# [前端面试总结](http://www.cnblogs.com/dunken/p/4475468.html)

## 前端开发面试知识点大纲

### HTML&CSS

对Web标准的理解、浏览器内核差异、兼容性、hack、CSS基本功：布局、盒子模型、选择器优先级及使用、HTML5、CSS3、移动端适应

### JavaScript

数据类型、面向对象、继承、闭包、插件、作用域、跨域、原型链、模块化、自定义事件、内存泄漏、事件机制、异步装载回调、模板引擎、Nodejs、JSON、ajax等。

### 其他

HTTP、安全、正则、优化、重构、响应式、移动端、团队协作、可维护、SEO、UED、架构、职业生涯

## 必备知识点

作为一名前端工程师，无论工作年头长短都应该必须掌握的知识点：此条由 王子墨 发表在 [前端随笔](http://julying.com/blog/front-end-engineers-must-master-knowledge/" \t "http://www.cnblogs.com/dunken/p/_blank)

1. DOM结构 —— 两个节点之间可能存在哪些关系以及如何在节点之间任意移动。
2. DOM操作 ——如何添加、移除、移动、复制、创建和查找节点等。
3. 事件 —— 如何使用事件，以及IE和标准DOM事件模型之间存在的差别。
4. XMLHttpRequest —— 这是什么、怎样完整地执行一次GET请求、怎样检测错误。
5. 严格模式与混杂模式 —— 如何触发这两种模式，区分它们有何意义。
6. 盒模型 —— 外边距、内边距和边框之间的关系，及IE8以下版本的浏览器中的盒模型
7. 块级元素与行内元素 —— 怎么用CSS控制它们、以及如何合理的使用它们
8. 浮动元素——怎么使用它们、它们有什么问题以及怎么解决这些问题。
9. HTML与XHTML——二者有什么区别，你觉得应该使用哪一个并说出理由。
10. JSON —— 作用、用途、设计结构。

备注：

根据自己需要选择性阅读，面试题是对理论知识的总结，让自己学会应该如何表达。资料答案不够正确和全面，欢迎补充答案、题目；最好是现在网上没有的。格式不断修改更新中。

## HTML

Doctype作用? 严格模式与混杂模式如何区分？它们有何意义?

* <!DOCTYPE> 声明位于文档中的最前面，处于 <html> 标签之前。告知浏览器的解析器， 用什么文档类型 规范来解析这个文档
* 严格模式的排版和 JS 运作模式是 以该浏览器支持的最高标准运行。
* 在混杂模式中，页面以宽松的向后兼容的方式显示。模拟老式浏览器的行为以防止站点无法工作。 （4）、DOCTYPE不存在或格式不正确会导致文档以混杂模式呈现。

行内元素有哪些？块级元素有哪些？ 空(void)元素有那些？

* CSS规范规定，每个元素都有display属性，确定该元素的类型，每个元素都有默认的display值， 比如div默认display属性值为“block”，成为“块级”元素； span默认display属性值为“inline”，是“行内”元素。
* 行内元素有：a b span img input select strong（强调的语气） 块级元素有：div ul ol li dl dt dd h1 h2 h3 h4…p
* 知名的空元素： <br> <hr> <img> <input> <link> <meta>
* 鲜为人知的是： <area> <base> <col> <command> <embed> <keygen> <param> <source> <track> <wbr>

link 和@import 的区别是？

* link属于XHTML标签，而@import是CSS提供的;
* 页面被加载的时，link会同时被加载，而@import引用的CSS会等到页面被加载完再加载;
* import只在IE5以上才能识别，而link是XHTML标签，无兼容问题;
* link方式的样式的权重 高于@import的权重.

浏览器的内核分别是什么?

\* IE浏览器的内核Trident、Mozilla的Gecko、Chrome的Blink（WebKit的分支）、Opera内核原为Presto，现为Blink；

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.

\* 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 #box{ float:left; width:10px; margin:0 0 0 100px;}

css

\* 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,even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 {}

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。

\* 绘画 canvas

用于媒介回放的 video 和 audio 元素

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

语意化更好的内容元素，比如 article、footer、header、nav、section

表单控件，calendar、date、time、email、url、search

新的技术webworker, websockt, 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；

支持HTML5新标签：

\* IE8/IE7/IE6支持通过document.createElement方法产生的标签，

可以利用这一特性让这些浏览器支持HTML5新标签，

浏览器支持新标签后，还需要添加标签默认的样式：

\* 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  
  
如何区分： DOCTYPE声明\新增的结构元素\功能元素

语义化的理解？

* 用正确的标签做正确的事情！
* html语义化就是让页面的内容结构化，便于对浏览器、搜索引擎解析；
* 在没有样式CCS情况下也以一种文档格式显示，并且是容易阅读的。
* 搜索引擎的爬虫依赖于标记来确定上下文和各个关键字的权重，利于 SEO。
* 使阅读源代码的人对网站更容易将网站分块，便于阅读维护理解。

HTML5的离线储存？

* localStorage 长期存储数据，浏览器关闭后数据不丢失；
* sessionStorage 数据在浏览器关闭后自动删除。

(写)描述一段语义的html代码

（HTML5中新增加的很多标签（如：<article>、<nav>、<header>和<footer>等）  
就是基于语义化设计原则）

|  |  |
| --- | --- |
| 1  2  3  4 | < div id="header">      < h1>标题< /h1>      < h2>专注Web前端技术< /h2>  < /div> |

iframe有那些缺点？

* \*iframe会阻塞主页面的Onload事件；
* \*iframe和主页面共享连接池，而浏览器对相同域的连接有限制，所以会影响页面的并行加载。 使用iframe之前需要考虑这两个缺点。如果需要使用iframe，最好是通过javascript 动态给iframe添加src属性值，这样可以可以绕开以上两个问题。

请描述一下 cookies，sessionStorage 和 localStorage 的区别？

* cookie在浏览器和服务器间来回传递。
* sessionStorage和localStorage不会 sessionStorage和localStorage的存储空间更大；
* sessionStorage和localStorage有更多丰富易用的接口；
* sessionStorage和localStorage各自独立的存储空间；

如何实现浏览器内多个标签页之间的通信? (阿里)

调用localstorge、cookies等本地存储方式

webSocket如何兼容低浏览器？(阿里)

Adobe Flash Socket 、 ActiveX HTMLFile (IE) 、 基于 multipart 编码发送 XHR 、 基于长轮询的 XHR

## CSS

介绍一下CSS的盒子模型？

（1）有两种， IE 盒子模型、标准 W3C 盒子模型；IE的content部分包含了 border 和 pading;

（2）盒模型： 内容(content)、填充(padding)、边界(margin)、 边框(border).

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, UL LI DL DD DT;

\* 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

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。

如何居中div？

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | // 给div设置一个宽度，然后添加margin:0 auto属性  div{      width:200px;      margin:0 auto;  } |

如何居中一个浮动元素？

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | 确定容器的宽高 宽500 高 300 的层  设置层的外边距    .div {    Width:500px ; height:300px;//高度可以不设    Margin: -150px 0 0 -250px;    position:relative;相对定位    background-color:pink;//方便看效果    left:50%;    top:50%;  } |

列出display的值，说明他们的作用。

* block 象块类型元素一样显示。
* none 缺省值。象行内元素类型一样显示。
* inline-block 象行内元素一样显示，但其内容象块类型元素一样显示。
* list-item 象块类型元素一样显示，并添加样式列表标记。

position的值， relative和absolute定位原点是？

* \*absolute：生成绝对定位的元素，相对于 static 定位以外的第一个父元素进行定位。
* \*fixed：生成绝对定位的元素，相对于浏览器窗口进行定位。
* \*relative：生成相对定位的元素，相对于其正常位置进行定位。
* \* static 默认值。没有定位，元素出现在正常的流中

\*（忽略 top, bottom, left, right z-index 声明）。

\* inherit 规定从父元素继承 position 属性的值。

CSS3有哪些新特性？

CSS3实现圆角（border-radius:8px），阴影（box-shadow:10px），

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

一个满屏 品 字布局 如何设计?

经常遇到的CSS的兼容性有哪些？原因，解决方法是什么？

为什么要初始化CSS样式?

– 因为浏览器的兼容问题，不同浏览器对有些标签的默认值是不同的，如果没对CSS初始化往往会出现浏览器之间的页面显示差异。

– 当然，初始化样式会对SEO有一定的影响，但鱼和熊掌不可兼得，但力求影响最小的情况下初始化。

\*最简单的初始化方法就是： \* {padding: 0; margin: 0;} （不建议）

淘宝的样式初始化：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | body, h1, h2, h3, h4, h5, h6, hr, p, blockquote, dl, dt, dd, ul, ol, li, pre, form, fieldset, legend, button, input, textarea, th, td { margin:0; padding:0; }  body, button, input, select, textarea { font:12px/1.5tahoma, arial, \5b8b\4f53; }  h1, h2, h3, h4, h5, h6{ font-size:100%; }  address, cite, dfn, em, var { font-style:normal; }  code, kbd, pre, samp { font-family:couriernew, courier, monospace; }  small{ font-size:12px; }  ul, ol { list-style:none; }  a { text-decoration:none; }  a:hover { text-decoration:underline; }  sup { vertical-align:text-top; }  sub{ vertical-align:text-bottom; }  legend { color:#000; }  fieldset, img { border:0; }  button, input, select, textarea { font-size:100%; }  table { border-collapse:collapse; border-spacing:0; } |

absolute的containing block计算方式跟正常流有什么不同？

position跟display、margin collapse、overflow、float这些特性相互叠加后会怎么样？

对BFC规范的理解？

（W3C CSS 2.1 规范中的一个概念,它决定了元素如何对其内容进行定位,以及与其他元素的关 系和相互作用。）

css定义的权重

以下是权重的规则：标签的权重为1，class的权重为10，id的权重为100，以下例子是演示各种定义的权重值：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | /\*权重为1\*/  div{  }  /\*权重为10\*/  .class1{  }  /\*权重为100\*/  #id1{  }  /\*权重为100+1=101\*/  #id1 div{  }  /\*权重为10+1=11\*/  .class1 div{  }  /\*权重为10+10+1=21\*/  .class1 .class2 div{  } |

如果权重相同，则最后定义的样式会起作用，但是应该避免这种情况出现

解释下浮动和它的工作原理？清除浮动的技巧

用过媒体查询，针对移动端的布局吗？

使用 CSS 预处理器吗？喜欢那个？

SASS

如果需要手动写动画，你认为最小时间间隔是多久，为什么？（阿里）

多数显示器默认频率是60Hz，即1秒刷新60次，所以理论上最小间隔为1/60＊1000ms ＝ 16.7ms

display:inline-block 什么时候会显示间隙？(携程)

移除空格、使用margin负值、使用font-size:0、letter-spacing、word-spacing

## JavaScript

JavaScript原型，原型链 ? 有什么特点？

eval是做什么的？

它的功能是把对应的字符串解析成JS代码并运行； 应该避免使用eval，不安全，非常耗性能（2次，一次解析成js语句，一次执行）。

null，undefined 的区别？

写一个通用的事件侦听器函数

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77 | // 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;        }    }; |

 Node.js的适用场景？

高并发、聊天、实时消息推送

介绍js的基本数据类型

|  |  |
| --- | --- |
| 1  2 | number,string,boolean,object,undefined |

Javascript如何实现继承？

通过原型和构造器

[“1″, “2”, “3”].map(parseInt) 答案是多少？

[1, NaN, NaN] 因为 parseInt 需要两个参数 (val, radix)，其中 radix 表示解析时用的基数。map 传了 3 个 (element, index, array)，对应的 radix 不合法导致解析失败。

如何创建一个对象? （画出此对象的内存图）

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | function Person(name, age) {      this.name = name;      this.age = age;      this.sing = function() { alert(this.name) }    } |

谈谈This对象的理解

this是js的一个关键字，随着函数使用场合不同，this的值会发生变化。

但是有一个总原则，那就是this指的是调用函数的那个对象。

this一般情况下：是全局对象Global。 作为方法调用，那么this就是指这个对象

事件是？IE与火狐的事件机制有什么区别？ 如何阻止冒泡？

1. 我们在网页中的某个操作（有的操作对应多个事件）。例如：当我们点击一个按钮就会产生一个事件。是可以被 JavaScript 侦测到的行为。

2. 事件处理机制：IE是事件冒泡、火狐是 事件捕获；

3. ev.stopPropagation();

什么是闭包（closure），为什么要用它？

执行say667()后,say667()闭包内部变量会存在,而闭包内部函数的内部变量不会存在.使得Javascript的垃圾回收机制GC不会收回say667()所占用的资源，因为say667()的内部函数的执行需要依赖say667()中的变量。这是对闭包作用的非常直白的描述.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | function say667() {      // Local variable that ends up within closure      var num = 666;      var sayAlert = function() { alert(num); }      num++;      return sayAlert;  }    var sayAlert = say667();  sayAlert()//执行结果应该弹出的667 |

“use strict”;是什么意思 ? 使用它的好处和坏处分别是什么？

如何判断一个对象是否属于某个类？

|  |  |
| --- | --- |
| 1  2  3  4  5 | 使用instanceof （待完善）  if(a instanceof Person){      alert('yes');  } |

new操作符具体干了什么呢?

1、创建一个空对象，并且 this 变量引用该对象，同时还继承了该函数的原型。  
2、属性和方法被加入到 this 引用的对象中。  
3、新创建的对象由 this 所引用，并且最后隐式的返回 this 。

|  |  |
| --- | --- |
| 1  2  3  4 | var obj  = {};  obj.\_\_proto\_\_ = Base.prototype;  Base.call(obj); |

Javascript中，有一个函数，执行时对象查找时，永远不会去查找原型，这个函数是？

|  |  |
| --- | --- |
| 1  2 | hasOwnProperty |

JSON 的了解？

JSON(JavaScript Object Notation) 是一种轻量级的数据交换格式。

它是基于JavaScript的一个子集。数据格式简单, 易于读写, 占用带宽小 {'age':'12', 'name':'back'}

js延迟加载的方式有哪些？

defer和async、动态创建DOM方式（用得最多）、按需异步载入js

ajax 是什么?

同步和异步的区别?

