

HTML

Doctype 作用？严格模式与混杂模式如何区分？它们有何意义？

- (1)、`<!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。
- (2)、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。
- (3)、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？

- (1) 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。
- (2) 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
- (3) 知名的空元素：`
` `<hr>` `` `<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？

- (1) link 属于 XHTML 标签，而@import 是 CSS 提供的;
- (2) 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;
- (3) import 只在 IE5 以上才能识别，而 link 是 XHTML 标签，无兼容问题;
- (4) 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、火狐浏览器的内核 Gecko、Chrome 的浏览器内核：WebKit 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;}**

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** 浏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 {}**

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

语义化更好的内容元素，比如 article、footer、header、nav、section

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

新的技术 webworker, 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；

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风格：

* 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

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 代码吧。

(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)

```
< 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 在浏览器和服务端间来回传递。 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？

调用 localstorage、cookies 等本地存储方式

websocket 如何兼容低浏览器？

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

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 padding;

(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？如何居中一个浮动元素？

给 div 设置一个宽度，然后添加 margin:0 auto 属性

```
div{  
    width:200px;  
    margin:0 auto;  
}
```

居中一个浮动元素

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。 position 的值， relative 和 absolute 定位原点是什么？

1.

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。

2.

***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 (老 IE 不支持)**

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

你怎么来实现页面设计图，你认为前端应该如何高质量完成工作？一个满屏 品 字布局 如何设计？

- * 首先划分成头部、body、脚部；。。。。。
- 实现效果图是最基本的工作，精确到 2px；
- 与设计师，产品经理的沟通和项目的参与
- 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 HTML5。

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？

- * 使用率较高的框架有 jQuery、YUI、Prototype、Dojo、Ext.js、Mootools 等。尤其是 jQuery，超过 91%。
- 轻量级框架有 Modernizr、underscore.js、backbone.js、Raphael.js 等。
- (理解这些框架的功能、性能、设计原理)
- * Sublime Text 、Eclipse、Notepad、Firebug、HttpWatch、Yslow。

- * 城市选择插件，汽车型号选择插件、幻灯片插件。弹出层。(写过开源程序，加载器，js 引擎更好)

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：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：

```
/*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

浮动元素脱离文档流，不占据空间。浮动元素碰到包含它的边框或者浮动元素的边框停留。

技巧:

1.使用空标签清除浮动。

这种方法是在所有浮动标签后面添加一个空标签 定义 css clear:both. 弊端就是增加了无意义标签。

2.使用 overflow。

给包含浮动元素的父标签添加 css 属性 overflow:auto; zoom:1; zoom:1 用于兼容 IE6。

3.使用 after 伪对象清除浮动。

该方法只适用于非 IE 浏览器。具体写法可参照以下示例。使用中需注意以下几点。一、该方法中必须为需要清除浮动元素的伪对象中设置 height:0，否则该元素会比实际高出若干像素；

4.在 html 页面添加：<div class=' clear' ></div>

用过媒体查询，针对移动端的布局吗？

用过 @media screen and (min-width:480px){ }

使用 CSS 预处理器吗？喜欢那个？

三款 CSS 预处理器框架，分别是 Sass、Less CSS、Stylus。 喜欢哪个？Sass

什么是 CSS 预处理器？

CSS 预处理器是一种语言用来为 CSS 增加一些编程的特性，无需考虑浏览器的兼容性问题，例如你可以在 CSS 中使用变量、简单的程序逻辑、函数等等在编程语言中的一些基本技巧，可以让你的 CSS 更见简洁，适应性更强，代码更直观等诸多好处。

title 和 alt 区别？

alt 属性的功能是不能显示图像信息时候的替换文字(即在图片显示的位置上显示文字，说明这个图片的要显示的内容)。Alt 属性值的长度必须少于100个英文字符。

title 属性可以用在除了 **base**, **basefont**, **head**, **html**, **meta**, **param**, **script** 和 **title** 之外的所有标签。

title 属性的功能是提示。额外的说明信息和非本质的信息请使用 **title** 属性。**title** 属性值可以比 **alt** 属性值设置的更长。

title 属性有一个很好的用途，即为链接添加描述性文字，特别是当连接本身并不是十分清楚的表达了链接的目的。这样就使得访问者知道那些链接将会带他们到什么地方，他们就不会加载一个可能完全不感兴趣的页面。另外一个潜在的应用就是为图像提供额外的说明信息，比如日期或者其他非本质的信息。

