

1. 列举你工作中遇到的IE6 BUG，谈谈解决方案。

答案参考：<http://css.doyoe.com/> 中的css hack。

2. 你做的页面在哪些浏览器测试过？这些浏览器的内核分别是什么？经常遇到的浏览器的兼容性有哪些？怎么会出现？解决方法是什么？

参考答案：

IE 内核浏览器：360，傲游，搜狗，世界之窗，腾讯 TT

非 IE 内核浏览器：firefox opera safari chrome

1. 就是 ie6 双倍边距的问题，在使用了 float 的情况下，不管是向左还是向右都会出现双倍，最简单的解决方法就是用 display:inline; 加到 css 里面去。

2. 文字本身的大小不兼容。同样是 font-size:14px 的宋体文字，在不同浏览器下占的空间是不一样的，ie 下实际占高 16px，下留白 3px，ff 下实际占高 17px，上留白 1px，下留白 3px，opera 下就更不一样了。解决方案：给文字设定 line-height。确保所有文字都有默认的 line-height 值。这点很重要，在高度上我们不能容忍 1px 的差异。

3. ff 下容器高度限定，即容器定义了 height 之后，容器边框的外形就确定了，不会被内容撑大，而 ie 下是会被内容撑大，高度限定失效。所以不要轻易给容器定义 height。

4. 还讨论内容撑破容器问题，横向上的。如果 float 容器未定义宽度，ff 下内容会尽可能撑开容器宽度，ie 下则会优先考虑内容折行。故，内容可能撑破的浮动容器需要定义 width。

5. 浮动的清除，ff 下不清除浮动是不行的。

6. mirrormargin bug，当外层元素内有 float 元素时，外层元素如定义 margin-top:14px，将自动生成 margin-bottom:14px。padding 也会出现类似问题，都是 ie6 下的特产，该类 bug 出现的情况较为复杂，远不只这一种出现条件，还没系统整理。解决方案：外层元素设定 border 或 设定 float。

7. 吞吃现象，限于篇幅，我就不展开了。还是 ie6，上下两个 div，上面的 div 设置背景，却发现下面没有设置背景的 div 也有了背景，这就是吞吃现象。对应上面的背景吞吃现象，还有滚动下边框缺失的现象。解决方案：使用 zoom:1。这个 zoom 好像是专门为了解决 ie6 bug 而生的。

8. 注释也能产生 bug~~~“多出来的一只猪。”这是前人总结这个 bug 使用的文案，ie6 的这个 bug 下，大家会在页面看到猪字出现两遍，重复的内容量因注释的多少而变。解决方案：用 “<! - [if !IE]>picRotate start <![endif] - >” 方法写注释。

9. <li/>里加 float <div/>，这是一个典型的，棘手的兼容问题，希望引起大家重视，给 li 不同的属性

会有不同的解释效果，ff下的解释稍可理解，ie6下的解释会让你摸不着头脑，由于问题的复杂性，将另起一文专门讨论该问题。在《ul使用心得》一文里有相关成果，却没给出问题解决的过程。

10. img下的留白。解决方案：给img设定 display:block。

11. 失去 line-height。<div style="line-height:20px"><img />文字</div>，很遗憾，在ie6下单行文字 line-height 效果消失了...，原因是<img />这个 inline-block 元素和 inline 元素写在一起了。解决方案：让img和文字都 float起来。

12. 链接的 hover 状态。a:hover img{width:300px} 我们想让鼠标 hover 时，链接里包含的图片宽度变化，可惜在ie6下无效，ie7、ff下有效。

13. 非链接的 hover 状态。div:hover{} 这样的样式ie6是不认的，在ie7、ff下才有效果。

14. ie下 overflow:hidden 对其下的绝对层 position:absolute 或者相对层 position:relative 无效。解决方案：给 overflow:hidden 加 position:relative 或者 position:absolute。另，ie6支持 overflow-x 或者 overflow-y 的特性，ie7、ff不支持。

15. ie6下严重的bug，float元素如没定义宽度，内部如有div定义了 height 或 zoom:1，这个div就会占满一整行，即使你给了宽度。float元素如果作为布局用或复杂的容器，都要给个宽度的。

16. ie6下的bug，绝对定位的div下包含相对定位的div，如果给内层相对定位的div高度 height 具体值，内层相对层将具有 100% 的 width 值，外层绝对层将被撑大。解决方案给内层相对层 float 属性。

17. ie6下的bug，<head></head>内有<base target="\_blank" />的情况下，position:relative 层下的 float 层内文字无法选中。

18. 终于来了个ff的缺点。width:100%这个东西在ie里用很方便，会向上逐层搜索 width 值，忽视浮动层的影响，ff下搜索至浮动层结束，如此，只能给中间的所有浮动层加 width:100%才行，累啊。opera这点倒学乖了跟了ie。

### 3. 多浏览器检测通过什么

参考答案：

```
(1) navigator.userAgent
(2) 能力检测：根据不同浏览器的特性，如addEventListener，有些浏览器支持，有些则不支持，所以可以用下面的代码来判断：
if(elem.addEventListener){//支持addEventListener的浏览器
    //...
}else{//不支持addEventListener的浏览器
    //...
}
```

### 4. 前端开发的优化问题

参考答案：

- (1) 减少http请求次数：css sprite,data uri
- (2) JS，CSS源码压缩
- (3) 前端模板 JS+数据，减少由于HTML标签导致的带宽浪费，前端用变量保存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

### 5. 如何控制网页在网络传输过程中的数据量

参考答案：

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

6. Flash、Ajax各自的优缺点，在使用中如何取舍？

参考答案：

Ajax的优势

- (1) 可搜索型
- (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

7. 总结一下IE与FF脚本兼容性问题

参考答案：

(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.getElementById()和document.all[name]

IE；document.getElementById()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

这里只列出了常见的，还有不少，更多的介绍可以参看JavaScript在IE浏览器和Firefox浏览器中的差异总结

8. 如何在FF下面实现outerHTML

参考答案：

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。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
    <title>获取outerHTML</title>
```

```

<style>
  div {
    background: #0000FF;
    width: 100px;
    height: 100px;
  }
  span {
    background: #00FF00;
    width: 100px;
    height: 100px;
  }
  p {
    background: #FF0000;
    width: 100px;
    height: 100px;
  }
</style>
</head>
<body>
  <div id="a">
    <span>SPAN</span>DIV
  </div>
  <span>SPAN</span>
  <p>
    P
  </p>
  <script type="text/javascript">
    function getOuterHTML(id) {
      var el = document.getElementById(id);
      var newNode = document.createElement("div");
      document.appendChild(newNode);
      var clone = el.cloneNode(true);
      newNode.appendChild(clone);
      alert(newNode.innerHTML);
      document.removeChild(newNode);
    }
    getOuterHTML("a");
  </script>
</body>
</html>

```

9. 说说YSlow(可以详细一点)

参考答案：

这个插件可以分析网站的页面，并告诉你为了提高网站性能，如何基于某些规则而进行优化。  
更多关于YSlow的信息可以在网上搜索...

10. IE6的双倍边距BUG指的是什么？怎么解决？

参考答案：

在IE6中，一个居左（或居右）浮动的元素放置进一个容器盒(box)，并在浮动元素上使用了左边距(或右边距) 在ie6内便产生双倍边距。  
解决方法是：设置display：inline；

11. 如何让一个浮动的div水平居中？

参考答案：

```
div{width:200px;height:200px;position:relative;top:50%;Left:50%;margin-left:-100px;}
```