CSS 选择符有哪些？哪些属性可以继承？优先级算法如何计算？ CSS3新增伪类有那些？

```
    *   1.id选择器（ # myid）
        2.类选择器（.myclassname）
        3.标签选择器（div, h1, p）
        4.相邻选择器（h1 + p）
        5.子选择器（ul > li）
        6.后代选择器（li a）
        7.通配符选择器（ * ）
        8.属性选择器（a[rel = "external"]）
        9.伪类选择器（a: hover, li:nth-child）
     *   可继承的样式： font-size font-family color, text-indent;
     *   不可继承的样式：border padding margin width height ;
     *   优先级就近原则，同权重情况下样式定义最近者为准；
     *   载入样式以最后载入的定位为准；

优先级为：
    !important >  id > class > tag
    important 比 内联优先级高,但内联比 id 要高

CSS3新增伪类举例：
p:first-of-type 选择属于其父元素的首个 <p> 元素的每个 <p> 元素。
p:last-of-type  选择属于其父元素的最后 <p> 元素的每个 <p> 元素。
```

```
p:only-of-type  选择属于其父元素唯一的 <p> 元素的每个 <p> 元素。
p:only-child    选择属于其父元素的唯一子元素的每个 <p> 元素。
p:nth-child(2)  选择属于其父元素的第二个子元素的每个 <p> 元素。
:enabled  :disabled 控制表单控件的禁用状态。
:checked        单选框或复选框被选中。
```

CSS3有哪些新特性?

```
CSS3实现圆角（border-radius），阴影（box-shadow），
对文字加特效（text-shadow、），线性渐变（gradient），旋转（transform）
transform:rotate(9deg) scale(0.85,0.90) translate(0px,-30px) skew(-9deg,0deg);//旋转,缩放,定位,倾斜
增加了更多的CSS选择器  多背景 rgba
在CSS3中唯一引入的伪元素是::selection.
媒体查询，多栏布局
border-image
```

常见兼容性问题?

```
* png24位的图片在iE6浏览器上出现背景，解决方案是做成PNG8.也可以引用一段脚本处理.

* 浏览器默认的margin和padding不同。解决方案是加一个全局的*{margin:0;padding:0;}来统一。

* IE6双边距bug:块属性标签float后，又有横行的margin情况下，在ie6显示margin比设置的大。

* 浮动ie产生的双倍距离（IE6双边距问题：在IE6下，如果对元素设置了浮动，同时又设置了margin-left或margin-right，margin值会加倍。）
  #box{ float:left; width:10px; margin:0 0 0 100px;}

 这种情况之下IE会产生20px的距离，解决方案是在float的标签样式控制中加入 --_display:inline;将其转化为行内属性。(_这个符号只有ie6会识别)

*  渐进识别的方式，从总体中逐渐排除局部。

  首先，巧妙的使用“\9”这一标记，将IE游览器从所有情况中分离出来。
  接着，再次使用“+”将IE8和IE7、IE6分离开来，这样IE8已经独立识别。

  css
      .bb{
       background-color:#f1ee18;/*所有识别*/
      .background-color:#00deff\9; /*IE6、7、8识别*/
      +background-color:#a200ff;/*IE6、7识别*/
      _background-color:#1e0bd1;/*IE6识别*/
      }

*  IE下,可以使用获取常规属性的方法来获取自定义属性,
   也可以使用getAttribute()获取自定义属性;
   Firefox下,只能使用getAttribute()获取自定义属性.
   解决方法:统一通过getAttribute()获取自定义属性.
```

```
* IE下,event对象有x,y属性,但是没有pageX,pageY属性;
  Firefox下,event对象有pageX,pageY属性,但是没有x,y属性.

* 解决方法:（条件注释）缺点是在IE浏览器下可能会增加额外的HTTP请求数。

* Chrome 中文界面下默认会将小于 12px 的文本强制按照 12px 显示,
  可通过加入 CSS 属性 -webkit-text-size-adjust: none; 解决.

* 超链接访问过后hover样式就不出现了 被点击访问过的超链接样式不在具有hover和active了解决方法是改变CSS属性的排列顺序:
L-V-H-A :  a:link {} a:visited {} a:hover {} a:active {}

* 怪异模式问题:漏写DTD声明,Firefox仍然会按照标准模式来解析网页,但在IE中会触发怪异模式。为避免怪异模式给我们带来不必要的麻烦,最好养成书写DTD声明的好习惯。现在可以使用[html5](http://www.w3.org/TR/html5/single-page.html)推荐的写法:`<doctype html>`

* 上下margin重合问题
ie和ff都存在,相邻的两个div的margin-left和margin-right不会重合,但是margin-top和margin-bottom却会发生重合。
解决方法,养成良好的代码编写习惯,同时采用margin-top或者同时采用margin-bottom。
* ie6对png图片格式支持不好(引用一段脚本处理)
```