解释 css sprites, 如何使用。

CSS Sprite 又叫 CSS 精灵，是目前大型网站中经常运用的图片处理方式。它的原理很简单，将网站上零散的小图片（或图标）整合在一张大图上，再用 CSS 中 “background-image” 属性来定位图片的 X/Y 轴位置，从而减轻服务器对图片的请求数量，提高网页加载速度。因为对于当前大多数网速而言，不高于 200KB 的单张图片所需载入时间基本是差不多的，如果页面上有 20 张小图片或图标，那么服务器会载入 20 次。但使用 CSS Sprite 将图片整合成一张大图后，服务器只需要载入一次就够了，很大程度上优化了图片加载速度，这在门户网站尤为明显。

CSS Sprites 优点

- 1.利用 CSS Sprites 能很好地减少了网页的 http 请求，从而大大的提高了页面的性能，这也是 CSS Sprites 最大的优点，也是其被广泛传播和应用的主要原因；
- 2.CSS Sprites 能减少图片的字节，曾经比较过多次 3 张图片合并成 1 张图片的字节总是小于这 3 张图片的字节总和。解决了网页设计师在图片命名上的困扰，只需对一张集合的图片上命名就可以了，不需要对每一个小元素进行命名，从而提高了网页的制作效率。
- 3.更换风格方便，只需要在一张或少张图片上修改图片的颜色或样式，整个网页的风格就可以改变。维护起来更加方便。

CSS Sprites 缺点

- 1.在图片合并的时候，你要把多张图片有序的合理的合并成一张图片，还要留好足够的空间，防止板块内不会出现不必要的背景；这些还好，最痛苦的是在宽屏，高分辨率的屏幕下的自适应页面，你的图片如果不够宽，很容易出现背景断裂；
- 2.CSS Sprites 在开发的时候比较麻烦，你要通过 photoshop 或其他工具测量计算每一个背景单元的精确位置，这是针线活，没什么难度，但是很繁琐；幸好腾讯的鬼哥用 RIA 开发了一个 CSS Sprites 样式生成工具，虽然还有一些使用上的不灵活，但是已经比 photoshop 测量来的方便多了，而且样式直接生成，复制，拷贝就 OK！
- 3.CSS Sprites 在维护的时候比较麻烦，如果页面背景有少许改动，一般就要改这张合并的图片，无需改的地方最好不要动，这样避免改动更多的 css，如果在原来的地方放不下，又只能（最好）往下加图片，这样图片的字节就增加了，还要改动 css。

22.在书写高效 css 文件时会有哪些问题需要考虑

一：css 命名、书写规范；（好的代码看上去就很整齐 很有条理性这样方便日后的维护和管理）

二：css 优先级； 选择器权重：内联样式：1000，id 选择器：1000， class 选择器：10，标签选择器：1 注意权重的问题。

三：少用滤镜，少用 hack，少用 position: absolute；。

四：多用继承属性。

五：使用简写样式：例如 margin: 10px； 浏览器会解释为上下左右各有 10px 的外补丁。margin: 0 10px 浏览器解析为 左右有 10px 的外补丁。

六：不要在 ID 选择器和 class 选择器前 使用标签名 例如：div.box { color: #f00; }; 直接 可以用类名， .box { color:#f00;} 这样浏览器找到这个 class 后 就不用再匹配是否存在 div 标签.从而提高了渲染效率。当然同一级的 有不同的样式可以这样写，但是不建议这样。

七：css 的层级关系不要太深 用 class 直接代替多余的层级元素。

JavaScript

JavaScript 原型，原型链？有什么特点？

* 原型对象也是普通的对象，是对象一个自带隐式的 __proto__ 属性，原型也有可能有自己的原型，如果一个原型对象的原型不为 null 的话，我们就称之为原型链。

* 原型链是由一些用来继承和共享属性的对象组成的（有限的）对象链。

* JavaScript 的数据对象有那些属性值？

writable：这个属性的值是否可以改。

configurable：这个属性的配置是否可以删除，修改。

enumerable：这个属性是否能在 for...in 循环中遍历出来或在 Object.keys 中列举出来。

value：属性值。

* 当我们需要一个属性的时，Javascript 引擎会先看当前对象中是否有这个属性， 如果没有的话，就会查找他的 Prototype 对象是否有这个属性。

```
function clone(proto) {  
    function Dummy() {}  
    Dummy.prototype = proto;  
    Dummy.prototype.constructor = Dummy;  
    return new Dummy(); //等价于 Object.create(Person);  
}  
  
function object(old) {  
    function F() {}  
    F.prototype = old;  
    return new F();  
}  
  
var newObj = object(oldObject);
```