如何解决跨域问题?

|  |  |
| --- | --- |
| 1  2 | jsonp、 iframe、window.name、window.postMessage、服务器上设置代理页面 |

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

Jquery与jQuery UI 有啥区别？

* \*jQuery是一个js库，主要提供的功能是选择器，属性修改和事件绑定等等。
* \*jQuery UI则是在jQuery的基础上，利用jQuery的扩展性，设计的插件。
* 提供了一些常用的界面元素，诸如对话框、拖动行为、改变大小行为等等

JQuery的源码看过吗？能不能简单说一下它的实现原理？

jquery 中如何将数组转化为json字符串，然后再转化回来？

jQuery中没有提供这个功能，所以你需要先编写两个jQuery的扩展：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | $.fn.stringifyArray = function(array) {  return JSON.stringify(array)  }    $.fn.parseArray = function(array) {  return JSON.parse(array)  }    然后调用：  $("").stringifyArray(array) |

针对 jQuery 的优化方法？

\*基于Class的选择性的性能相对于Id选择器开销很大，因为需遍历所有DOM元素。

\*频繁操作的DOM，先缓存起来再操作。用Jquery的链式调用更好。

比如：var str=$("a").attr("href"); \*for (var i = size; i < arr.length; i++) {}

for 循环每一次循环都查找了数组 (arr) 的.length 属性，在开始循环的时候设置一个变量来存储这个数字，可以让循环跑得更快：

for (var i = size, length = arr.length; i < length; i++) {}

JQuery一个对象可以同时绑定多个事件，这是如何实现的？

如何判断当前脚本运行在浏览器还是node环境中？（阿里）

通过判断Global对象是否为window，如果不为window，当前脚本没有运行在浏览器中

## 其他问题

你遇到过比较难的技术问题是？你是如何解决的？

常使用的库有哪些？常用的前端开发工具？开发过什么应用或组件？

对Node的优点和缺点提出了自己的看法？

\*（优点）因为Node是基于事件驱动和无阻塞的，所以非常适合处理并发请求， 因此构建在Node上的代理服务器相比其他技术实现（如Ruby）的服务器表现要好得多。 此外，与Node代理服务器交互的客户端代码是由javascript语言编写的， 因此客户端和服务器端都用同一种语言编写，这是非常美妙的事情。

\*（缺点）Node是一个相对新的开源项目，所以不太稳定，它总是一直在变， 而且缺少足够多的第三方库支持。看起来，就像是Ruby/Rails当年的样子。

你有哪些性能优化的方法？（看雅虎14条性能优化原则）

（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。

（8） 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。

http状态码有那些？分别代表是什么意思？

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | 100-199 用于指定客户端应相应的某些动作。  200-299 用于表示请求成功。  300-399 用于已经移动的文件并且常被包含在定位头信息中指定新的地址信息。  400-499 用于指出客户端的错误。  400    1、语义有误，当前请求无法被服务器理解。  401   当前请求需要用户验证  403  服务器已经理解请求，但是拒绝执行它。  500-599 用于支持服务器错误。 503 – 服务不可用 |

一个页面从输入 URL 到页面加载显示完成，这个过程中都发生了什么？（流程说的越详细越好）

* 查找浏览器缓存
* DNS解析、查找该域名对应的IP地址、重定向（301）、发出第二个GET请求
* 进行HTTP协议会话
* 客户端发送报头(请求报头)
* 服务器回馈报头(响应报头)
* html文档开始下载
* 文档树建立，根据标记请求所需指定MIME类型的文件
* 文件显示

{浏览器这边做的工作大致分为以下几步：

加载：根据请求的URL进行域名解析，向服务器发起请求，接收文件（HTML、JS、CSS、图象等）。

解析：对加载到的资源（HTML、JS、CSS等）进行语法解析，建议相应的内部数据结构（比如HTML的DOM树，JS的（对象）属性表，CSS的样式规则等等）}

除了前端以外还了解什么其它技术么？你最最厉害的技能是什么？

你常用的开发工具是什么，为什么？

对前端界面工程师这个职位是怎么样理解的？它的前景会怎么样？

* 前端是最贴近用户的程序员，比后端、数据库、产品经理、运营、安全都近。
  + 实现界面交互
  + 提升用户体验
  + 有了Node.js，前端可以实现服务端的一些事情
* 前端是最贴近用户的程序员，前端的能力就是能让产品从 90分进化到 100 分，甚至更好，
* 参与项目，快速高质量完成实现效果图，精确到1px；
* 与团队成员，UI设计，产品经理的沟通；
* 做好的页面结构，页面重构和用户体验；
* 处理hack，兼容、写出优美的代码格式；
* 针对服务器的优化、拥抱最新前端技术。

加班的看法？

加班就像借钱，原则应当是------救急不救穷

平时如何管理你的项目？

* 先期团队必须确定好全局样式（globe.css），编码模式(utf-8) 等；
* 编写习惯必须一致（例如都是采用继承式的写法，单样式都写成一行）；
* 标注样式编写人，各模块都及时标注（标注关键样式调用的地方）；
* 页面进行标注（例如 页面 模块 开始和结束）；
* CSS跟HTML 分文件夹并行存放，命名都得统一（例如style.css）；
* JS 分文件夹存放 命名以该JS功能为准的英文翻译。
* 图片采用整合的 images.png png8 格式文件使用 尽量整合在一起使用方便将来的管理

说说最近最流行的一些东西吧？常去哪些网站？

|  |  |
| --- | --- |
| 1  2 | Node.js、Mongodb、npm、MVVM、three.js |

你在现在的团队处于什么样的角色，起到了什么明显的作用？

你认为怎样才是全端工程师（Full Stack developer）？

介绍一个你最得意的作品吧？

你的优点是什么？缺点是什么？

最近在学什么？能谈谈你未来3，5年给自己的规划吗？

想问公司的问题？

问公司问题：

* 目前关注哪些最新的Web前端技术（未来的发展方向）？
* 前端团队如何工作的（实现一个产品的流程）？
* 公司的薪资结构是什么样子的？

## 优质网站推荐

1. 极客标签： [http://www.gbtags.com/](http://www.gbtags.com/" \t "http://www.cnblogs.com/dunken/p/_blank)
2. 码农周刊： [http://weekly.manong.io/issues/](http://weekly.manong.io/issues/" \t "http://www.cnblogs.com/dunken/p/_blank)
3. 前端周刊： [http://www.feweekly.com/issues](http://www.feweekly.com/issues" \t "http://www.cnblogs.com/dunken/p/_blank)
4. 极客头条： [http://geek.csdn.net/](http://geek.csdn.net/" \t "http://www.cnblogs.com/dunken/p/_blank)
5. Startup News：[http://news.dbanotes.net/](http://news.dbanotes.net/" \t "http://www.cnblogs.com/dunken/p/_blank)
6. Hacker News： [https://news.ycombinator.com/news](https://news.ycombinator.com/news" \t "http://www.cnblogs.com/dunken/p/_blank)
7. InfoQ： [http://www.infoq.com/](http://www.infoq.com/" \t "http://www.cnblogs.com/dunken/p/_blank)
8. w3cplus： [http://www.w3cplus.com/](http://www.w3cplus.com/" \t "http://www.cnblogs.com/dunken/p/_blank)
9. Stack Overflow： [http://stackoverflow.com/](http://stackoverflow.com/" \t "http://www.cnblogs.com/dunken/p/_blank)
10. Atp： [http://atp-posts.b0.upaiyun.com/posts](http://atp-posts.b0.upaiyun.com/posts/" \t "http://www.cnblogs.com/dunken/p/_blank)

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\_\_");复制代码2 截取字符串abcde

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5-20

var reg = /^[a-zA-Z][a-zA-Z\_0-9]{4,19}$/;

reg.test("a1a\_\_a1a\_\_a1a\_\_a1a\_\_");

2. 截取字符串abcdefg的efg

var str = "abcdefg";

if (/efg/.test(str)) {

var efg = str.substr(str.indexOf("efg"), 3);

alert(efg);

}

3. 判断一个字符串中出现次数最多的字符，统计这个次数

//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

var str = "abcdefgaddda";

var obj = {};

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

var key = str[i];

if (!obj[key]) {

obj[key] = 1;

} else {

obj[key]++;

}

}

/\*遍历这个hash table，获取value最大的key和value\*/

var max = -1;

var max\_key = "";

var key;

for (key in obj) {

if (max < obj[key]) {

max = obj[key];

max\_key = key;

}

}

alert("max:"+max+" max\_key:"+max\_key);

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：

function Animal(name) {

this.name = name;

}

Animal.prototype.getName = function() {alert(this.name)}

function Dog() {};

Dog.prototype = new Animal("Buddy");

Dog.prototype.constructor = Dog;

var dog = new Dog();

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

function GetBytes(str){

var len = str.length;

var bytes = len;

for(var i=0; i<len; i++){

if (str.charCodeAt(i) > 255) bytes++;

}

return bytes;

}

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

var arr = [1 ,1 ,2, 3, 3, 2, 1];

Array.prototype.unique = function(){

var ret = [];

var o = {};

var len = this.length;

for (var i=0; i<len; i++){

var v = this[i];

if (!o[v]){

o[v] = 1;

ret.push(v);

}

}

return ret;

};

alert(arr.unique());

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:

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

（2）构造函数

function Animal(name, color) {

this.name = name;

this.color = color;

}

（3）

<input type="button" id="text" value="点击一下" />

<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>">

var btn = document.getElementById("text");

btn.onclick = function() {

alert(this.value); //此处的this是按钮元素

}

</script>

（4）CSS expression表达式中使用this关键字

<table width="100px" height="100px">

<tr>

<td>

<div style="width:expression(this.parentNode.width);">div element</div>

</td>

</tr>

</table>

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”);

function IsString(str){

return (typeof str == "string" || str.constructor == String);

}

var str = "";

alert(IsString(1));

alert(IsString(str));

alert(IsString(new String(str)));

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”

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>倒计时</title>

</head>

<body>

<input type="text" value="" id="input" size="1000"/>

<script type="text/javascript">

function counter() {

var date = new Date();

var year = date.getFullYear();

var date2 = new Date(year, 12, 31, 23, 59, 59);

var time = (date2 - date)/1000;

var day =Math.floor(time/(24\*60\*60))

var hour = Math.floor(time%(24\*60\*60)/(60\*60))

var minute = Math.floor(time%(24\*60\*60)%(60\*60)/60);

var second = Math.floor(time%(24\*60\*60)%(60\*60)%60);

var str = year + "年还剩"+day+"天"+hour+"时"+minute+"分"+second+"秒";

document.getElementById("input").value = str;

}

window.setInterval("counter()", 1000);

</script>

</body>

</html>

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

<div> <input type=”button” id =”button1″ value=”1″ onclick=”???”>

<input type=”button” id =”button2″ value=”2″ /”> </div>

复制代码

<div>

<input type="button" id ="button1" value="1" onclick="moveBtn(this);">

<input type="button" id ="button2" value="2" />

</div>

<script type="text/javascript">

function moveBtn(obj) {

var clone = obj.cloneNode(true);

var parent = obj.parentNode;

parent.appendChild(clone);

parent.removeChild(obj);

}

</script>

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

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;

};

19. 如何控制alert中的换行

\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）

<!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>鼠标点击页面中的任意标签，alert该标签的名称</title>

<style>

div{ background:#0000FF;width:100px;height:100px;}

span{ background:#00FF00;width:100px;height:100px;}

p{ background:#FF0000;width:100px;height:100px;}

</style>

<script type="text/javascript">

document.onclick = function(evt){

var e = window.event || evt;

var tag = e["target"] || e["srcElement"];

alert(tag.tagName);

};

</script>

</head>

<body>

<div id="div"><span>SPAN</span>DIV</div>

<span>SPAN</span>

<p>P</p>

</body>

</html>

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″;

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"]);

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/）

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?

<ul id=”test”>

<li>这是第一条</li>

<li>这是第二条</li>

<li>这是第三条</li>

</ul>

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。

(function A() {

var index = 0;

var ul = document.getElementById("test");

var obj = {};

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

if (ul.childNodes[i].nodeName.toLowerCase() == "li") {

var li = ul.childNodes[i];

li.onclick = function() {

index++;

alert(index);

}

}

}

})();

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

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

　　Html篇：

1.你做的页面在哪些流览器测试过？这些浏览器的内核分别是什么?

? IE: trident内核

? Firefox：gecko内核

? Safari:webkit内核

? Opera:以前是presto内核，Opera现已改用Google Chrome的Blink内核

? Chrome:Blink(基于webkit，Google与Opera Software共同开发)

2.每个HTML文件里开头都有个很重要的东西，Doctype，知道这是干什么的吗？

　　答案：<!DOCTYPE> 声明位于文档中的最前面的位置，处于 <html> 标签之前。此标签可告知浏览器文档使用哪种 HTML 或 XHTML 规范。

（重点：告诉浏览器按照何种规范解析页面）

3.Quirks模式是什么？它和Standards模式有什么区别

　　答案：

　　从IE6开始，引入了Standards模式，标准模式中，浏览器尝试给符合标准的文档在规范上的正确处理达到在指定浏览器中的程度。

　　在IE6之前CSS还不够成熟，所以IE5等之前的浏览器对CSS的支持很差， IE6将对CSS提供更好的支持，然而这时的问题就来了，因为有很多页面是基于旧的布局方式写的，而如果IE6 支持CSS则将令这些页面显示不正常，如何在即保证不破坏现有页面，又提供新的渲染机制呢？

　　在写程序时我们也会经常遇到这样的问题，如何保证原来的接口不变，又提供更强大的功能，尤其是新功能不兼容旧功能时。遇到这种问题时的一个常见做法是增加参数和分支，即当某个参数为真时，我们就使用新功能，而如果这个参数 不为真时，就使用旧功能，这样就能不破坏原有的程序，又提供新功能。IE6也是类似这样做的，它将DTD当成了这个“参数”，因为以前的页面大家都不会去写DTD，所以IE6就假定 如果写了DTD，就意味着这个页面将采用对CSS支持更好的布局，而如果没有，则采用兼容之前的布局方式。这就是Quirks模式（怪癖模式，诡异模式，怪异模式）。

　　区别：

　　总体会有布局、样式解析和脚本执行三个方面的区别。

　　盒模型：在W3C标准中，如果设置一个元素的宽度和高度，指的是元素内容的宽度和高度，而在Quirks 模式下，IE的宽度和高度还包含了padding和border。

设置行内元素的高宽：在Standards模式下，给<span>等行内元素设置wdith和height都不会生效，而在quirks模式下，则会生效。

　　设置百分比的高度：在standards模式下，一个元素的高度是由其包含的内容来决定的，如果父元素没有设置百分比的高度，子元素设置一个百分比的高度是无效的用margin:0 auto设置水平居中：使用margin:0 auto在standards模式下可以使元素水平居中，但在quirks模式下却会失效。

　　（还有很多，答出什么不重要，关键是看他答出的这些是不是自己经验遇到的，还是说都是看文章看的，甚至完全不知道。）

　　4.div+css的布局较table布局有什么优点？

?改版的时候更方便 只要改css文件。

?页面加载速度更快、结构化清晰、页面显示简洁。

?表现与结构相分离。

?易于优化（seo）搜索引擎更友好，排名更容易靠前。

　　5.a：img的alt与title有何异同？b：strong与em的异同？

　　答案：

　　a:

?alt(alt text):为不能显示图像、窗体或applets的用户代理（UA），alt属性用来指定替换文字。替换文字的语言由lang属性指定。(在IE浏览器下会在没有title时把alt当成 tool tip显示)

?title(tool tip):该属性为设置该属性的元素提供建议性的信息。

　　b:

?strong:粗体强调标签，强调，表示内容的重要性

?em:斜体强调标签，更强烈强调，表示内容的强调点

　　6.你能描述一下渐进增强和优雅降级之间的不同吗?