清除浮动

1. **父级div定义伪类：after和zoom** #
2. **在结尾处添加空div标签clear:both** #
3. **父级div定义height** #
4. **父级div定义overflow:hidden** #
5. **父级div定义overflow:auto** #
6. **父级div也一起浮动** #
7. **父级div定义display:table** #
8. **结尾处加br标签clear:both** #

DOM操作——怎样添加、移除、移动、复制、创建和查找节点。

```
（1）创建新节点
      createDocumentFragment()    //创建一个DOM片段
      createElement()   //创建一个具体的元素
      createTextNode()   //创建一个文本节点

（2）添加、移除、替换、插入
```

```
      appendChild()
      removeChild()
      replaceChild()
      insertBefore() //在已有的子节点前插入一个新的子节点

（3）查找
      getElementsByTagName()    //通过标签名称
      getElementsByName()    //通过元素的Name属性的值(IE容错能力较强，会得到一个数组，其中包括id
等于name值的)
      getElementById()    //通过元素Id，唯一性
```

html5有哪些新特性、移除了那些元素？如何处理HTML5新标签的浏览器兼容问题？如何区分 HTML 和 HTML5?

```
* HTML5 现在已经不是 SGML 的子集，主要是关于图像，位置，存储，多任务等功能的增加。

* 拖拽释放(Drag and drop) API
  语义化更好的内容标签（header,nav,footer,aside,article,section）
  音频、视频API(audio,video)
  画布(Canvas) API
  地理(Geolocation) API
  本地离线存储 localStorage 长期存储数据，浏览器关闭后数据不丢失；
  sessionStorage 的数据在浏览器关闭后自动删除

  表单控件，calendar、date、time、email、url、search
  新的技术webworker, websocket, Geolocation

* 移除的元素

纯表现的元素：basefont，big，center，font， s，strike，tt，u；

对可用性产生负面影响的元素：frame，frameset，noframes；

当然最好的方式是直接使用成熟的框架、使用最多的是==html5shim==框架
   <!--[if lt IE 9]>
   <script> src="http://html5shim.googlecode.com/svn/trunk/html5.js"</script>
   <![endif]-->
如何区分：  DOCTYPE声明\新增的结构元素\功能元素
```

如何进行网站性能优化

1. 从用户角度而言，优化能够让页面加载得更快、对用户的操作响应得更及时，能够给用户提供更为友好的体验。
2. 从服务商角度而言，优化能够减少页面请求数、或者减小请求所占带宽，能够节省可观的资源。总之，恰当的优化不仅能够改善站点的用户体验并且能够节省相当的资源利用。　　前端优化的途径有很多，按粒度大致可以分为两类，第一类是页面级别的优化，例如 HTTP请求数、脚本的无阻塞加载、内联脚本的位置优化等 ;第二类则是代码级别的优化，例如 Javascript中的DOM 操作优化、CSS选择符优化、图片优化以及 HTML结构优化等等。另外，本着提高投入产出比的目的，后

文提到的各种优化策略大致按照投入产出比从大到小的顺序排列。　　一、页面级优化

3. JavaScript 压缩和模块打包
4. 按需加载资源
5. 在使用 DOM 操作库时用上 array-ids
6. 缓存
7. 启用 HTTP/2
8. 应用性能分析
9. 使用负载均衡方案
10. 为了更快的启动时间考虑一下同构
11. 使用索引加速数据库查询
12. 使用更快的转译方案
13. 避免或最小化 JavaScript 和 CSS 的使用而阻塞渲染
14. 用于未来的一个建议：使用 service workers + 流
15. 图片编码优化

（1） 减少http请求次数：CSS Sprites， JS、CSS源码压缩、图片大小控制合适；网页Gzip，CDN托管，data缓存 ，图片服务器。
（2） 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
（3） 用innerHTML代替DOM操作，减少DOM操作次数，优化javascript性能。
（4） 当需要设置的样式很多时设置className而不是直接操作style。
（5） 少用全局变量、缓存DOM节点查找的结果。减少IO读取操作。
（6） 避免使用CSS Expression（css表达式)又称Dynamic properties(动态属性)。
（7） 图片预加载，将样式表放在顶部，将脚本放在底部　加上时间戳。