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。

```
// 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。

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) 但 map 传了 3 个 (element, index, array)

如何创建一个对象？（画出此对象的内存图）

```

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。

```
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
```

1. 如何判断一个对象是否属于某个类？

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** 。

```
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？

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？

Ajax 是什么：AJAX 即“Asynchronous Javascript And XML”（异步 JavaScript 和 XML），是指一种创建交互式网页应用的网页开发技术。

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?

待完善

1. 通过异步模式，提升了用户体验
2. 优化了浏览器和服务器之间的传输，减少不必要的的数据往返，减少了带宽占用
3. Ajax 在客户端运行，承担了一部分本来由服务器承担的工作，减少了大用户量下的服务器负载。
2. Ajax 的最大的特点是什么。

Ajax 可以实现动态不刷新（局部刷新）

readyState 属性 状态 有 5 个可取值：0=未初始化，1=正在加载，2=以加载，3=交互中，4=完成

ajax 的缺点

- 1、ajax 不支持浏览器 back 按钮。
- 2、安全问题 AJAX 暴露了与服务器交互的细节。
- 3、对搜索引擎的支持比较弱。
- 4、破坏了程序的异常机制。
- 5、不容易调试。

跨域：jsonp、iframe、window.name、window.postMessage、服务器上设置代理页面

同步和异步的区别？

同步是指： 发送方发出数据后，等接收方发回响应以后才发下一个数据包的通讯方式。

异步是指：发送方发出数据后，不等接收方发回响应，接着发送下个数据包的通讯方式。

如何解决跨域问题？

jsonp、iframe、**window.name**、**window.postMessage**、服务器上设置代理页面

模块化怎么做？

立即执行函数,不暴露私有成员

```
var module1 = (function(){
    var _count = 0;
    var m1 = function(){
        //...
    };
    var m2 = function(){
        //...
    };
    return {
        m1 : m1,
        m2 : m2
    };
});
```



```
})();
```

异步加载的方式有哪些？

(1) defer，只支持 IE

(2) async：

(3) 创建 script，插入到 DOM 中，加载完毕后 callBack

document.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

.call() 和 .apply() 的区别？

例子中用 add 来替换 **sub**，add.call(**sub**,3,1) == add(3,1)，所以运行结果为：alert(4);

注意：js 中的函数其实是对象，函数名是对 Function 对象的引用。

```
function add(a,b)
```

```
{
```

```
    alert(a+b);
```

```
}
```

```
function sub(a,b)
```

```
{
```

```
    alert(a-b);
```

```
}
```

```
add.call(sub,3,1);
```

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 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：

```
$.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++) {}

那些操作会造成内存泄漏？

内存泄漏指任何对象在您不再拥有或需要它之后仍然存在。

垃圾回收器定期扫描对象，并计算引用了每个对象的其他对象的数量。如果一个对象的引用数量为 0(没有其他对象引用过该对象)，或对该对象的惟一引用是循环的，那么该对象的内存即可回收。

setTimeout 的第一个参数使用字符串而非函数的话，会引发内存泄漏。

闭包、控制台日志、循环（在两个对象彼此引用且彼此保留时，就会产生一个循环）

如何判断当前脚本运行在浏览器还是 node 环境中？

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

请写出一个获取 5 位数随机数的方法，要求最大数字不超过 99999

```
function GetRandomNum(Min,Max)

{
    var Range = Max - Min;
    var Rand = Math.random();
    return(Min + Math.round(Rand * Range));
}

var num = GetRandomNum(1,99999);
```

alert(num);

写一个时钟：格式为 **hh mm ss**

```
<!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=utf-8" />
<title>简易网页时钟</title>
<style>
body,div{margin:0;padding:0;}
body{color:#fff;font: 16px/1.5 \5fae\8f6f\96c5\9ed1;}
#clock{width:300px;text-align:center;background:#1a1a1a;margin:10px auto;padding:20px 0;}
span{color:#000;width:80px;line-height:2;background:#fbfbfb;border:2px solid #b4b4b4;margin:0 10px;padding:0 10px;}
</style>
<script>
window.onload = function ()
{
    var oClock = document.getElementById("clock");
    var aSpan = oClock.getElementsByTagName("span");

    setInterval(getTimes, 1000);
    getTimes();
    function getTimes ()
    {
        var oDate = new Date();
        var aDate = [oDate.getHours(), oDate.getMinutes(), oDate.getSeconds()];
        for (var i in aDate) aSpan[i].innerHTML = format(aDate[i])
    }
    function format(a)
    {
        return a.toString().replace(/^(\\d)$/, "0$1")
    }
}
</script>
</head>
```

```
<body>
<div id="clock">
    <span></span>点<span></span>分<span></span>秒
</div>
</body>
</html>
```