?渐进增强 progressive enhancement：针对低版本浏览器进行构建页面，保证最基本的功能，然后再针对高级浏览器进行效果、交互等改进和追加功能达到更好的用户体验。

?优雅降级 graceful degradation：一开始就构建完整的功能，然后再针对低版本浏览器进行兼容。

　　区别：优雅降级是从复杂的现状开始，并试图减少用户体验的供给，而渐进增强则是从一个非常基础的，能够起作用的版本开始，并不断扩充，以适应未来环境的需要。降级（功能衰减）意味着往回看；而渐进增强则意味着朝前看，同时保证其根基处于安全地带。

　　“优雅降级”观点

　　“优雅降级”观点认为应该针对那些最高级、最完善的浏览器来设计网站。而将那些被认为“过时”或有功能缺失的浏览器下的测试工作安排在开发周期的最后阶段，并把测试对象限定为主流浏览器（如 IE、Mozilla 等）的前一个版本。

　　在这种设计范例下，旧版的浏览器被认为仅能提供“简陋却无妨 (poor, but passable)” 的浏览体验。你可以做一些小的调整来适应某个特定的浏览器。但由于它们并非我们所关注的焦点，因此除了修复较大的错误之外，其它的差异将被直接忽略。

　　“渐进增强”观点

　　“渐进增强”观点则认为应关注于内容本身。

　　内容是我们建立网站的诱因。有的网站展示它，有的则收集它，有的寻求，有的操作，还有的网站甚至会包含以上的种种，但相同点是它们全都涉及到内容。这使得“渐进增强”成为一种更为合理的设计范例。这也是它立即被 Yahoo! 所采纳并用以构建其“分级式浏览器支持 (Graded Browser Support)”策略的原因所在。

　　那么问题了。现在产品经理看到IE6,7,8网页效果相对高版本现代浏览器少了很多圆角，阴影（CSS3），要求兼容（使用图片背景，放弃CSS3），你会如何说服他？

　　（自由发挥）

　　7.为什么利用多个域名来存储网站资源会更有效？

?CDN缓存更方便

?突破浏览器并发限制

?节约cookie带宽

?节约主域名的连接数，优化页面响应速度

?防止不必要的安全问题

　　8.请谈一下你对网页标准和标准制定机构重要性的理解。

　　（无标准答案）网页标准和标准制定机构都是为了能让web发展的更‘健康’，开发者遵循统一的标准，降低开发难度，开发成本，SEO也会更好做，也不会因为滥用代码导致各种BUG、安全问题，最终提高网站易用性。

　　9.请描述一下cookies，sessionStorage和localStorage的区别？

　　sessionStorage用于本地存储一个会话（session）中的数据，这些数据只有在同一个会话中的页面才能访问并且当会话结束后数据也随之销毁。因此sessionStorage不是一种持久化的本地存储，仅仅是会话级别的存储。而localStorage用于持久化的本地存储，除非主动删除数据，否则数据是永远不会过期的。

　　web storage和cookie的区别

?Web Storage的概念和cookie相似，区别是它是为了更大容量存储设计的。Cookie的大小是受限的，并且每次你请求一个新的页面的时候Cookie都会被发送过去，这样无形中浪费了带宽，另外cookie还需要指定作用域，不可以跨域调用。

?除此之外，Web Storage拥有setItem,getItem,removeItem,clear等方法，不像cookie需要前端开发者自己封装setCookie，getCookie。但是Cookie也是不可以或缺的：Cookie的作用是与服务器进行交互，作为HTTP规范的一部分而存在 ，而Web Storage仅仅是为了在本地“存储”数据而生。

　　10.简述一下src与href的区别。

　　答案：

　　src用于替换当前元素，href用于在当前文档和引用资源之间确立联系。

　　src是source的缩写，指向外部资源的位置，指向的内容将会嵌入到文档中当前标签所在位置；在请求src资源时会将其指向的资源下载并应用到文档内，例如js脚本，img图片和frame等元素。

　　<script src ="js.js"></script>

　　当浏览器解析到该元素时，会暂停其他资源的下载和处理，直到将该资源加载、编译、执行完毕，图片和框架等元素也如此，类似于将所指向资源嵌入当前标签内。这也是为什么将js脚本放在底部而不是头部。

　　href是Hypertext Reference的缩写，指向网络资源所在位置，建立和当前元素（锚点）或当前文档（链接）之间的链接，如果我们在文档中添加

　　<link href="common.css" rel="stylesheet"/>

　　那么浏览器会识别该文档为css文件，就会并行下载资源并且不会停止对当前文档的处理。这也是为什么建议使用link方式来加载css，而不是使用@import方式。

　　11.知道的网页制作会用到的图片格式有哪些？

　　答案：

　　png-8，png-24，jpeg，gif，svg。

　　但是上面的那些都不是面试官想要的最后答案。面试官希望听到是Webp,Apng。（是否有关注新技术，新鲜事物）

　　科普一下Webp：WebP格式，谷歌（google）开发的一种旨在加快图片加载速度的图片格式。图片压缩体积大约只有JPEG的2/3，并能节省大量的服务器带宽资源和数据空间。Facebook Ebay等知名网站已经开始测试并使用WebP格式。

　　在质量相同的情况下，WebP格式图像的体积要比JPEG格式图像小40%。

Apng：全称是“Animated Portable Network Graphics”, 是PNG的位图动画扩展，可以实现png格式的动态图片效果。04年诞生，但一直得不到各大浏览器厂商的支持，直到日前得到 iOS safari 8的支持，有望代替GIF成为下一代动态图标准。

　　12.知道什么是微格式吗？谈谈理解。在前端构建中应该考虑微格式吗？

　　答案：

　　微格式（Microformats）是一种让机器可读的语义化XHTML词汇的集合，是结构化数据的开放标准。是为特殊应用而制定的特殊格式。

　　优点：将智能数据添加到网页上，让网站内容在搜索引擎结果界面可以显示额外的提示。（应用范例：豆瓣，有兴趣自行google）

　　13.在css/js代码上线之后开发人员经常会优化性能，从用户刷新网页开始，一次js请求一般情况下有哪些地方会有缓存处理？

　　答案：dns缓存，cdn缓存，浏览器缓存，服务器缓存。

　　14.一个页面上有大量的图片（大型电商网站），加载很慢，你有哪些方法优化这些图片的加载，给用户更好的体验。

图片懒加载，在页面上的未可视区域可以添加一个滚动条事件，判断图片位置与浏览器顶端的距离与页面的距离，如果前者小于后者，优先加载。

如果为幻灯片、相册等，可以使用图片预加载技术，将当前展示图片的前一张和后一张优先下载。

如果图片为css图片，可以使用CSSsprite，SVGsprite，Iconfont、Base64等技术。

如果图片过大，可以使用特殊编码的图片，加载时会先加载一张压缩的特别厉害的缩略图，以提高用户体验。

如果图片展示区域小于图片的真实大小，则因在服务器端根据业务需要先行进行图片压缩，图片压缩后大小与展示一致。

　　15.你如何理解HTML结构的语义化？

去掉或样式丢失的时候能让页面呈现清晰的结构：

html本身是没有表现的，我们看到例如<h1>是粗体，字体大小2em，加粗；<strong>是加粗的，不要认为这是html的表现，这些其实html默认的css样式在起作用，所以去掉或样式丢失的时候能让页面呈现清晰的结构不是语义化的HTML结构的优点，但是浏览器都有有默认样式，默认样式的目的也是为了更好的表达html的语义，可以说浏览器的默认样式和语义化的HTML结构是不可分割的。

屏幕阅读器（如果访客有视障）会完全根据你的标记来“读”你的网页.

　　例如,如果你使用的含语义的标记,屏幕阅读器就会“逐个拼出”你的单词,而不是试着去对它完整发音.

PDA、手机等设备可能无法像普通电脑的浏览器一样来渲染网页（通常是因为这些设备对CSS的支持较弱）

　　使用语义标记可以确保这些设备以一种有意义的方式来渲染网页.理想情况下,观看设备的任务是符合设备本身的条件来渲染网页.

　　语义标记为设备提供了所需的相关信息,就省去了你自己去考虑所有可能的显示情况（包括现有的或者将来新的设备）.例如,一部手机可以选择使一段标记了标题的文字以粗体显示.而掌上电脑可能会以比较大的字体来显示.无论哪种方式一旦你对文本标记为标题,您就可以确信读取设备将根据其自身的条件来合适地显示页面.

搜索引擎的爬虫也依赖于标记来确定上下文和各个关键字的权重

　　过去你可能还没有考虑搜索引擎的爬虫也是网站的“访客”,但现在它们他们实际上是极其宝贵的用户.没有他们的话,搜索引擎将无法索引你的网站,然后一般用户将很难过来访问.

你的页面是否对爬虫容易理解非常重要,因为爬虫很大程度上会忽略用于表现的标记,而只注重语义标记.

　　因此,如果页面文件的标题被标记,而不是,那么这个页面在搜索结果的位置可能会比较靠后.除了提升易用性外,语义标记有利于正确使用CSS和JavaScript,因为其本身提供了许多“钩钩”来应用页面的样式与行为.

SEO主要还是靠你网站的内容和外部链接的。

便于团队开发和维护

　　W3C给我们定了一个很好的标准，在团队中大家都遵循这个标准，可以减少很多差异化的东西，方便开发和维护，提高开发效率，甚至实现模块化开发。

　　16.谈谈以前端角度出发做好SEO需要考虑什么？

了解搜索引擎如何抓取网页和如何索引网页

　　你需要知道一些搜索引擎的基本工作原理，各个搜索引擎之间的区别，搜索机器人（SE robot 或叫 web crawler）如何进行工作，搜索引擎如何对搜索结果进行排序等等。

Meta标签优化

　　主要包括主题（Title)，网站描述(Description)，和关键词（Keywords）。还有一些其它的隐藏文字比如Author（作者），Category（目录），Language（编码语种）等。

如何选取关键词并在网页中放置关键词

　　搜索就得用关键词。关键词分析和选择是SEO最重要的工作之一。首先要给网站确定主关键词（一般在5个上下），然后针对这些关键词进行优化，包括关键词密度（Density），相关度（Relavancy），突出性（Prominency）等等。

了解主要的搜索引擎

　　虽然搜索引擎有很多，但是对网站流量起决定作用的就那么几个。比如英文的主要有Google，Yahoo，Bing等；中文的有百度，搜狗，有道等。不同的搜索引擎对页面的抓取和索引、排序的规则都不一样。还要了解各搜索门户和搜索引擎之间的关系，比如AOL网页搜索用的是Google的搜索技术，MSN用的是Bing的技术。

主要的互联网目录

　　Open Directory自身不是搜索引擎，而是一个大型的网站目录，他和搜索引擎的主要区别是网站内容的收集方式不同。目录是人工编辑的，主要收录网站主页；搜索引擎是自动收集的，除了主页外还抓取大量的内容页面。

按点击付费的搜索引擎

　　搜索引擎也需要生存，随着互联网商务的越来越成熟，收费的搜索引擎也开始大行其道。最典型的有Overture和百度，当然也包括Google的广告项目Google Adwords。越来越多的人通过搜索引擎的点击广告来定位商业网站，这里面也大有优化和排名的学问，你得学会用最少的广告投入获得最多的点击。

搜索引擎登录

　　网站做完了以后，别躺在那里等着客人从天而降。要让别人找到你，最简单的办法就是将网站提交（submit）到搜索引擎。如果你的是商业网站，主要的搜索引擎和目录都会要求你付费来获得收录（比如Yahoo要299美元），但是好消息是（至少到目前为止）最大的搜索引擎Google目前还是免费，而且它主宰着60％以上的搜索市场。

链接交换和链接广泛度（Link Popularity）

　　网页内容都是以超文本（Hypertext）的方式来互相链接的，网站之间也是如此。除了搜索引擎以外，人们也每天通过不同网站之间的链接来Surfing（“冲浪”）。其它网站到你的网站的链接越多，你也就会获得更多的访问量。更重要的是，你的网站的外部链接数越多，会被搜索引擎认为它的重要性越大，从而给你更高的排名。

合理的标签使用

Css篇：

1.有哪项方式可以对一个DOM设置它的CSS样式？

?外部样式表，引入一个外部css文件

?内部样式表，将css代码放在 <head> 标签内部

?内联样式，将css样式直接定义在 HTML 元素内部

2.CSS都有哪些选择器？

?派生选择器（用HTML标签申明）

?id选择器（用DOM的ID申明）

?类选择器（用一个样式类名申明）

?属性选择器（用DOM的属性申明，属于CSS2，IE6不支持，不常用，不知道就算了）

除了前3种基本选择器，还有一些扩展选择器，包括

?后代选择器（利用空格间隔，比如div .a{ }）

?群组选择器（利用逗号间隔，比如p,div,#a{ }）

那么问题来了，CSS选择器的优先级是怎么样定义的？

基本原则：

一般而言，选择器越特殊，它的优先级越高。也就是选择器指向的越准确，它的优先级就越高。

复杂的计算方法：

?用1表示派生选择器的优先级

?用10表示类选择器的优先级

?用100标示ID选择器的优先级

?div.test1 .span var 优先级 1+10 +10 +1

?span#xxx .songs li 优先级1+100 + 10 + 1

?#xxx li 优先级 100 +1

那么问题来了，看下列代码，<p>标签内的文字是什么颜色的？。

<style>

.classA{ color:blue;}

.classB{ color:red;}

</style>

<body>

<p class='classB classA'> 123 </p>

</body>

答案：red。与样式定义在文件中的先后顺序有关，即是后面的覆盖前面的，与在<p class='classB classA'>中的先后关系无关。

3.CSS中可以通过哪些属性定义，使得一个DOM元素不显示在浏览器可视范围内？

最基本的：

设置display属性为none，或者设置visibility属性为hidden

技巧性：

设置宽高为0，设置透明度为0，设置z-index位置在-1000

4.超链接访问过后hover样式就不出现的问题是什么？如何解决？

答案：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（link,visited,hover,active）

5.什么是Css Hack？ie6,7,8的hack分别是什么？

答案：针对不同的浏览器写不同的CSS code的过程，就是CSS hack。

示例如下：

#test {

width:300px;

height:300px;

background-color:blue; /\*firefox\*/

background-color:red\9; /\*all ie\*/

background-color:yellow\0; /\*ie8\*/

+background-color:pink; /\*ie7\*/

\_background-color:orange; /\*ie6\*/ }

:root #test { background-color:purple\9; } /\*ie9\*/

@media all and (min-width:0px){ #test {background-color:black\0;} } /\*opera\*/

@media screen and (-webkit-min-device-pixel-ratio:0){ #test {background-color:gray;} } /\*chrome and safari\*/

6.请用Css写一个简单的幻灯片效果页面

　　答案：知道是要用css3。使用animation动画实现一个简单的幻灯片效果。

/\*\*HTML\*\*/

div.ani

/\*\*css\*\*/

.ani{

width:480px;

height:320px;

margin:50px auto;

overflow: hidden;

box-shadow:0 0 5px rgba(0,0,0,1);

background-size: cover;

background-position: center;

-webkit-animation-name: "loops";

-webkit-animation-duration: 20s;

-webkit-animation-iteration-count: infinite;

}

@-webkit-keyframes "loops" {

0% {

background:url(http://d.hiphotos.baidu.com/image/w%3D400/sign=c01e6adca964034f0fcdc3069fc27980/e824b899a9014c08e5e38ca4087b02087af4f4d3.jpg) no-repeat;

}

25% {

background:url(http://b.hiphotos.baidu.com/image/w%3D400/sign=edee1572e9f81a4c2632edc9e72b6029/30adcbef76094b364d72bceba1cc7cd98c109dd0.jpg) no-repeat;

}

50% {

background:url(http://b.hiphotos.baidu.com/image/w%3D400/sign=937dace2552c11dfded1be2353266255/d8f9d72a6059252d258e7605369b033b5bb5b912.jpg) no-repeat;

}

75% {

background:url(http://g.hiphotos.baidu.com/image/w%3D400/sign=7d37500b8544ebf86d71653fe9f9d736/0df431adcbef76095d61f0972cdda3cc7cd99e4b.jpg) no-repeat;

}

100% {

background:url(http://c.hiphotos.baidu.com/image/w%3D400/sign=cfb239ceb0fb43161a1f7b7a10a54642/3b87e950352ac65ce2e73f76f9f2b21192138ad1.jpg) no-repeat;

}

}

7.行内元素和块级元素的具体区别是什么？行内元素的padding和margin可设置吗？

块级元素(block)特性：

?总是独占一行，表现为另起一行开始，而且其后的元素也必须另起一行显示;

?宽度(width)、高度(height)、内边距(padding)和外边距(margin)都可控制;

内联元素(inline)特性：

?和相邻的内联元素在同一行;

