境的时候，比如函数中声明一个变量，垃圾回收器将其标记为“进入环境”，当变量离开环境的时候（函数执行结束）将其标记为“离开环境”。

垃圾回收器会在运行的时候给存储在内存中的所有变量加上标记，然后去掉环境中的变量以及被环境中变量所引用的变量（闭包），在这些完成之后仍存在标记的就是要删除的变量了

引用计数(reference counting)

在低版本IE中经常会出现内存泄露，很多时候就是因为其采用引用计数方式进行垃圾回收。引用计数的策略是跟踪记录每个值被使用的次数，当声明了一个 变量并将一个引用类型赋值给该变量的时候这个值的引用次数就加1，如果该变量的值变成了另外一个，则这个值得引用次数减1，当这个值的引用次数变为0的时 候，说明没有变量在使用，这个值没法被访问了，因此可以将其占用的空间回收，这样垃圾回收器会在运行的时候清理掉引用次数为0的值占用的空间。

在IE中虽然JavaScript对象通过标记清除的方式进行垃圾回收，但BOM与DOM对象却是通过引用计数回收垃圾的，

也就是说只要涉及BOM及DOM就会出现循环引用问题。

代码题

1.

function cloneObject(o) {

  if(!o || 'object' !== typeof o) {

       return o;

   }

  var c = 'function' === typeof o.pop ? [] : {};

    var p, v;

    for(p in o) {

         if(o.hasOwnProperty(p)) {

           v = o[p];

           if(v && 'object' === typeof v) {

                                          c[p] = Ext.ux.clone(v);

             }else {

                  c[p] = v;

              }

      }

    }

    return c;

};

2.

function parseQueryString(url){

   var params = {};

   var arr = url.split("?");

   if (arr.length <= 1)return params;

   arr = arr[1].split("&");

   for(var i=0, l=arr.length; i<l; i++){

     var a = arr[i].split("=");

      params[a[0]] = a[1];

    }

   return params;

}

var url = "http://witmax.cn/index.php?key0=0&key1=1&key2=2";

var ps = parseQueryString(url);

alert(ps["key1"]);

3.

// url获取，变量声明

var arrURL = location.search.substr(1).split("&"), arrParms , key, val, getURL = {};

// 取值，存入对象

for(var i = 0; i < arrURL.length; i++) {

arrParms = arrURL[i].split('=');

key = arrParms[0];

val = arrParms[1];

getURL[key] = val;

};

console.log(getURL);

4.

function sum(){

var total=0; // 接受和

for(var i=0;i<arguments.length;i++){

var cur=Number(arguments[i]);

if(!isNaN(cur)){

total+=cur;

}

}

return total;

}

alert(sum(1,2,'3','4px'));