4、如何阻止冒泡和默认事件？

通过 使用 `preventDefault()` 方法 只 取消 默 认 的 行 为 。

jQuery 代码：

代码如下：

```
$("#form").bind(
    "submit",
    function(event) {
        event.preventDefault();
    }
);
```

通过 使用 `stopPropagation()` 方法 只 阻 止 一 个 事 件 起 泡 。

jQuery 代码：

代码如下：

```
$("#form").bind(
    "submit",
    function(event) {
        event.stopPropagation();
    }
);
```

19.请指出“get()”“[]”“eq()”的区别？

答：仔细查看下 API 文档就可以知道。`eq` 返回的是一个 jquery 对象，`get` 返回的是一个 html 对象数组。举个例子：

```
<p style="color:yellow">绯雨</p>
```

使用 `eq` 来获得第一个 `p` 标签的 `color` 值：

```
$("#p").eq(0).css("color") //因为 eq(num) 返回的是个 jq 对象，所以可以用 jq 的方法 css
```

使用 `get` 来获得第一个 `p` 标签的 `color` 值：

```
$("#p").get(0).style.color //因为 get(num) 返回的是个 html 对象，所以要使用传统的 HTML 对象方法，jq 对象此时就没用了。
```

当然，你也可以 `get(num)` 后把对象转为 `jq` 的对象再进行操作：

```
$( $("p").get(0) ).css("color")
```

20. 请说出你可以传递到 jQuery 方法的四种不同值。

答：选择器（字符串），HTML（字符串），回调函数，HTML 元素，对象，数组，元素数组，jQuery 对象等。

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。

其他问题

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？

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)

```
}
```

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。

- 1、实现界面交互
- 2、提升用户体验
- 3、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。

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?

Node.js、Mongodb、npm、MVVM、MEAN、three.js

移动端 (Android IOS) 怎么做好用户体验?

清晰的视觉纵线、信息的分组、极致的减法、□ 利用选择代替输入、标签及文字的排布方式、□ 依靠明文确认密码、合理的键盘利用、

前端页面有哪三层构成, 分别是什么? 作用是什么?

最准确的网页设计思路是把网页分成三个层次, 即: 结构层、表示层、行为层。

网页的结构层 (structural layer) 由 HTML 或 XHTML 之类的标记语言负责创建。标签, 也就是那些出现在尖括号里的单词, 对网页内容的语义含义做出了描述, 但这些标签不包含任何关于如何显示有关内容的信息。例如, P 标

签表达了这样一种语义：“这是一个文本段。”

网页的表示层（presentation layer）由 CSS 负责创建。CSS 对“如何显示有关内容”的问题做出了回答。

网页的行为层（behavior layer）负责回答“内容应该如何对事件做出反应”这一问题。这是 Javascript 语言和 DOM 主宰的领域。

页面重构怎么操作？

编写 CSS、让页面结构更合理化，提升用户体验，实现良好的页面效果和提升性能。

网站为什么要进行重构（网站重构的好处）

- a、使页面加载得更快速；
- b、降低带宽带来的费用：节约成本；
- c、让你在修改设计时更有效率而代价更低；
- d、帮助你的整个站点保持视觉的一致性；
- e、更利于搜索引擎的检索（符合 SEO 的规范）；
- f、令站点更容易被各种浏览器和用户访问（包括手机、PDA 和残障人士使用的文字浏览器）；
- g、兼容不容忽视的 Mozilla 系浏览器（Firefox 份额）；
- h、提高你的职场竞争实力(事实上也就是降低失业的风险)。

什么叫优雅降级和渐进增强？

优雅降级：Web 站点在所有新式浏览器中都能正常工作，如果用户使用的是老式浏览器，则代码会检查以确认它们是否能正常工作。由于 IE 独特的盒模型布局问题，针对不同版本的 IE 的 hack 实践过优雅降级了，为那些无法支持功能的浏览器增加候选方案，使之在旧式浏览器上以某种形式降级体验却不至于完全失效。

渐进增强：从被所有浏览器支持的基本功能开始，逐步地添加那些只有新式浏览器才支持的功能，向页面增加无害于基础浏览器的额外样式和功能的。当浏览器支持时，它们会自动地呈现出来并发挥作用。

WEB 应用从服务器主动推送 Data 到客户端有那些方式？

html5 websocket

WebSocket 通过 Flash

XHR 长时间连接

XHR Multipart Streaming

不可见的 Iframe

<script>标签的长时间连接(可跨域)