?宽度(width)、高度(height)、内边距的top/bottom(padding-top/padding-bottom)和外边距的top/bottom(margin-top/margin-bottom)都不可改变（也就是padding和margin的left和right是可以设置的），就是里面文字或图片的大小。

那么问题来了，浏览器还有默认的天生inline-block元素（拥有内在尺寸，可设置高宽，但不会自动换行），有哪些？

答案：<input> 、<img> 、<button> 、<textarea> 、<label>。

8.什么是外边距重叠？重叠的结果是什么？

答案：

外边距重叠就是margin-collapse。

在CSS当中，相邻的两个盒子（可能是兄弟关系也可能是祖先关系）的外边距可以结合成一个单独的外边距。这种合并外边距的方式被称为折叠，并且因而所结合成的外边距称为折叠外边距。

折叠结果遵循下列计算规则：

1.两个相邻的外边距都是正数时，折叠结果是它们两者之间较大的值。

2.两个相邻的外边距都是负数时，折叠结果是两者绝对值的较大值。

3.两个外边距一正一负时，折叠结果是两者的相加的和。

9.rgba()和opacity的透明效果有什么不同？

答案：

rgba()和opacity都能实现透明效果，但最大的不同是opacity作用于元素，以及元素内的所有内容的透明度，

而rgba()只作用于元素的颜色或其背景色。（设置rgba透明的元素的子元素不会继承透明效果！）

10.css中可以让文字在垂直和水平方向上重叠的两个属性是什么？

答案：

垂直方向：line-height

水平方向：letter-spacing

那么问题来了，关于letter-spacing的妙用知道有哪些么？

答案:可以用于消除inline-block元素间的换行符空格间隙问题。

11.如何垂直居中一个浮动元素？

// 方法一：已知元素的高宽

#div1{

background-color:#6699FF;

width:200px;

height:200px;

position: absolute; //父元素需要相对定位

top: 50%;

left: 50%;

margin-top:-100px ; //二分之一的height，width

margin-left: -100px;

}

//方法二:未知元素的高宽

#div1{

width: 200px;

height: 200px;

background-color: #6699FF;

margin:auto;

position: absolute; //父元素需要相对定位

left: 0;

top: 0;

rightright: 0;

bottombottom: 0;

}

那么问题来了，如何垂直居中一个<img>?（用更简便的方法。）

#container //<img>的容器设置如下

{

display:table-cell;

text-align:center;

vertical-align:middle;

}

12.px和em的区别。

　　px和em都是长度单位，区别是，px的值是固定的，指定是多少就是多少，计算比较容易。em得值不是固定的，并且em会继承父级元素的字体大小。

　　浏览器的默认字体高都是16px。所以未经调整的浏览器都符合: 1em=16px。那么12px=0.75em, 10px=0.625em。

13.描述一个"reset"的CSS文件并如何使用它。知道normalize.css吗？你了解他们的不同之处？

　　重置样式非常多，凡是一个前端开发人员肯定有一个常用的重置CSS文件并知道如何使用它们。他们是盲目的在做还是知道为什么这么做呢？原因是不同的浏览器对一些元素有不同的默认样式，如果你不处理，在不同的浏览器下会存在必要的风险，或者更有戏剧性的性发生。

　　你可能会用Normalize来代替你的重置样式文件。它没有重置所有的样式风格，但仅提供了一套合理的默认样式值。既能让众多浏览器达到一致和合理，但又不扰乱其他的东西（如粗体的标题）。

在这一方面，无法做每一个复位重置。它也确实有些超过一个重置，它处理了你永远都不用考虑的怪癖，像HTML的audio元素不一致或line-height不一致。

　　14.Sass、LESS是什么？大家为什么要使用他们？

　　他们是CSS预处理器。他是CSS上的一种抽象层。他们是一种特殊的语法/语言编译成CSS。

　　例如Less是一种动态样式语言. 将CSS赋予了动态语言的特性，如变量，继承，运算， 函数. LESS 既可以在客户端上运行 (支持IE 6+, Webkit, Firefox)，也可一在服务端运行 (借助 Node.js)。

为什么要使用它们？

?结构清晰，便于扩展。

?可以方便地屏蔽浏览器私有语法差异。这个不用多说，封装对浏览器语法差异的重复处理，减少无意义的机械劳动。

?可以轻松实现多重继承。

?完全兼容 CSS 代码，可以方便地应用到老项目中。LESS 只是在 CSS 语法上做了扩展，所以老的 CSS 代码也可以与 LESS 代码一同编译。

15.display:none与visibility:hidden的区别是什么？

?display : 隐藏对应的元素但不挤占该元素原来的空间。

?visibility: 隐藏对应的元素并且挤占该元素原来的空间。

　　即是，使用CSS display:none属性后，HTML元素（对象）的宽度、高度等各种属性值都将“丢失”;而使用visibility:hidden属性后，HTML元素（对象）仅仅是在视觉上看不见（完全透明），而它所占据的空间位置仍然存在。

16.知道css有个content属性吗？有什么作用？有什么应用？

　　答案：

　　知道。css的content属性专门应用在 before/after 伪元素上，用于来插入生成内容。

　　最常见的应用是利用伪类清除浮动。

//一种常见利用伪类清除浮动的代码

.clearfix:after {

content:"."; //这里利用到了content属性

display:block;

height:0;

visibility:hidden;

clear:both; }

.clearfix {

\*zoom:1;

}

after伪元素通过 content 在元素的后面生成了内容为一个点的块级元素，再利用clear:both清除浮动。

　　那么问题继续还有，知道css计数器（序列数字字符自动递增）吗？如何通过css content属性实现css计数器？

　　答案：css计数器是通过设置counter-reset 、counter-increment 两个属性 、及 counter()/counters()一个方法配合after / before 伪类实现。 BAT及各大互联网公司2014前端笔试面试题--JavaScript篇

　　Html篇：

1.你做的页面在哪些流览器测试过？这些浏览器的内核分别是什么?

? IE: trident内核

? Firefox：gecko内核

? Safari:webkit内核

? Opera:以前是presto内核，Opera现已改用Google Chrome的Blink内核

? Chrome:Blink(基于webkit，Google与Opera Software共同开发)

2.每个HTML文件里开头都有个很重要的东西，Doctype，知道这是干什么的吗？

　　答案：<!DOCTYPE> 声明位于文档中的最前面的位置，处于 <html> 标签之前。此标签可告知浏览器文档使用哪种 HTML 或 XHTML 规范。

（重点：告诉浏览器按照何种规范解析页面）

3.Quirks模式是什么？它和Standards模式有什么区别

　　答案：

　　从IE6开始，引入了Standards模式，标准模式中，浏览器尝试给符合标准的文档在规范上的正确处理达到在指定浏览器中的程度。

　　在IE6之前CSS还不够成熟，所以IE5等之前的浏览器对CSS的支持很差， IE6将对CSS提供更好的支持，然而这时的问题就来了，因为有很多页面是基于旧的布局方式写的，而如果IE6 支持CSS则将令这些页面显示不正常，如何在即保证不破坏现有页面，又提供新的渲染机制呢？

　　在写程序时我们也会经常遇到这样的问题，如何保证原来的接口不变，又提供更强大的功能，尤其是新功能不兼容旧功能时。遇到这种问题时的一个常见做法是增加参数和分支，即当某个参数为真时，我们就使用新功能，而如果这个参数 不为真时，就使用旧功能，这样就能不破坏原有的程序，又提供新功能。IE6也是类似这样做的，它将DTD当成了这个“参数”，因为以前的页面大家都不会去写DTD，所以IE6就假定 如果写了DTD，就意味着这个页面将采用对CSS支持更好的布局，而如果没有，则采用兼容之前的布局方式。这就是Quirks模式（怪癖模式，诡异模式，怪异模式）。

　　区别：

　　总体会有布局、样式解析和脚本执行三个方面的区别。

　　盒模型：在W3C标准中，如果设置一个元素的宽度和高度，指的是元素内容的宽度和高度，而在Quirks 模式下，IE的宽度和高度还包含了padding和border。

设置行内元素的高宽：在Standards模式下，给<span>等行内元素设置wdith和height都不会生效，而在quirks模式下，则会生效。

　　设置百分比的高度：在standards模式下，一个元素的高度是由其包含的内容来决定的，如果父元素没有设置百分比的高度，子元素设置一个百分比的高度是无效的用margin:0 auto设置水平居中：使用margin:0 auto在standards模式下可以使元素水平居中，但在quirks模式下却会失效。

　　（还有很多，答出什么不重要，关键是看他答出的这些是不是自己经验遇到的，还是说都是看文章看的，甚至完全不知道。）

　　4.div+css的布局较table布局有什么优点？

?改版的时候更方便 只要改css文件。

?页面加载速度更快、结构化清晰、页面显示简洁。

?表现与结构相分离。

?易于优化（seo）搜索引擎更友好，排名更容易靠前。

　　5.a：img的alt与title有何异同？b：strong与em的异同？

　　答案：

　　a:

?alt(alt text):为不能显示图像、窗体或applets的用户代理（UA），alt属性用来指定替换文字。替换文字的语言由lang属性指定。(在IE浏览器下会在没有title时把alt当成 tool tip显示)

?title(tool tip):该属性为设置该属性的元素提供建议性的信息。

　　b:

?strong:粗体强调标签，强调，表示内容的重要性

?em:斜体强调标签，更强烈强调，表示内容的强调点

　　6.你能描述一下渐进增强和优雅降级之间的不同吗?

?渐进增强 progressive enhancement：针对低版本浏览器进行构建页面，保证最基本的功能，然后再针对高级浏览器进行效果、交互等改进和追加功能达到更好的用户体验。

?优雅降级 graceful degradation：一开始就构建完整的功能，然后再针对低版本浏览器进行兼容。

　　区别：优雅降级是从复杂的现状开始，并试图减少用户体验的供给，而渐进增强则是从一个非常基础的，能够起作用的版本开始，并不断扩充，以适应未来环境的需要。降级（功能衰减）意味着往回看；而渐进增强则意味着朝前看，同时保证其根基处于安全地带。

　　“优雅降级”观点

　　“优雅降级”观点认为应该针对那些最高级、最完善的浏览器来设计网站。而将那些被认为“过时”或有功能缺失的浏览器下的测试工作安排在开发周期的最后阶段，并把测试对象限定为主流浏览器（如 IE、Mozilla 等）的前一个版本。

　　在这种设计范例下，旧版的浏览器被认为仅能提供“简陋却无妨 (poor, but passable)” 的浏览体验。你可以做一些小的调整来适应某个特定的浏览器。但由于它们并非我们所关注的焦点，因此除了修复较大的错误之外，其它的差异将被直接忽略。

　　“渐进增强”观点

　　“渐进增强”观点则认为应关注于内容本身。

　　内容是我们建立网站的诱因。有的网站展示它，有的则收集它，有的寻求，有的操作，还有的网站甚至会包含以上的种种，但相同点是它们全都涉及到内容。这使得“渐进增强”成为一种更为合理的设计范例。这也是它立即被 Yahoo! 所采纳并用以构建其“分级式浏览器支持 (Graded Browser Support)”策略的原因所在。

　　那么问题了。现在产品经理看到IE6,7,8网页效果相对高版本现代浏览器少了很多圆角，阴影（CSS3），要求兼容（使用图片背景，放弃CSS3），你会如何说服他？

　　（自由发挥）

　　7.为什么利用多个域名来存储网站资源会更有效？

?CDN缓存更方便

?突破浏览器并发限制

?节约cookie带宽

?节约主域名的连接数，优化页面响应速度

?防止不必要的安全问题

　　8.请谈一下你对网页标准和标准制定机构重要性的理解。

　　（无标准答案）网页标准和标准制定机构都是为了能让web发展的更‘健康’，开发者遵循统一的标准，降低开发难度，开发成本，SEO也会更好做，也不会因为滥用代码导致各种BUG、安全问题，最终提高网站易用性。

　　9.请描述一下cookies，sessionStorage和localStorage的区别？

　　sessionStorage用于本地存储一个会话（session）中的数据，这些数据只有在同一个会话中的页面才能访问并且当会话结束后数据也随之销毁。因此sessionStorage不是一种持久化的本地存储，仅仅是会话级别的存储。而localStorage用于持久化的本地存储，除非主动删除数据，否则数据是永远不会过期的。

　　web storage和cookie的区别

?Web Storage的概念和cookie相似，区别是它是为了更大容量存储设计的。Cookie的大小是受限的，并且每次你请求一个新的页面的时候Cookie都会被发送过去，这样无形中浪费了带宽，另外cookie还需要指定作用域，不可以跨域调用。

?除此之外，Web Storage拥有setItem,getItem,removeItem,clear等方法，不像cookie需要前端开发者自己封装setCookie，getCookie。但是Cookie也是不可以或缺的：Cookie的作用是与服务器进行交互，作为HTTP规范的一部分而存在 ，而Web Storage仅仅是为了在本地“存储”数据而生。

　　10.简述一下src与href的区别。

　　答案：

　　src用于替换当前元素，href用于在当前文档和引用资源之间确立联系。

　　src是source的缩写，指向外部资源的位置，指向的内容将会嵌入到文档中当前标签所在位置；在请求src资源时会将其指向的资源下载并应用到文档内，例如js脚本，img图片和frame等元素。

　　<script src ="js.js"></script>

　　当浏览器解析到该元素时，会暂停其他资源的下载和处理，直到将该资源加载、编译、执行完毕，图片和框架等元素也如此，类似于将所指向资源嵌入当前标签内。这也是为什么将js脚本放在底部而不是头部。

　　href是Hypertext Reference的缩写，指向网络资源所在位置，建立和当前元素（锚点）或当前文档（链接）之间的链接，如果我们在文档中添加

　　<link href="common.css" rel="stylesheet"/>

　　那么浏览器会识别该文档为css文件，就会并行下载资源并且不会停止对当前文档的处理。这也是为什么建议使用link方式来加载css，而不是使用@import方式。

　　11.知道的网页制作会用到的图片格式有哪些？

　　答案：

　　png-8，png-24，jpeg，gif，svg。

　　但是上面的那些都不是面试官想要的最后答案。面试官希望听到是Webp,Apng。（是否有关注新技术，新鲜事物）

　　科普一下Webp：WebP格式，谷歌（google）开发的一种旨在加快图片加载速度的图片格式。图片压缩体积大约只有JPEG的2/3，并能节省大量的服务器带宽资源和数据空间。Facebook Ebay等知名网站已经开始测试并使用WebP格式。

　　在质量相同的情况下，WebP格式图像的体积要比JPEG格式图像小40%。

Apng：全称是“Animated Portable Network Graphics”, 是PNG的位图动画扩展，可以实现png格式的动态图片效果。04年诞生，但一直得不到各大浏览器厂商的支持，直到日前得到 iOS safari 8的支持，有望代替GIF成为下一代动态图标准。

　　12.知道什么是微格式吗？谈谈理解。在前端构建中应该考虑微格式吗？

　　答案：

　　微格式（Microformats）是一种让机器可读的语义化XHTML词汇的集合，是结构化数据的开放标准。是为特殊应用而制定的特殊格式。

　　优点：将智能数据添加到网页上，让网站内容在搜索引擎结果界面可以显示额外的提示。（应用范例：豆瓣，有兴趣自行google）

　　13.在css/js代码上线之后开发人员经常会优化性能，从用户刷新网页开始，一次js请求一般情况下有哪些地方会有缓存处理？

　　答案：dns缓存，cdn缓存，浏览器缓存，服务器缓存。

　　14.一个页面上有大量的图片（大型电商网站），加载很慢，你有哪些方法优化这些图片的加载，给用户更好的体验。

图片懒加载，在页面上的未可视区域可以添加一个滚动条事件，判断图片位置与浏览器顶端的距离与页面的距离，如果前者小于后者，优先加载。

如果为幻灯片、相册等，可以使用图片预加载技术，将当前展示图片的前一张和后一张优先下载。

如果图片为css图片，可以使用CSSsprite，SVGsprite，Iconfont、Base64等技术。

如果图片过大，可以使用特殊编码的图片，加载时会先加载一张压缩的特别厉害的缩略图，以提高用户体验。

如果图片展示区域小于图片的真实大小，则因在服务器端根据业务需要先行进行图片压缩，图片压缩后大小与展示一致。

　　15.你如何理解HTML结构的语义化？

去掉或样式丢失的时候能让页面呈现清晰的结构：

html本身是没有表现的，我们看到例如<h1>是粗体，字体大小2em，加粗；<strong>是加粗的，不要认为这是html的表现，这些其实html默认的css样式在起作用，所以去掉或样式丢失的时候能让页面呈现清晰的结构不是语义化的HTML结构的优点，但是浏览器都有有默认样式，默认样式的目的也是为了更好的表达html的语义，可以说浏览器的默认样式和语义化的HTML结构是不可分割的。

屏幕阅读器（如果访客有视障）会完全根据你的标记来“读”你的网页.