请说出三种减少页面加载时间的方法。

1.优化图片
2.图像格式的选择（GIF：提供的颜色较少，可用在一些对颜色要求不高的地方）
3.优化CSS（压缩合并css，如margin-top,margin-left...）
4.网址后加斜杠（如www.campr.com/目录，会判断这个“目录是什么文件类型，或者是目录。）
5.标明高度和宽度（如果浏览器没有找到这两个参数，它需要一边下载图片一边计算大小，如果图片很多，浏览器需要不断地调整页面。这不但影响速度，也影响浏览体验。
当浏览器知道了高度和宽度参数后，即使图片暂时无法显示，页面上也会腾出图片的空位，然后继续加载后面的内容。从而加载时间快了，浏览体验也更好了。）

6.减少http请求（合并文件，合并图片）。

你都使用哪些工具来测试代码的性能?

- Profiler
- JSPerf (http://jsperf.com/nexttick-vs-setzerotimeout-vs-settimeout)
- Dromaeo

列举IE与其他浏览器不一样的特性?

1）IE支持currentStyle，Firefox使用getComputStyle;

2）IE使用innerText，FireFox使用textContent；

3）滤镜方面：IE：filter:alpha(opacity= num)；Firefox：-moz-opacity:num；

4）事件方面：IE：attachEvent：火狐是addEventListener；

5）鼠标位置：IE是event.clientX；火狐是event.pageX；

6）IE使用event.srcElement；Firefox使用event.target；

7）IE中消除list的原点仅需margin:0即可达到最终效果；FIrefox需要设置margin:0;padding:0以及list-style:none；

javascript对象的几种创建方式

1，工厂模式
2，构造函数模式
3，原型模式
4，混合构造函数和原型模式
5，动态原型模式
6，寄生构造函数模式
7，稳妥构造函数模式

javascript继承的6种方法

1，原型链继承
2，借用构造函数继承
3，组合继承(原型+借用构造)
4，原型式继承
5，寄生式继承
6，寄生组合式继承

ajax过程

(1)创建XMLHttpRequest对象,也就是创建一个异步调用对象.

(2)创建一个新的HTTP请求,并指定该HTTP请求的方法、URL及验证信息.

(3)设置响应HTTP请求状态变化的函数.

(4)发送HTTP请求.

(5)获取异步调用返回的数据.

(6)使用JavaScript和DOM实现局部刷新.

异步加载和延迟加载

1.异步加载的方案： 动态插入script标签
2.通过ajax去获取js代码，然后通过eval执行
3.script标签上添加defer或者async属性
4.创建并插入iframe，让它异步执行js
5.延迟加载：有些 js 代码并不是页面初始化的时候就立刻需要的，而稍后的某些情况才需要的。

写一个通用的事件侦听器函数?

```
// event(事件)工具集，来源：github.com/markyun
    markyun.Event = {
        // 页面加载完成后
        readyEvent : function(fn) {
            if (fn==null) {
                fn=document;
            }
            var oldonload = window.onload;
            if (typeof window.onload != 'function') {
                window.onload = fn;
            } else {
                window.onload = function() {
                    oldonload();
                    fn();
                };
            }
        },
        // 视能力分别使用dom0||dom2||IE方式 来绑定事件
        // 参数： 操作的元素,事件名称 ,事件处理程序
        addEvent : function(element, type, handler) {
            if (element.addEventListener) {
                //事件类型、需要执行的函数、是否捕捉
                element.addEventListener(type, handler, false);
            } else if (element.attachEvent) {
                element.attachEvent('on' + type, function() {
```

```
                handler.call(element);
            });
        } else {
            element['on' + type] = handler;
        }
    },
    // 移除事件
    removeEvent : function(element, type, handler) {
        if (element.removeEnentListener) {
            element.removeEnentListener(type, handler, false);
        } else if (element.datachEvent) {
            element.detachEvent('on' + type, handler);
        } else {
            element['on' + type] = null;
        }
    },
    // 阻止事件 (主要是事件冒泡，因为IE不支持事件捕获)
    stopPropagation : function(ev) {
        if (ev.stopPropagation) {
            ev.stopPropagation();
        } else {
            ev.cancelBubble = true;
        }
    },
    // 取消事件的默认行为
    preventDefault : function(event) {
        if (event.preventDefault) {
            event.preventDefault();
        } else {
            event.returnValue = false;
        }
    },
    // 获取事件目标
    getTarget : function(event) {
        return event.target || event.srcElement;
    },
    // 获取event对象的引用，取到事件的所有信息，确保随时能使用event；
    getEvent : function(e) {
        var ev = e || window.event;
        if (!ev) {
            var c = this.getEvent.caller;
            while (c) {
                ev = c.arguments[0];
                if (ev && Event == ev.constructor) {
                    break;
                }
                c = c.caller;
            }
        }
        return ev;
    }
};
```

