1. overflow-x  属于 CSS2 还是 CSS3

2.  请列举几种可以清除浮动的方法（至少两种）

3.  display:none  和  visibility:hidden  的区别是什么

4.  请缩写以下代码：  
.box {  
     background-position: 10px 20px;  
     background-repeat: no-repeat;  
     background-attachment: fixed;  
     background-color: red;  
     background-image: url(box.png);  
}

5.  如何让一段文本中的所有英文单词的首字母大写

6.  请解释一下什么是闭包

7.  call 和 apply 的区别是什么

8.  如何使用原生 Javascript 代码深度克隆一个对象（注意区分对象类型）

9.  jQuery 中  $('.class')  和  $('div.class')  在 IE 8 下哪个效率更高，请解释原因

10.   以下哪个不是 HTML5 的新标签：  
a.  <article>  
b.  <section>  
c.  <address>  
d.  <time>

11.  最近关注了哪些前端相关的技术/博客/论坛，简述心得和看法

1.css3

2.（1）添加额外(空)标签  
通过在浮动元素末尾添加一个空的标签例如 <div style=”clear:both”></div>  
优点：通俗易懂，容易掌握；  
缺点：增加无意义的标签，有违结构与表现的分离。

（2）父元素设置 overflow：hidden  
通过设置父元素overflow值为hidden；在IE6中还需要触发 hasLayout ，例如 zoom：1；  
优点：不存在结构和语义化问题，代码量极少；  
缺点：内容增多时候容易造成不会自动换行导致内容被隐藏掉，无法显示需要溢出的元素。

（3）父元素也设置浮动  
优点：不存在结构和语义化问题，代码量极少;  
缺点：使得与父元素相邻的元素的布局会受到影响，不可能一直浮动到body，不推荐使用。

（4）给父元素添加clearfix类  
由于IE6-7不支持:after，使用 zoom:1触发 hasLayout。  
优点：结构和语义化完全正确,代码量居中;  
缺点：复用方式不当会造成代码量增加。

3.相同点：display:none与visibility:hidden都可以用来隐藏某个元素；

不同点：display:none在隐藏元素的时候，将其占位空间也去掉；而visibility:hidden只是隐藏了内容而已，其占位空间仍然保留。

4.这个没什么好说的，只要是使用过css背景属性的人都知道缩写形式，另外需一提的是color属性可放在最前面也可以放在最后面。

.box {

background: red url(box.png) no-repeat 10px 20px fixed;

}

5.css的text-transform：none(默认值) | capitalize | uppercase | lowercase | full-width(css3新增)；

取值方面可查看手册，具体是什么就不多说了。

6.对于闭包，相信每个人都有自己的见解，并且网上一大推关于闭包的概念，我的理解是能够访问到其包含(外部)函数内部变量的函数。

主要有两个特性：

一个是可以读取其包含函数内部的变量；另一个就是让这些变量的值始终保持在内存中。

7.相同点：都可以在特定作用域(环境)下调用函数；

不同点：传递的参数形式不同，call方法的参数必须一个个列举出来，而apply方法的参数是以一个数组的形式进行传递，也支持arguments参数。

8.对于深度克隆，我的理解是：在克隆或扩展一个对象时，当该对象是一个包括子对象的对象，也会遍历该子对象的属性并进行复制拷贝。所以深度克隆适用于对象的属性也是对象的情况。

贴出js代码：

[复制代码](http://runfeel.iteye.com/admin/blogs/)

/\*\*

\* 深度扩展对象--适用于对象的属性也是对象的情况

\* @param {Object}

\* @return {Object}

\*/

var deepextend = function (destination, source) {

for (var property in source) {

var copy = source[property]; // 获取source属性值

if (destination === copy) {

continue;

}

// 如果copy是一个对象，则递归调用(并传入copy参数)，直到copy不是一个对象为止

if (typeof copy === 'object') {//$.isObj(copy)

destination[property] = arguments.callee(destination[property] || {}, copy); //递归调用

// 否则直接把copy赋值给destination对象的属性(此时与$.extend方法等价)

} else {

destination[property] = copy;

}

}

return destination;

};

[复制代码](http://runfeel.iteye.com/admin/blogs/)

例子测试：

[复制代码](http://runfeel.iteye.com/admin/blogs/)

var destination = {

name: "hcy",

age: 22,

info: {sex: "man", job: "student"}

};

var source = {

name: "hc",

age: 23,

info: {sex: "woman", job: "developer"}

};

console.log(deepextend(destination, source));

// 输出destination的全部属性

for (var i in destination) {

if (typeof destination[i] === "object") {

for (var j in destination[i]) {

console.log(destination[i] + ":\t" + destination[i][j]);

}

} else {

console.log(i + ":\t" + destination[i]);

}

}

10.address不是html5新增的标签