　　例如,如果你使用的含语义的标记,屏幕阅读器就会“逐个拼出”你的单词,而不是试着去对它完整发音.

PDA、手机等设备可能无法像普通电脑的浏览器一样来渲染网页（通常是因为这些设备对CSS的支持较弱）

　　使用语义标记可以确保这些设备以一种有意义的方式来渲染网页.理想情况下,观看设备的任务是符合设备本身的条件来渲染网页.

　　语义标记为设备提供了所需的相关信息,就省去了你自己去考虑所有可能的显示情况（包括现有的或者将来新的设备）.例如,一部手机可以选择使一段标记了标题的文字以粗体显示.而掌上电脑可能会以比较大的字体来显示.无论哪种方式一旦你对文本标记为标题,您就可以确信读取设备将根据其自身的条件来合适地显示页面.

搜索引擎的爬虫也依赖于标记来确定上下文和各个关键字的权重

　　过去你可能还没有考虑搜索引擎的爬虫也是网站的“访客”,但现在它们他们实际上是极其宝贵的用户.没有他们的话,搜索引擎将无法索引你的网站,然后一般用户将很难过来访问.

你的页面是否对爬虫容易理解非常重要,因为爬虫很大程度上会忽略用于表现的标记,而只注重语义标记.

　　因此,如果页面文件的标题被标记,而不是,那么这个页面在搜索结果的位置可能会比较靠后.除了提升易用性外,语义标记有利于正确使用CSS和JavaScript,因为其本身提供了许多“钩钩”来应用页面的样式与行为.

SEO主要还是靠你网站的内容和外部链接的。

便于团队开发和维护

　　W3C给我们定了一个很好的标准，在团队中大家都遵循这个标准，可以减少很多差异化的东西，方便开发和维护，提高开发效率，甚至实现模块化开发。

　　16.谈谈以前端角度出发做好SEO需要考虑什么？

了解搜索引擎如何抓取网页和如何索引网页

　　你需要知道一些搜索引擎的基本工作原理，各个搜索引擎之间的区别，搜索机器人（SE robot 或叫 web crawler）如何进行工作，搜索引擎如何对搜索结果进行排序等等。

Meta标签优化

　　主要包括主题（Title)，网站描述(Description)，和关键词（Keywords）。还有一些其它的隐藏文字比如Author（作者），Category（目录），Language（编码语种）等。

如何选取关键词并在网页中放置关键词

　　搜索就得用关键词。关键词分析和选择是SEO最重要的工作之一。首先要给网站确定主关键词（一般在5个上下），然后针对这些关键词进行优化，包括关键词密度（Density），相关度（Relavancy），突出性（Prominency）等等。

了解主要的搜索引擎

　　虽然搜索引擎有很多，但是对网站流量起决定作用的就那么几个。比如英文的主要有Google，Yahoo，Bing等；中文的有百度，搜狗，有道等。不同的搜索引擎对页面的抓取和索引、排序的规则都不一样。还要了解各搜索门户和搜索引擎之间的关系，比如AOL网页搜索用的是Google的搜索技术，MSN用的是Bing的技术。

主要的互联网目录

　　Open Directory自身不是搜索引擎，而是一个大型的网站目录，他和搜索引擎的主要区别是网站内容的收集方式不同。目录是人工编辑的，主要收录网站主页；搜索引擎是自动收集的，除了主页外还抓取大量的内容页面。

按点击付费的搜索引擎

　　搜索引擎也需要生存，随着互联网商务的越来越成熟，收费的搜索引擎也开始大行其道。最典型的有Overture和百度，当然也包括Google的广告项目Google Adwords。越来越多的人通过搜索引擎的点击广告来定位商业网站，这里面也大有优化和排名的学问，你得学会用最少的广告投入获得最多的点击。

搜索引擎登录

　　网站做完了以后，别躺在那里等着客人从天而降。要让别人找到你，最简单的办法就是将网站提交（submit）到搜索引擎。如果你的是商业网站，主要的搜索引擎和目录都会要求你付费来获得收录（比如Yahoo要299美元），但是好消息是（至少到目前为止）最大的搜索引擎Google目前还是免费，而且它主宰着60％以上的搜索市场。

链接交换和链接广泛度（Link Popularity）

　　网页内容都是以超文本（Hypertext）的方式来互相链接的，网站之间也是如此。除了搜索引擎以外，人们也每天通过不同网站之间的链接来Surfing（“冲浪”）。其它网站到你的网站的链接越多，你也就会获得更多的访问量。更重要的是，你的网站的外部链接数越多，会被搜索引擎认为它的重要性越大，从而给你更高的排名。

合理的标签使用

Css篇：

1.有哪项方式可以对一个DOM设置它的CSS样式？

?外部样式表，引入一个外部css文件

?内部样式表，将css代码放在 <head> 标签内部

?内联样式，将css样式直接定义在 HTML 元素内部

2.CSS都有哪些选择器？

?派生选择器（用HTML标签申明）

?id选择器（用DOM的ID申明）

?类选择器（用一个样式类名申明）

?属性选择器（用DOM的属性申明，属于CSS2，IE6不支持，不常用，不知道就算了）

除了前3种基本选择器，还有一些扩展选择器，包括

?后代选择器（利用空格间隔，比如div .a{ }）

?群组选择器（利用逗号间隔，比如p,div,#a{ }）

那么问题来了，CSS选择器的优先级是怎么样定义的？

基本原则：

一般而言，选择器越特殊，它的优先级越高。也就是选择器指向的越准确，它的优先级就越高。

复杂的计算方法：

?用1表示派生选择器的优先级

?用10表示类选择器的优先级

?用100标示ID选择器的优先级

?div.test1 .span var 优先级 1+10 +10 +1

?span#xxx .songs li 优先级1+100 + 10 + 1

?#xxx li 优先级 100 +1

那么问题来了，看下列代码，<p>标签内的文字是什么颜色的？。

<style>

.classA{ color:blue;}

.classB{ color:red;}

</style>

<body>

<p class='classB classA'> 123 </p>

</body>

答案：red。与样式定义在文件中的先后顺序有关，即是后面的覆盖前面的，与在<p class='classB classA'>中的先后关系无关。

3.CSS中可以通过哪些属性定义，使得一个DOM元素不显示在浏览器可视范围内？

最基本的：

设置display属性为none，或者设置visibility属性为hidden

技巧性：

设置宽高为0，设置透明度为0，设置z-index位置在-1000

4.超链接访问过后hover样式就不出现的问题是什么？如何解决？

答案：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（link,visited,hover,active）

5.什么是Css Hack？ie6,7,8的hack分别是什么？

答案：针对不同的浏览器写不同的CSS code的过程，就是CSS hack。

示例如下：

#test {

width:300px;

height:300px;

background-color:blue; /\*firefox\*/

background-color:red\9; /\*all ie\*/

background-color:yellow\0; /\*ie8\*/

+background-color:pink; /\*ie7\*/

\_background-color:orange; /\*ie6\*/ }

:root #test { background-color:purple\9; } /\*ie9\*/

@media all and (min-width:0px){ #test {background-color:black\0;} } /\*opera\*/

@media screen and (-webkit-min-device-pixel-ratio:0){ #test {background-color:gray;} } /\*chrome and safari\*/

6.请用Css写一个简单的幻灯片效果页面

　　答案：知道是要用css3。使用animation动画实现一个简单的幻灯片效果。

/\*\*HTML\*\*/

div.ani

/\*\*css\*\*/

.ani{

width:480px;

height:320px;

margin:50px auto;

overflow: hidden;

box-shadow:0 0 5px rgba(0,0,0,1);

background-size: cover;

background-position: center;

-webkit-animation-name: "loops";

-webkit-animation-duration: 20s;

-webkit-animation-iteration-count: infinite;

}

@-webkit-keyframes "loops" {

0% {

background:url(http://d.hiphotos.baidu.com/image/w%3D400/sign=c01e6adca964034f0fcdc3069fc27980/e824b899a9014c08e5e38ca4087b02087af4f4d3.jpg) no-repeat;

}

25% {

background:url(http://b.hiphotos.baidu.com/image/w%3D400/sign=edee1572e9f81a4c2632edc9e72b6029/30adcbef76094b364d72bceba1cc7cd98c109dd0.jpg) no-repeat;

}

50% {

background:url(http://b.hiphotos.baidu.com/image/w%3D400/sign=937dace2552c11dfded1be2353266255/d8f9d72a6059252d258e7605369b033b5bb5b912.jpg) no-repeat;

}

75% {

background:url(http://g.hiphotos.baidu.com/image/w%3D400/sign=7d37500b8544ebf86d71653fe9f9d736/0df431adcbef76095d61f0972cdda3cc7cd99e4b.jpg) no-repeat;

}

100% {

background:url(http://c.hiphotos.baidu.com/image/w%3D400/sign=cfb239ceb0fb43161a1f7b7a10a54642/3b87e950352ac65ce2e73f76f9f2b21192138ad1.jpg) no-repeat;

}

}

7.行内元素和块级元素的具体区别是什么？行内元素的padding和margin可设置吗？

块级元素(block)特性：

?总是独占一行，表现为另起一行开始，而且其后的元素也必须另起一行显示;

?宽度(width)、高度(height)、内边距(padding)和外边距(margin)都可控制;

内联元素(inline)特性：

?和相邻的内联元素在同一行;

?宽度(width)、高度(height)、内边距的top/bottom(padding-top/padding-bottom)和外边距的top/bottom(margin-top/margin-bottom)都不可改变（也就是padding和margin的left和right是可以设置的），就是里面文字或图片的大小。

那么问题来了，浏览器还有默认的天生inline-block元素（拥有内在尺寸，可设置高宽，但不会自动换行），有哪些？

答案：<input> 、<img> 、<button> 、<textarea> 、<label>。

8.什么是外边距重叠？重叠的结果是什么？

答案：

外边距重叠就是margin-collapse。

在CSS当中，相邻的两个盒子（可能是兄弟关系也可能是祖先关系）的外边距可以结合成一个单独的外边距。这种合并外边距的方式被称为折叠，并且因而所结合成的外边距称为折叠外边距。

折叠结果遵循下列计算规则：

1.两个相邻的外边距都是正数时，折叠结果是它们两者之间较大的值。

2.两个相邻的外边距都是负数时，折叠结果是两者绝对值的较大值。

3.两个外边距一正一负时，折叠结果是两者的相加的和。

9.rgba()和opacity的透明效果有什么不同？

答案：

rgba()和opacity都能实现透明效果，但最大的不同是opacity作用于元素，以及元素内的所有内容的透明度，

而rgba()只作用于元素的颜色或其背景色。（设置rgba透明的元素的子元素不会继承透明效果！）

10.css中可以让文字在垂直和水平方向上重叠的两个属性是什么？

答案：

垂直方向：line-height

水平方向：letter-spacing

那么问题来了，关于letter-spacing的妙用知道有哪些么？

答案:可以用于消除inline-block元素间的换行符空格间隙问题。

11.如何垂直居中一个浮动元素？

// 方法一：已知元素的高宽

#div1{

background-color:#6699FF;

width:200px;

height:200px;

position: absolute; //父元素需要相对定位

top: 50%;

left: 50%;

margin-top:-100px ; //二分之一的height，width

margin-left: -100px;

}

//方法二:未知元素的高宽

#div1{

width: 200px;

height: 200px;

background-color: #6699FF;

margin:auto;

position: absolute; //父元素需要相对定位

left: 0;

top: 0;

rightright: 0;

bottombottom: 0;

}

那么问题来了，如何垂直居中一个<img>?（用更简便的方法。）

#container //<img>的容器设置如下

{

display:table-cell;

text-align:center;

vertical-align:middle;

}

12.px和em的区别。

　　px和em都是长度单位，区别是，px的值是固定的，指定是多少就是多少，计算比较容易。em得值不是固定的，并且em会继承父级元素的字体大小。

　　浏览器的默认字体高都是16px。所以未经调整的浏览器都符合: 1em=16px。那么12px=0.75em, 10px=0.625em。

13.描述一个"reset"的CSS文件并如何使用它。知道normalize.css吗？你了解他们的不同之处？

　　重置样式非常多，凡是一个前端开发人员肯定有一个常用的重置CSS文件并知道如何使用它们。他们是盲目的在做还是知道为什么这么做呢？原因是不同的浏览器对一些元素有不同的默认样式，如果你不处理，在不同的浏览器下会存在必要的风险，或者更有戏剧性的性发生。

　　你可能会用Normalize来代替你的重置样式文件。它没有重置所有的样式风格，但仅提供了一套合理的默认样式值。既能让众多浏览器达到一致和合理，但又不扰乱其他的东西（如粗体的标题）。

在这一方面，无法做每一个复位重置。它也确实有些超过一个重置，它处理了你永远都不用考虑的怪癖，像HTML的audio元素不一致或line-height不一致。

　　14.Sass、LESS是什么？大家为什么要使用他们？

　　他们是CSS预处理器。他是CSS上的一种抽象层。他们是一种特殊的语法/语言编译成CSS。

　　例如Less是一种动态样式语言. 将CSS赋予了动态语言的特性，如变量，继承，运算， 函数. LESS 既可以在客户端上运行 (支持IE 6+, Webkit, Firefox)，也可一在服务端运行 (借助 Node.js)。

为什么要使用它们？

?结构清晰，便于扩展。

?可以方便地屏蔽浏览器私有语法差异。这个不用多说，封装对浏览器语法差异的重复处理，减少无意义的机械劳动。

?可以轻松实现多重继承。

?完全兼容 CSS 代码，可以方便地应用到老项目中。LESS 只是在 CSS 语法上做了扩展，所以老的 CSS 代码也可以与 LESS 代码一同编译。

15.display:none与visibility:hidden的区别是什么？

?display : 隐藏对应的元素但不挤占该元素原来的空间。

?visibility: 隐藏对应的元素并且挤占该元素原来的空间。

　　即是，使用CSS display:none属性后，HTML元素（对象）的宽度、高度等各种属性值都将“丢失”;而使用visibility:hidden属性后，HTML元素（对象）仅仅是在视觉上看不见（完全透明），而它所占据的空间位置仍然存在。