js对象的深度克隆

```
  function clone(Obj) {
        var buf;
        if (Obj instanceof Array) {
            buf = [];  //创建一个空的数组
            var i = Obj.length;
            while (i--) {
                buf[i] = clone(Obj[i]);
            }
            return buf;
        }else if (Obj instanceof Object){
            buf = {};  //创建一个空对象
            for (var k in Obj) {  //为这个对象添加新的属性
                buf[k] = clone(Obj[k]);
            }
            return buf;
        }else{
            return Obj;
        }
    }
```

js数组去重

以下是数组去重的三种方法：

```
Array.prototype.unique1 = function () {
  var n = []; //一个新的临时数组
  for (var i = 0; i < this.length; i++) //遍历当前数组
  {
    //如果当前数组的第i已经保存进了临时数组，那么跳过，
    //否则把当前项push到临时数组里面
    if (n.indexOf(this[i]) == -1) n.push(this[i]);
  }
  return n;
}

Array.prototype.unique2 = function()
{
    var n = {},r=[]; //n为hash表，r为临时数组
    for(var i = 0; i < this.length; i++) //遍历当前数组
    {
        if (!n[this[i]]) //如果hash表中没有当前项
        {
            n[this[i]] = true; //存入hash表
            r.push(this[i]); //把当前数组的当前项push到临时数组里面
        }
    }
```

```
    return r;
}

Array.prototype.unique3 = function()
{
    var n = [this[0]]; //结果数组
    for(var i = 1; i < this.length; i++) //从第二项开始遍历
    {
        //如果当前数组的第i项在当前数组中第一次出现的位置不是i,
        //那么表示第i项是重复的，忽略掉。否则存入结果数组
        if (this.indexOf(this[i]) == i) n.push(this[i]);
    }
    return n;
}
```

cache-control

网页的缓存是由HTTP消息头中的`private、no-cache、max-age、must-revalidate`等，默认为`max-age的效果。但是如果同时存在，则被max-age覆盖。`

```
Expires = "Expires" ":" HTTP-date
```

例如

```
Expires: Thu, 01 Dec 1994 16:00:00 GMT （必须是GMT格式）
```

如果把它设置为`max-age都可以用来指定文档的过期时间，但是二者有一些细微差别`

```
1.Expires在HTTP/1.0中已经定义，Cache-Control:max-age在HTTP/1.1中才有定义，为了向下兼容，仅使用max-age不够；
2.Expires指定一个绝对的过期时间(GMT格式),这么做会导致至少2个问题：1)客户端和服务器时间不同步导致Expires的配置出现问题。 2）很容易在配置后忘记具体的过期时间，导致过期来临出现浪涌现象；

3.max-age 指定的是从文档被访问后的存活时间，这个时间是个相对值(比如:3600s),相对的是文档第一次被请求时服务器记录的Request_time(请求时间)

4.Expires指定的时间可以是相对文件的最后访问时间(Atime)或者修改时间(MTime),而max-age相对对的是文档的请求时间(Atime)

如果值为no-cache,那么每次都会访问服务器。如果值为max-age,则在过期之前不会重复访问服务器。
```

js操作获取和设置cookie

```
//创建cookie
function setCookie(name, value, expires, path, domain, secure) {
    var cookieText = encodeURIComponent(name) + '=' + encodeURIComponent(value);
    if (expires instanceof Date) {
```

```
        cookieText += '; expires=' + expires;
    }
    if (path) {
        cookieText += '; expires=' + expires;
    }
    if (domain) {
        cookieText += '; domain=' + domain;
    }
    if (secure) {
        cookieText += '; secure';
    }
    document.cookie = cookieText;
}

//获取cookie
function getCookie(name) {
    var cookieName = encodeURIComponent(name) + '=';
    var cookieStart = document.cookie.indexOf(cookieName);
    var cookieValue = null;
    if (cookieStart > -1) {
        var cookieEnd = document.cookie.indexOf(';', cookieStart);
        if (cookieEnd == -1) {
            cookieEnd = document.cookie.length;
        }
```