16.知道css有个content属性吗？有什么作用？有什么应用？

　　答案：

　　知道。css的content属性专门应用在 before/after 伪元素上，用于来插入生成内容。

　　最常见的应用是利用伪类清除浮动。

//一种常见利用伪类清除浮动的代码

.clearfix:after {

content:"."; //这里利用到了content属性

display:block;

height:0;

visibility:hidden;

clear:both; }

.clearfix {

\*zoom:1;

}

after伪元素通过 content 在元素的后面生成了内容为一个点的块级元素，再利用clear:both清除浮动。

　　那么问题继续还有，知道css计数器（序列数字字符自动递增）吗？如何通过css content属性实现css计数器？

　　答案：css计数器是通过设置counter-reset 、counter-increment 两个属性 、及 counter()/counters()一个方法配合after / before 伪类实现。

初级Javascript：

　　1.Javascript是一门什么样的语言，它有哪些特点？

　　没有标准答案。

　　2.Javascript的数据类型都有什么？

　　基本数据类型：String,boolean,Number,Undefined, Null

　　引用数据类型：Object(Array,Date,RegExp,Function)

　　那么问题来了，如何判断某变量是否为数组数据类型？

?方法一.判断其是否具有"数组性质”，如slice()方法。可自己给该变量定义slice方法，故有时会失效

?方法二.obj instanceof Array 在某些IE版本中不正确

?方法三.方法一二皆有漏洞，在ECMA script5中定义了新方法Array.isArray(), 保证其兼容性，最好的方法如下

if(typeof Array.isArray==="undefined")

{

Array.isArray = function(arg){

return Object.prototype.toString.call(arg)==="[object Array]"

};

}

3.已知ID的Input输入框，希望获取这个输入框的输入值，怎么做？(不使用第三方框架)

复制代码 代码如下:

document.getElementById("ID").value

4.希望获取到页面中所有的checkbox怎么做？(不使用第三方框架)

var domList = document.getElementsByTagName(‘input')

var checkBoxList = [];

var len = domList.length;　　//缓存到局部变量

while (len--) {　　//使用while的效率会比for循环更高

　　if (domList[len].type == ‘checkbox') {

　　checkBoxList.push(domList[len]);

　　}

}

5.设置一个已知ID的DIV的html内容为xxxx，字体颜色设置为黑色(不使用第三方框架)

复制代码 代码如下:

var dom = document.getElementById("ID”);

dom.innerHTML = "xxxx”

dom.style.color = "#000”

6.当一个DOM节点被点击时候，我们希望能够执行一个函数，应该怎么做？

? 直接在DOM里绑定事件：<div onclick=”test()”></div>

? 在JS里通过onclick绑定：xxx.onclick = test

? 通过事件添加进行绑定：addEventListener(xxx, ‘click', test)

　　那么问题来了，Javascript的事件流模型都有什么？

?"事件冒泡”：事件开始由最具体的元素接受，然后逐级向上传播

?"事件捕捉”：事件由最不具体的节点先接收，然后逐级向下，一直到最具体的

?"DOM事件流”：三个阶段：事件捕捉，目标阶段，事件冒泡

7.什么是Ajax和JSON，它们的优缺点。

　　Ajax是异步Javascript和XML，用于在Web页面中实现异步数据交互。

　　优点：

?可以使得页面不重载全部内容的情况下加载局部内容，降低数据传输量

?避免用户不断刷新或者跳转页面，提高用户体验

　　缺点：

?对搜索引擎不友好（

?要实现ajax下的前后退功能成本较大

?可能造成请求数的增加

?跨域问题限制

　　JSON是一种轻量级的数据交换格式，ECMA的一个子集

　　优点：轻量级、易于人的阅读和编写，便于机器（Javascript）解析，支持复合数据类型（数组、对象、字符串、数字）

　　8.看下列代码输出为何？解释原因。

复制代码 代码如下:

var a;

alert(typeof a); // undefined

alert(b); // 报错

解释：Undefined是一个只有一个值的数据类型，这个值就是"undefined”，在使用var声明变量但并未对其赋值进行初始化时，这个变量的值就是undefined。而b由于未声明将报错。注意未申明的变量和声明了未赋值的是不一样的。

9.看下列代码,输出什么？解释原因。

复制代码 代码如下:

var a = null;

alert(typeof a); //object

解释：null是一个只有一个值的数据类型，这个值就是null。表示一个空指针对象，所以用typeof检测会返回”object”。

10.看下列代码,输出什么？解释原因。

var undefined;

undefined == null; // true

1 == true; // true

2 == true; // false

0 == false; // true

0 == ''; // true

NaN == NaN; // false

[] == false; // true

[] == ![]; // true

?undefined与null相等，但不恒等（===）

?一个是number一个是string时，会尝试将string转换为number

?尝试将boolean转换为number，0或1

?尝试将Object转换成number或string，取决于另外一个对比量的类型

?所以，对于0、空字符串的判断，建议使用 "===” 。"===”会先判断两边的值类型，类型不匹配时为false。

那么问题来了，看下面的代码，输出什么，foo的类型为什么？

复制代码 代码如下:

var foo = "11"+2-"1";

console.log(foo);

console.log(typeof foo);

执行完后foo的值为111，foo的类型为Number。

复制代码 代码如下:

var foo = "11"+2+"1"; //体会加一个字符串'1' 和 减去一个字符串'1'的不同

console.log(foo);

console.log(typeof foo);　　执行完后foo的值为1121(此处是字符串拼接)，foo的类型为String。

11.看代码给答案。

复制代码 代码如下:

var a = new Object();

a.value = 1;

b = a;

b.value = 2;

alert(a.value);

答案：2（考察引用数据类型细节）

12.已知数组var stringArray = ["This”, "is”, "Baidu”, "Campus”]，Alert出”This is Baidu Campus”。

答案：alert(stringArray.join(" "))

那么问题来了，已知有字符串foo="get-element-by-id",写一个function将其转化成驼峰表示法"getElementById"。

function combo(msg){

var arr = msg.split("-");

var len = arr.length; //将arr.length存储在一个局部变量可以提高for循环效率

for(var i=1;i<len;i++){

arr[i]=arr[i].charAt(0).toUpperCase()+arr[i].substr(1,arr[i].length-1);

}

msg=arr.join("");

return msg;

}

(考察基础API)

13.var numberArray = [3,6,2,4,1,5]; （考察基础API）

1) 实现对该数组的倒排，输出[5,1,4,2,6,3]

2) 实现对该数组的降序排列，输出[6,5,4,3,2,1]

var numberArray = [3,6,2,4,1,5];

numberArray.reverse(); // 5,1,4,2,6,3

numberArray.sort(function(a,b){ //6,5,4,3,2,1

return b-a;

})

14.输出今天的日期，以YYYY-MM-DD的方式，比如今天是2014年9月26日，则输出2014-09-26

var d = new Date();

// 获取年，getFullYear()返回4位的数字

var year = d.getFullYear();

// 获取月，月份比较特殊，0是1月，11是12月

var month = d.getMonth() + 1;

// 变成两位

month = month < 10 ? '0' + month : month;

// 获取日

var day = d.getDate();

day = day < 10 ? '0' + day : day;

alert(year + '-' + month + '-' + day);

15.将字符串”<tr><td>{$id}</td><td>{$name}</td></tr>”中的{$id}替换成10，{$name}替换成Tony （使用正则表达式）

答案："<tr><td>{$id}</td><td>{$id}\_{$name}</td></tr>".replace(/{\$id}/g, '10').replace(/{\$name}/g, ‘Tony');

16.为了保证页面输出安全，我们经常需要对一些特殊的字符进行转义，请写一个函数escapeHtml，将<, >, &, "进行转义

function escapeHtml(str) {

return str.replace(/[<>”&]/g, function(match) {

switch (match) {

case "<”:

return "<”;

case ">”:

return ">”;

case "&”:

return "&”;

case "\””:

return ""”;

}

});

}

17.foo = foo||bar ，这行代码是什么意思？为什么要这样写？

答案：if(!foo) foo = bar; //如果foo存在，值不变，否则把bar的值赋给foo。

短路表达式：作为"&&"和"||"操作符的操作数表达式，这些表达式在进行求值时，只要最终的结果已经可以确定是真或假，求值过程便告终止，这称之为短路求值。

18.看下列代码，将会输出什么?(变量声明提升)

复制代码 代码如下:

var foo = 1;

function(){

console.log(foo);

var foo = 2;

console.log(foo);

}

答案：输出undefined 和 2。上面代码相当于：

var foo = 1;

function(){

var foo;

console.log(foo); //undefined

foo = 2;

console.log(foo); // 2;

}

函数声明与变量声明会被Javascript引擎隐式地提升到当前作用域的顶部，但是只提升名称不会提升赋值部分。

19.用js实现随机选取10--100之间的10个数字，存入一个数组，并排序。

var iArray = [];

funtion getRandom(istart, iend){

var iChoice = istart - iend +1;

return Math.floor(Math.random() \* iChoice + istart;

}

for(var i=0; i<10; i++){

iArray.push(getRandom(10,100));

}

iArray.sort();

20.把两个数组合并，并删除第二个元素。

复制代码 代码如下:

var array1 = ['a','b','c'];

var bArray = ['d','e','f'];

var cArray = array1.concat(bArray);

cArray.splice(1,1);

21.怎样添加、移除、移动、复制、创建和查找节点（原生JS，实在基础，没细写每一步）

　　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() //添加

removeChild() //移除

replaceChild() //替换

insertBefore() //插入

　　3）查找

getElementsByTagName() //通过标签名称

getElementsByName() //通过元素的Name属性的值

getElementById() //通过元素Id，唯一性

22.有这样一个URL：http://item.taobao.com/item.htm?a=1&b=2&c=&d=xxx&e，请写一段JS程序提取URL中的各个GET参数(参数名和参数个数不确定)，将其按key-value形式返回到一个json结构中，如{a:'1', b:'2', c:'', d:'xxx', e:undefined}。

答案：

function serilizeUrl(url) {

var result = {};

url = url.split("?")[1];

var map = url.split("&");

for(var i = 0, len = map.length; i < len; i++) {

result[map[i].split("=")[0]] = map[i].split("=")[1];

}

return result;

}

23.正则表达式构造函数var reg=new RegExp("xxx")与正则表达字面量var reg=//有什么不同？匹配邮箱的正则表达式？

　　答案：当使用RegExp()构造函数的时候，不仅需要转义引号（即\"表示"），并且还需要双反斜杠（即\\表示一个\）。使用正则表达字面量的效率更高。

　　邮箱的正则匹配：

复制代码 代码如下:

var regMail = /^([a-zA-Z0-9\_-])+@([a-zA-Z0-9\_-])+((.[a-zA-Z0-9\_-]{2,3}){1,2})$/;

24.看下面代码，给出输出结果。

复制代码 代码如下:

for(var i=1;i<=3;i++){

setTimeout(function(){

console.log(i);

},0);

};

答案：4 4 4。

　　原因：Javascript事件处理器在线程空闲之前不会运行。那么问题来了，如何让上述代码输出1 2 3？

for(var i=1;i<=3;i++){

setTimeout((function(a){ //改成立即执行函数

console.log(a);

})(i),0);

};

25.写一个function，清除字符串前后的空格。（兼容所有浏览器）

　　使用自带接口trim()，考虑兼容性：

if (!String.prototype.trim) {

String.prototype.trim = function() {

return this.replace(/^\s+/, "").replace(/\s+$/,"");

}

}

// test the function

var str = " \t\n test string ".trim();

alert(str == "test string"); // alerts "true"

26.Javascript中callee和caller的作用？

　　答案：

　　caller是返回一个对函数的引用，该函数调用了当前函数；

　　callee是返回正在被执行的function函数，也就是所指定的function对象的正文。

　　那么问题来了？如果一对兔子每月生一对兔子；一对新生兔，从第二个月起就开始生兔子；假定每对兔子都是一雌一雄，试问一对兔子，第n个月能繁殖成多少对兔子？（使用callee完成）

var result=[];

function fn(n){ //典型的斐波那契数列

if(n==1){

return 1;

}else if(n==2){

return 1;

}else{

if(result[n]){

return result[n];

}else{

//argument.callee()表示fn()

result[n]=arguments.callee(n-1)+arguments.callee(n-2);

return result[n];

}

}

}

中级Javascript：

　　1.实现一个函数clone，可以对Javascript中的5种主要的数据类型（包括Number、String、Object、Array、Boolean）进行值复制

?考察点1：对于基本数据类型和引用数据类型在内存中存放的是值还是指针这一区别是否清楚

?考察点2：是否知道如何判断一个变量是什么类型的

?考察点3：递归算法的设计

// 方法一：

Object.prototype.clone = function(){

var o = this.constructor === Array ? [] : {};

for(var e in this){

o[e] = typeof this[e] === "object" ? this[e].clone() : this[e];

}

return o;

}

//方法二：

/\*\*

\* 克隆一个对象

\* @param Obj

\* @returns

\*/

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{ //普通变量直接赋值

return Obj;

}

}

2.如何消除一个数组里面重复的元素？

var arr=[1,2,3,3,4,4,5,5,6,1,9,3,25,4];

function deRepeat(){

var newArr=[];

var obj={};

var index=0;

var l=arr.length;

for(var i=0;i<l;i++){

if(obj[arr[i]]==undefined)

{

obj[arr[i]]=1;

newArr[index++]=arr[i];

}

else if(obj[arr[i]]==1)

continue;

}

return newArr;

}

var newArr2=deRepeat(arr);

alert(newArr2); //输出1,2,3,4,5,6,9,25

3.小贤是一条可爱的小狗(Dog)，它的叫声很好听(wow)，每次看到主人的时候就会乖乖叫一声(yelp)。从这段描述可以得到以下对象：

function Dog() {

this.wow = function() {

alert('Wow');

}

this.yelp = function() {

this.wow();

}

}

小芒和小贤一样，原来也是一条可爱的小狗，可是突然有一天疯了(MadDog)，一看到人就会每隔半秒叫一声(wow)地不停叫唤(yelp)。请根据描述，按示例的形式用代码来实。（继承，原型，setInterval）

　　答案：

function MadDog() {

this.yelp = function() {

var self = this;

setInterval(function() {

self.wow();

}, 500);

}

}

MadDog.prototype = new Dog();

//for test

var dog = new Dog();

dog.yelp();

var madDog = new MadDog();

madDog.yelp();

4.下面这个ul，如何点击每一列的时候alert其index?（闭包）

复制代码 代码如下:

<ul id=”test”>

<li>这是第一条</li>

<li>这是第二条</li>

<li>这是第三条</li>

</ul>

答案：

// 方法一：

var lis=document.getElementById('2223').getElementsByTagName('li');

for(var i=0;i<3;i++)

{

lis[i].index=i;

lis[i].onclick=function(){

alert(this.index);

};

}

//方法二：

var lis=document.getElementById('2223').getElementsByTagName('li');

for(var i=0;i<3;i++)

{

lis[i].index=i;

lis[i].onclick=(function(a){

return function() {

alert(a);

}

})(i);

}

5.编写一个Javascript函数，输入指定类型的选择器(仅需支持id，class，tagName三种简单CSS选择器，无需兼容组合选择器)可以返回匹配的DOM节点，需考虑浏览器兼容性和性能。

　　/\*\*\* @param selector {String} 传入的CSS选择器。\* @return {Array}\*/

　　答案：（过长，点击打开）

var query = function(selector) {

var reg = /^(#)?(\.)?(\w+)$/img;

var regResult = reg.exec(selector);

var result = [];

//如果是id选择器

if(regResult[1]) {

if(regResult[3]) {

if(typeof document.querySelector === "function") {

result.push(document.querySelector(regResult[3]));

}

else {

result.push(document.getElementById(regResult[3]));

}

}

}

//如果是class选择器

else if(regResult[2]) {

if(regResult[3]) {

if(typeof document.getElementsByClassName === 'function') {

var doms = document.getElementsByClassName(regResult[3]);

if(doms) {

result = converToArray(doms);

}

}

//如果不支持getElementsByClassName函数

else {

var allDoms = document.getElementsByTagName("\*") ;

for(var i = 0, len = allDoms.length; i < len; i++) {

if(allDoms[i].className.search(new RegExp(regResult[2])) > -1) {

result.push(allDoms[i]);

}

}

}

}

}

//如果是标签选择器

else if(regResult[3]) {

var doms = document.getElementsByTagName(regResult[3].toLowerCase());

if(doms) {

result = converToArray(doms);

}

}

return result;

}

function converToArray(nodes){

var array = null;

try{

array = Array.prototype.slice.call(nodes,0);//针对非IE浏览器

}catch(ex){

array = new Array();

for( var i = 0 ,len = nodes.length; i < len ; i++ ) {

array.push(nodes[i])

}

}

return array;

}

6.请评价以下代码并给出改进意见。

if(window.addEventListener){

var addListener = function(el,type,listener,useCapture){

el.addEventListener(type,listener,useCapture);

};

}

else if(document.all){

addListener = function(el,type,listener){

el.attachEvent("on"+type,function(){

listener.apply(el);

});

}

}

评价：

?不应该在if和else语句中声明addListener函数，应该先声明；

?不需要使用window.addEventListener或document.all来进行检测浏览器，应该使用能力检测；

?由于attachEvent在IE中有this指向问题，所以调用它时需要处理一下

　　改进如下：

function addEvent(elem, type, handler){

　　if(elem.addEventListener){

　　　　elem.addEventListener(type, handler, false);

　　}else if(elem.attachEvent){

　　　　elem['temp' + type + handler] = handler;

　　　　elem[type + handler] = function(){

　　　　elem['temp' + type + handler].apply(elem);

　　};

　　elem.attachEvent('on' + type, elem[type + handler]);

}else{

　　elem['on' + type] = handler;

　　}

}

7.给String对象添加一个方法，传入一个string类型的参数，然后将string的每个字符间价格空格返回，例如：

　　addSpace("hello world") // -> 'h e l l o w o r l d'

复制代码 代码如下:

String.prototype.spacify = function(){

return this.split('').join(' ');

};

接着上述答题，那么问题来了

　　1）直接在对象的原型上添加方法是否安全？尤其是在Object对象上。(这个我没能答出？希望知道的说一下。)

　　2）函数声明与函数表达式的区别？

　　答案：在Javscript中，解析器在向执行环境中加载数据时，对函数声明和函数表达式并非是一视同仁的，解析器会率先读取函数声明，并使其在执行任何代码之前可用（可以访问），至于函数表达式，则必须等到解析器执行到它所在的代码行，才会真正被解析执行。（函数声明提升）

　　8.定义一个log方法，让它可以代理console.log的方法。

　　可行的方法一：

复制代码 代码如下:

function log(msg)　{

console.log(msg);

}

log("hello world!") // hello world!

如果要传入多个参数呢？显然上面的方法不能满足要求，所以更好的方法是：

复制代码 代码如下:

function log(){

console.log.apply(console, arguments);

};

那么问题来了，apply和call方法的异同？

答案：

　　对于apply和call两者在作用上是相同的，即是调用一个对象的一个方法，以另一个对象替换当前对象。将一个函数的对象上下文从初始的上下文改变为由 thisObj 指定的新对象。

　　但两者在参数上有区别的。对于第一个参数意义都一样，但对第二个参数： apply传入的是一个参数数组，也就是将多个参数组合成为一个数组传入，而call则作为call的参数传入（从第二个参数开始）。 如 func.call(func1,var1,var2,var3)对应的apply写法为：func.apply(func1,[var1,var2,var3]) 。

　　9.在Javascript中什么是伪数组？如何将伪数组转化为标准数组？

　　答案：

　　伪数组（类数组）：无法直接调用数组方法或期望length属性有什么特殊的行为，但仍可以对真正数组遍历方法来遍历它们。典型的是函数的argument参数，还有像调用getElementsByTagName,document.childNodes之类的,它们都返回NodeList对象都属于伪数组。可以使用Array.prototype.slice.call(fakeArray)将数组转化为真正的Array对象。

　　假设接第八题题干，我们要给每个log方法添加一个"(app)"前缀，比如'hello world!' ->'(app)hello world!'。方法如下：

function log(){

var args = Array.prototype.slice.call(arguments); //为了使用unshift数组方法，将argument转化为真正的数组

args.unshift('(app)');

console.log.apply(console, args);

};

10.对作用域上下文和this的理解，看下列代码：

var User = {

count: 1,

getCount: function() {

return this.count;

}

};

console.log(User.getCount()); // what?

var func = User.getCount;

console.log(func()); // what?

问两处console输出什么？为什么？

　　答案是1和undefined。

　　func是在winodw的上下文中被执行的，所以会访问不到count属性。

　　那么问题来了，如何确保Uesr总是能访问到func的上下文，即正确返回1。

　　答案：正确的方法是使用Function.prototype.bind。兼容各个浏览器完整代码如下：

Function.prototype.bind = Function.prototype.bind || function(context){

var self = this;

return function(){

return self.apply(context, arguments);

};

}

var func = User.getCount.bind(User);

console.log(func());

11.原生JS的window.onload与Jquery的$(document).ready(function(){})有什么不同？如何用原生JS实现Jq的ready方法？

　　window.onload()方法是必须等到页面内包括图片的所有元素加载完毕后才能执行。

　　$(document).ready()是DOM结构绘制完毕后就执行，不必等到加载完毕。

/\*

\* 传递函数给whenReady()

\* 当文档解析完毕且为操作准备就绪时，函数作为document的方法调用

\*/

var whenReady = (function() { //这个函数返回whenReady()函数

var funcs = []; //当获得事件时，要运行的函数

var ready = false; //当触发事件处理程序时,切换为true

//当文档就绪时,调用事件处理程序

function handler(e) {

if(ready) return; //确保事件处理程序只完整运行一次

//如果发生onreadystatechange事件，但其状态不是complete的话,那么文档尚未准备好

if(e.type === 'onreadystatechange' && document.readyState !== 'complete') {

return;

}

//运行所有注册函数

//注意每次都要计算funcs.length

//以防这些函数的调用可能会导致注册更多的函数

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

funcs[i].call(document);

}

//事件处理函数完整执行,切换ready状态, 并移除所有函数

ready = true;

funcs = null;

}

//为接收到的任何事件注册处理程序

if(document.addEventListener) {

document.addEventListener('DOMContentLoaded', handler, false);

document.addEventListener('readystatechange', handler, false); //IE9+

window.addEventListener('load', handler, false);

}else if(document.attachEvent) {

document.attachEvent('onreadystatechange', handler);

window.attachEvent('onload', handler);

}

//返回whenReady()函数

return function whenReady(fn) {

if(ready) { fn.call(document); }

else { funcs.push(fn); }

}

})();

如果上述代码十分难懂，下面这个简化版：

function ready(fn){

if(document.addEventListener) { //标准浏览器

document.addEventListener('DOMContentLoaded', function() {

//注销事件, 避免反复触发

document.removeEventListener('DOMContentLoaded',arguments.callee, false);

fn(); //执行函数

}, false);

}else if(document.attachEvent) { //IE

document.attachEvent('onreadystatechange', function() {

if(document.readyState == 'complete') {

document.detachEvent('onreadystatechange', arguments.callee);

fn(); //函数执行

}

});

}

};

12.（设计题）想实现一个对页面某个节点的拖曳？如何做？（使用原生JS）

　　回答出概念即可，下面是几个要点

1.给需要拖拽的节点绑定mousedown, mousemove, mouseup事件

2.mousedown事件触发后，开始拖拽

3.mousemove时，需要通过event.clientX和clientY获取拖拽位置，并实时更新位置

4.mouseup时，拖拽结束

5.需要注意浏览器边界的情况

13.

function setcookie(name,value,days){ //给cookie增加一个时间变量

　　var exp = new Date();

　　exp.setTime(exp.getTime() + days\*24\*60\*60\*1000); //设置过期时间为days天

　　document.cookie = name + "="+ escape (value) + ";expires=" + exp.toGMTString();

}

function getCookie(name){

　　var result = "";

　　var myCookie = ""+document.cookie+";";

　　var searchName = "+name+"=";

　　var startOfCookie = myCookie.indexOf(searchName);

　　var endOfCookie;

　　if(satrtOfCookie != -1){

　　　　startOfcookie += searchName.length;

　　　　endOfCookie = myCookie.indexOf(";",startOfCookie);

　　　　result = (myCookie.substring(startOfCookie,endOfCookie));

　　}

　　return result;

}

(function(){

　　var oTips = document.getElementById('tips');//假设tips的id为tips

　　var page = {

　　check: function(){//检查tips的cookie是否存在并且允许显示

　　　　var tips = getCookie('tips');

　　　　if(!tips || tips == 'show') return true;//tips的cookie不存在

　　　　if(tips == "never\_show\_again") return false;

　　},

　　hideTip: function(bNever){

　　　　if(bNever) setcookie('tips', 'never\_show\_again', 365);

　　　　oTips.style.display = "none";//隐藏

　　},

　　showTip: function(){

　　oTips.style.display = "inline";//显示，假设tips为行级元素

　　},

　　init: function(){

　　　　var \_this = this;

　　　　if(this.check()){

　　　　\_this.showTip();

　　　　setcookie('tips', 'show', 1);

　　}

　　oTips.onclick = function(){

　　　　\_this.hideTip(true);

　　};

　　}

　　};

page.init();

})();

14.说出以下函数的作用是？空白区域应该填写什么？

//define

(function(window){

function fn(str){

this.str=str;

}

fn.prototype.format = function(){

var arg = \_\_\_\_\_\_;

return this.str.replace(\_\_\_\_\_,function(a,b){

return arg[b]||"";

});

}

window.fn = fn;

})(window);

//use

(function(){

var t = new fn('<p><a href="{0}">{1}</a><span>{2}</span></p>');

console.log(t.format('http://www.alibaba.com','Alibaba','Welcome'));

})();

答案：访函数的作用是使用format函数将函数的参数替换掉{0}这样的内容，返回一个格式化后的结果：

第一个空是：arguments

第二个空是：/\{(\d+)\}/ig

15.用面向对象的Javascript来介绍一下自己。（没答案哦亲，自己试试吧）

答案： 对象或者Json都是不错的选择哦。

16.讲解原生Js实现ajax的原理。

Ajax 的全称是Asynchronous Javascript and XML，其中，Asynchronous 是异步的意思，它有别于传统web开发中采用的同步的方式。

Ajax的原理简单来说通过XmlHttpRequest对象来向服务器发异步请求，从服务器获得数据，然后用javascript来操作DOM而更新页面。

XMLHttpRequest是ajax的核心机制，它是在IE5中首先引入的，是一种支持异步请求的技术。简单的说，也就是javascript可以及时向服务器提出请求和处理响应，而不阻塞用户。达到无刷新的效果。

XMLHttpRequest这个对象的属性有：

? onreadystatechang 每次状态改变所触发事件的事件处理程序。

? responseText 从服务器进程返回数据的字符串形式。

? responseXML 从服务器进程返回的DOM兼容的文档数据对象。

? status 从服务器返回的数字代码，比如常见的404（未找到）和200（已就绪）

? status Text 伴随状态码的字符串信息

? readyState 对象状态值

? 0 (未初始化) 对象已建立，但是尚未初始化（尚未调用open方法）

? 1 (初始化) 对象已建立，尚未调用send方法

? 2 (发送数据) send方法已调用，但是当前的状态及http头未知

? 3 (数据传送中) 已接收部分数据，因为响应及http头不全，这时通过responseBody和responseText获取部分数据会出现错误，

? 4 (完成) 数据接收完毕,此时可以通过通过responseXml和responseText获取完整的回应数据

下面简单封装一个函数：（略长，点击打开）

ajax({

url: "./TestXHR.aspx", //请求地址

type: "POST", //请求方式

data: { name: "super", age: 20 }, //请求参数

dataType: "json",

success: function (response, xml) {

// 此处放成功后执行的代码

},

fail: function (status) {

// 此处放失败后执行的代码

}

});

function ajax(options) {

options = options || {};

options.type = (options.type || "GET").toUpperCase();

options.dataType = options.dataType || "json";

var params = formatParams(options.data);

//创建 - 非IE6 - 第一步

if (window.XMLHttpRequest) {

var xhr = new XMLHttpRequest();

} else { //IE6及其以下版本浏览器

var xhr = new ActiveXObject('Microsoft.XMLHTTP');

}

//接收 - 第三步

xhr.onreadystatechange = function () {

if (xhr.readyState == 4) {

var status = xhr.status;

if (status >= 200 && status < 300) {

options.success && options.success(xhr.responseText, xhr.responseXML);

} else {

options.fail && options.fail(status);

}

}

}

//连接 和 发送 - 第二步

if (options.type == "GET") {

xhr.open("GET", options.url + "?" + params, true);

xhr.send(null);

} else if (options.type == "POST") {

xhr.open("POST", options.url, true);

//设置表单提交时的内容类型

xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");

xhr.send(params);

}

}

//格式化参数

function formatParams(data) {

var arr = [];

for (var name in data) {

arr.push(encodeURIComponent(name) + "=" + encodeURIComponent(data[name]));

}

arr.push(("v=" + Math.random()).replace("."));

return arr.join("&");

}

上述代码大致表述了ajax的过程，释义自行google，问题未完，那么知道什么是Jsonp和pjax吗？

　　答案：

　　Jsonp：(JSON with Padding)是一种跨域请求方式。主要原理是利用了script 标签可以跨域请求的特点，由其 src 属性发送请求到服务器，服务器返回 js 代码，网页端接受响应，然后就直接执行了，这和通过 script 标签引用外部文件的原理是一样的。JSONP由两部分组成：回调函数和数据，回调函数一般是由网页端控制，作为参数发往服务器端，服务器端把该函数和数据拼成字符串返回。

　　pjax：pjax是一种基于ajax+history.pushState的新技术，该技术可以无刷新改变页面的内容，并且可以改变页面的URL。（关键点：可以实现ajax无法实现的后退功能）pjax是ajax+pushState的封装，同时支持本地存储、动画等多种功能。目前支持jquery、qwrap、kissy等多种版本。

1.1请描述一个网页从开始请求到最终显示的完整过程？

一般可分为7个步骤：

1.在浏览器中输入网址；

2.发送至DNS服务器并获得域名对应的WEB服务器的IP地址；

3.与WEB服务器建立TCP连接；

4.浏览器向WEB服务器的IP地址发送相应的HTTP请求；

5.WEB服务器响应请求并返回指定的URL的数据，或错误信息，如果设定重定向，则重定向到新的URL地址。

6.浏览器下载数据后解析HTML源文件，解析的过程中实现对页面的排版，解析完成后在浏览器中显示基础页面。

7.分析页面中的超链接并显示在当前页面，重复以上过程直至无超链接需要发送，完成全部显示。

1.2 DOCTYPE声明的作用是什么？严格模式与混杂模式如何区分？

HTML语言已经存在太久了，目前必然会有一些不同版本的文档存在。为了能够让浏览器清楚你的文档的版本、类型和风格，需要在文档的起始用DOCTYPE声明指定当前文档的版本和风格。如果在网页中提供了版本信息，则可以有利于验证页面中的代码是否符合当前的版本和风格。

<！DOCTYPE>声明位于文档中的最前面，处于标签之前，告知浏览器的解析器，用什么文档类型规范来解析这个文档。

在严格模式（标准模式）中，浏览器根据规范呈现页面；在混杂模式中，页面以向后兼容的方式显示，以防止老站点无法工作。

如果HTML文档包含形式完整的DOCTYPE，那么它一般以标准模式呈现。对于HTML4.01文档，包含严格DTD的DOCTYPE常常导致页面以标准模式呈现。DOCTYPE不存在或格式不正确会导致文档以混杂模式呈现。

1.3简要描述常见的浏览器内核。

浏览器内核赋值对网页语法的解释并显示网页，它决定了浏览器如何显示网页的内容以及页面的格式信息。

常见的浏览器内核：

Trident：IE浏览器

Gecko：Mozilla浏览器，如Firefox

Webkit：Safari浏览器，也是Chrome浏览器的内核

Blink：Chrome浏览器，Opera浏览器

1.4 如何理解html标签语义化？

语义化的主要目的在于，直观的认识标签(markup)和属性(attribute)的用途和作用。

可以概括为：用正确的标签做正确的事情。

Html语义化可以让页面的内容结构化，便于浏览器解析，便于搜索引擎解析，并提高带帽的可维护度和可重用性。

比如，尽可能少的使用无语义的标签div，使用结构化标签<header>、<section>、<footer>。

1.5 锚点的作用是什么？如何创建锚点？

锚点是文档中某行的一个记号，类似于书签，用于链接到文档中的某个位置。当定义锚点后，我们可以创建直接跳至该锚点（比如页面中某个小节）的链接，这样使用者就无需不停地滚动页面来寻找他们需要的信息了。

在使用<a>元素创建锚点时，需要使用name属性为其命名，代码如下所示：

<a name=”anchorname1”>锚点一</a>

然后就可以创建链接，直接跳转到锚点，代码如下所示：

<a href=”#anchorname1”>回到锚点一</a>

1.6 使用<label>元素显示文本与使用其他文本标记显示文本有什么不同？

<label>元素的直观效果是直接显示标记之间的文本，而且不会为文本呈现任何特殊效果。但是，它和其他文本标记所不同的是，它为鼠标用户改进了用户体验性。

这是因为，<label>元素可以附带一个for属性，只要将该属性的值设置为表单中任何一个控件的id属性的值，则当该用户点击标签（文本）时，浏览器就会自动将焦点转到和标签相关的表单控件上。即：如果在<label>元素内点击文本，就会触发此控件。

1.7 列举常用的结构标记，并描述其作用

结构标签专门用于标识页面的不同结构，相对于使用<div>元素而言，可以实现语义化的标签。

常用的结构标签有：

<header>元素：用于定义文档的页眉；

<nav>元素：用于定义页面的导航链接部分；

<section>元素：用于定义文档中的节，表示文档中的一个具体的组成部分；

<article>元素：常用于定义独立于文档的其他部分的内容

<footer>元素：常用于定义某区域的脚注信息；

<aside>元素：常用于定义页面的一些额外组成部分，如广告栏、侧边栏和相关引用信息等。

1.8 超级链接有哪些常见的表现形式？

<a>元素用于创建超级链接，常见的表现形式有：

普通超级链接。语法：<a href=”” target=””>文本</a>

下载链接，即目标文档为下载资源。语法：<a href=”day.zip”>下载</a>

电子邮件链接，用于链接到Email。语法：<a href=”mailo:1234546@qq.com”>

联系我们</a>

空链接，用于返回页面顶部。语法：<a href=”#”>...</a>

链接到JavaScript，以实现特定的代码功能。语法：<a href=”javascript:...”>...</a>

1.9 简要描述行内元素和块级元素的区别？

块级元素的前后都会自动换行，如图存在换行符一样。默认情况下，块级元素会独占一行。例如<p>、<hn>、<div>都是块级元素。显示这些元素中间的文本时，都将从新行中开始显示，其后内容也将在新行中显示。

行内元素可以和企业行内元素位于同一行，在浏览器中显示是不会换行。例如<a><span>等。对于行内元素，不能设置其高度和宽度。

还有一种元素，行内块级元素，比如<img><input>元素等。这些元素可以和其他行内元素位于同一行，同时可以设置其高度和宽度。

1.10 表单向服务器提交数据有几种方式？这些方式有什么区别？

将表单数据发送给服务器的常用方式有两种：Get和Post。

浏览器发送给服务器的HTTP请求分为请求头（header）和请求主体（body）两部分。其中，必须包含头部分，用于指定发送请求的方式、目的以及其他关键信息；而主体是可选的。在头数据和主体数据之间用一个空白行来隔开。

比如，需要发送请求到页面getstockprice.PHP，而且需要附带数据Symbol=MSFT。那么如果使用Get方式发送数据，简化后的请求数据内容如下所示：

GET/Trading/GetStockPrice.aspx?Symbol=MSFT HTTP/1.1

Host:localhost

如果使用Post方式发送数据，则简化后的请求数据内容如下所示：

GET/Trading/GetStockPrice.aspx HTTP/1.1

Host:localhost

Content=Type:application/x-www-form-urlencoded 表示请求的MIME类型

Content-Length:11 表示请求数据的长度（大小）

Symbol=MSFT

由此可见，两种方式的区别主要在于发送数据方式不同。

使用Get方式向服务器发送表单数据时，表单数据附加在URL属性的末端；才有POST方法发送数据时，数据都会放在在主体中发送。

## [浅谈HTTP中Get与Post的区别](http://www.cnblogs.com/hyddd/archive/2009/03/31/1426026.html)

　　Http定义了与服务器交互的不同方法，最基本的方法有4种，分别是GET，POST，PUT，DELETE。URL全称是资源描述符，我们可以这样认为：一个URL地址，它用于描述一个网络上的资源，而HTTP中的GET，POST，PUT，DELETE就对应着对这个资源的查，改，增，删4个操作。到这里，大家应该有个大概的了解了，GET一般用于获取/查询资源信息，而POST一般用于更新资源信息。

**1**.根据HTTP规范，GET用于信息获取，而且应该是安全的和幂等的。

　　(1).所谓安全的意味着该操作用于获取信息而非修改信息。换句话说，GET 请求一般不应产生副作用。就是说，它仅仅是获取资源信息，就像数据库查询一样，不会修改，增加数据，不会影响资源的状态。

　　\* 注意：这里安全的含义仅仅是指是非修改信息。

　　(2).幂等的意味着对同一URL的多个请求应该返回同样的结果。这里我再解释一下**幂等**这个概念：

[IMG_256](http://www.cnblogs.com/hyddd/archive/2009/03/31/javascript:void(0);)

**幂等**（idempotent、idempotence）是一个数学或计算机学概念，常见于抽象代数中。  
　　幂等有一下几种定义：  
　　对于单目运算，如果一个运算对于在范围内的所有的一个数多次进行该运算所得的结果和进行一次该运算所得的结果是一样的，那么我们就称该运算是幂等的。比如绝对值运算就是一个例子，在实数集中，有abs(a)=abs(abs(a))。  
　　对于双目运算，则要求当参与运算的两个值是等值的情况下，如果满足运算结果与参与运算的两个值相等，则称该运算幂等，如求两个数的最大值的函数，有在在实数集中幂等，即max(x,x) = x。

[IMG_257](http://www.cnblogs.com/hyddd/archive/2009/03/31/javascript:void(0);)

看完上述解释后，应该可以理解GET幂等的含义了。

　　但在实际应用中，以上2条规定并没有这么严格。引用别人文章的例子：比如，新闻站点的头版不断更新。虽然第二次请求会返回不同的一批新闻，该操作仍然被认为是安全的和幂等的，因为它总是返回当前的新闻。从根本上说，如果目标是当用户打开一个链接时，他可以确信从自身的角度来看没有改变资源即可。

**2**.根据HTTP规范，POST表示可能修改变服务器上的资源的请求。继续引用上面的例子：还是新闻以网站为例，读者对新闻发表自己的评论应该通过POST实现，因为在评论提交后站点的资源已经不同了，或者说资源被修改了。

　　上面大概说了一下HTTP规范中GET和POST的一些原理性的问题。但在实际的做的时候，很多人却没有按照HTTP规范去做，导致这个问题的原因有很多，比如说：

**1**.很多人贪方便，更新资源时用了GET，因为用POST必须要到FORM（表单），这样会麻烦一点。

**2**.对资源的增，删，改，查操作，其实都可以通过GET/POST完成，不需要用到PUT和DELETE。

**3**.另外一个是，早期的Web MVC框架设计者们并没有有意识地将URL当作抽象的资源来看待和设计，所以导致一个比较严重的问题是传统的Web MVC框架基本上都只支持GET和POST两种HTTP方法，而不支持PUT和DELETE方法。

 　　\* 简单解释一下MVC：MVC本来是存在于Desktop程序中的，M是指数据模型，V是指用户界面，C则是控制器。使用MVC的目的是将M和V的实现代码分离，从而使同一个程序可以使用不同的表现形式。

　　以上3点典型地描述了老一套的风格（没有严格遵守HTTP规范），随着架构的发展，现在出现REST(Representational State Transfer)，一套支持HTTP规范的新风格，这里不多说了，可以参考《RESTful Web Services》。

　　说完原理性的问题，我们再从表面现像上面看看GET和POST的区别：

**1**.GET请求的数据会附在URL之后（就是把数据放置在HTTP协议头中），以?分割URL和传输数据，参数之间以&相连，如：login.action?name=hyddd&password=idontknow&verify=%E4%BD%A0%E5%A5%BD。如果数据是英文字母/数字，原样发送，如果是空格，转换为+，如果是中文/其他字符，则直接把字符串用BASE64加密，得出如：%E4%BD%A0%E5%A5%BD，其中％XX中的XX为该符号以16进制表示的ASCII。

　　POST把提交的数据则放置在是HTTP包的包体中。

**2**."GET方式提交的数据最多只能是1024字节，理论上POST没有限制，可传较大量的数据，IIS4中最大为80KB，IIS5中为100KB"？？！

　　以上这句是我从其他文章转过来的，其实这样说是错误的，不准确的：

　　(1).首先是"GET方式提交的数据最多只能是1024字节"，因为GET是通过URL提交数据，那么GET可提交的数据量就跟URL的长度有直接关系了。而实际上，URL不存在参数上限的问题，HTTP协议规范没有对URL长度进行限制。这个限制是特定的浏览器及服务器对它的限制。IE对URL长度的限制是2083字节(2K+35)。对于其他浏览器，如Netscape、FireFox等，理论上没有长度限制，其限制取决于操作系统的支持。

　　注意这是限制是整个URL长度，而不仅仅是你的参数值数据长度。[见参考资料5]

　　(2).理论上讲，POST是没有大小限制的，HTTP协议规范也没有进行大小限制，说“POST数据量存在80K/100K的大小限制”是不准确的，POST数据是没有限制的，起限制作用的是服务器的处理程序的处理能力。

　　对于ASP程序，Request对象处理每个表单域时存在100K的数据长度限制。但如果使用Request.BinaryRead则没有这个限制。

　　由这个延伸出去，对于IIS 6.0，微软出于安全考虑，加大了限制。我们还需要注意：

　　　　 1).IIS 6.0默认ASP POST数据量最大为200KB，每个表单域限制是100KB。  
　　　　 2).IIS 6.0默认上传文件的最大大小是4MB。  
　　　　 3).IIS 6.0默认最大请求头是16KB。  
　　IIS 6.0之前没有这些限制。[见参考资料5]

　　所以上面的80K，100K可能只是默认值而已(注：关于IIS4和IIS5的参数，我还没有确认)，但肯定是可以自己设置的。由于每个版本的IIS对这些参数的默认值都不一样，具体请参考相关的IIS配置文档。

**3**.在ASP中，服务端获取GET请求参数用Request.QueryString，获取POST请求参数用Request.Form。在JSP中，用request.getParameter(\"XXXX\")来获取，虽然jsp中也有request.getQueryString()方法，但使用起来比较麻烦，比如：传一个test.jsp?name=hyddd&password=hyddd，用request.getQueryString()得到的是：name=hyddd&password=hyddd。在PHP中，可以用$\_GET和$\_POST分别获取GET和POST中的数据，而$\_REQUEST则可以获取GET和POST两种请求中的数据。值得注意的是，JSP中使用request和PHP中使用$\_REQUEST都会有隐患，这个下次再写个文章总结。

**4**.POST的安全性要比GET的安全性高。注意：这里所说的安全性和上面GET提到的“安全”不是同个概念。上面“安全”的含义仅仅是不作数据修改，而这里安全的含义是真正的Security的含义，比如：通过GET提交数据，用户名和密码将明文出现在URL上，因为(1)登录页面有可能被浏览器缓存，(2)其他人查看浏览器的历史纪录，那么别人就可以拿到你的账号和密码了，除此之外，使用GET提交数据还可能会造成Cross-site request forgery攻击。

总结一下，Get是向服务器发索取数据的一种请求，而Post是向服务器提交数据的一种请求，在FORM（表单）中，Method默认为"GET"，实质上，GET和POST只是发送机制不同，并不是一个取一个发！

在清除浮动前我们要了解两个重要的定义：

****浮动的定义****：使元素脱离文档流，按照指定方向发生移动，遇到父级边界或者相邻的浮动元素停了下来。

****高度塌陷****：浮动元素父元素高度自适应（父元素不写高度时，子元素写了浮动后，父元素会发生高度塌陷）

知道浮动和为什么要清除浮动之后我们可以开始学习如何清除浮动了，这时候我们就需要用到清除浮动的属性****clear****,   
****clear:left | right | both | none | inherit****：元素的某个方向上不能有浮动元素   
****clear:both****：在左右两侧均不允许浮动元素。

具体清楚浮动的方法主要一下几种：

1、clear清除浮动（添加空div法）

在浮动元素下方添加空div,并给该元素写css样式： {clear:both;height:0;overflow:hidden;}  
2、方法：给浮动元素父级****设置高度****

我们知道了高度塌陷是应为给浮动元素的父级高度是自适应导致的，那么我们给它的设置适当的高度就可以解决这个问题了。

缺点：在浮动元素高度不确定的时候不适用  
  
3、方法：以浮制浮（****父级同时浮动****）

何谓“以浮制浮”呢？就是\*\*让浮动元素的父级也浮动\*\*。

缺点：需要给每个浮动元素父级添加浮动，浮动多了容易出现问题。  
4、方法：****父级设置成inline-block****

缺点：父级的margin左右auto失效，无法使用margin: 0 auto;居中了  
5、 ****br**** 清浮动

<div class="box">

<div class="top"></div>

<br clear="both" />

</div>

br 标签自带clear属性，将它设置成both其实和添加空div原理是一样的。   
问题：不符合工作中：结构、样式、行为，三者分离的要求。  
6、给父级添加****overflow:hidden**** 清浮动方法；

问题：需要配合 宽度 或者 zoom 兼容IE6 IE7；

overflow: hidden;

\*zoom: 1;  
7、万能清除法 ****after伪类**** 清浮动（现在****主流****方法，****推荐使用****）

选择符:after{

content:".";

clear:both;

display:block;

height:0;

overflow:hidden;

visibility:hidden; }

同时为了兼容 IE6，7 同样需要配合zoom使用例如：

.clear:after{content:'';display:block;clear:both;height:0;overflow:hidden;visibility:hidden;}.clear{zoom:1;}

需要注意的东西：

after伪类： 元素内部末尾添加内容；

:after{content"添加的内容";} IE6，7下不兼容

zoom 缩放

a、触发 IE下 haslayout，使元素根据自身内容计算宽高。

b、FF 不支持；