HTML 常见题目

01、Doctype作用？严格模式与混杂模式如何区分？它们有何意义？

Doctype作用：告知浏览器的解析器用什么文档标准解析这个文

档，DOCTYPE不存在或格式不正确会导致文档以兼容模式（混杂模式）呈现；

标准模式(严格模式)的排版和JS运作模式都是以该浏览器支持的最高标准运行。

在兼容模式（混杂模式或怪异模式）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，就是尽可能的显示能显示的东西给用户看。；

随着标准一致性越来越重要，浏览器开发商不得不面临一个艰难的抉择：逐渐遵循w3c的标准是前进的方向。但是改变现有的css，完全去遵循标准，会使许多旧网站或多或少的收到破坏，如果浏览器突然以正确的方式解析现存的css，陈旧的网站的显示必然会受到影响。所以，所有的浏览器都需要提供两种模式，混杂模式服务于旧世规则，严格模式服务于标准规则；

具体的说：

1.width不同

在严格模式中： width是内容宽度，元素真正的宽度 = margin-left + border-left-width + padding-left + width + padding-right + border-right- width + margin-right;

在兼容模式中： width则是元素的实际宽度，内容宽度 = width - (padding-left + padding-right + border-left-width + border-right-width)

2.兼容模式下可设置百分比的高度和行内元素的高宽

在Standards模式下，给span等行内元素设置wdith和height都不会生效，而在兼容模式下，则会生效。

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高度，子元素设置一个百分比的高度是无效的。

3.用margin:0 auto设置水平居中在IE下会失效

使用margin:0 auto在standards模式下可以使元素水平居中，但在兼容模式下却会失效（用text-align属性解决）

body{text-align:center};#content{text-align:left}

4.兼容模式下Table中的字体属性不能继承上层的设置，white-space:pre会失效，设置图片的padding会失效

02、HTML5 为什么只需要写 <!DOCTYPE HTML>?

HTML 4.01 中的 doctype 需要对 DTD 进行引用，因为 HTML 4.01 基于 SGML。而 HTML 5 不基于 SGML，因此不需要对 DTD 进行引用，但是需要 doctype 来规范浏览器的行为（让浏览器按照它们应该的方式来运行。

03、行内元素有哪些？块级元素有哪些？ 空(void)元素有那些?

（1）行内元素有：a b span img input select strong（强调的语气）

（2）块级元素有：div ul ol li dl dt dd h1 h2 h3 h4…p （导航菜单，列表，选择框，页脚）

（3）常见的空元素：

 <hr> <img> <input> <link> <meta>

少见的：<embed> <source> <col>

04、页面导入样式时，使用link和@import有什么区别?

（1）link属于XHTML标签，除了加载CSS外，还能用于定义RSS, 定义rel连接属性等作用；而@import是CSS提供的，只能用于加载CSS;

（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是CSS2.1 提出的，只在IE5以上才能被识别，而link是XHTML标签，无兼容问题;

05、介绍一下你对浏览器内核的理解?

主要分成两部分：**渲染引擎**(layout engineer或Rendering Engine)和**JS引擎。**

渲染引擎：负责取得网页的内容（HTML、XML、图像等等）、整理讯息（例如加入CSS等），以及计算网页的显示方式，然后会输出至显示器或打印机。浏览器的内核的不同对于网页的语法解释会有不同，所以渲染的效果也不相同。所有网页浏览器、电子邮件客户端以及

其它需要编辑、显示网络内容的应用程序都需要内核。
JS引擎：解析和执行javascript来实现网页的动态效果。
最开始渲染引擎和JS引擎并没有区分的很明确，后来JS引擎越来越独立，内核就倾向于只指渲染引擎。

06、常见的浏览器内核有哪些?

Trident-IE浏览器内核 :IE,MaxThon,TT,The World,360,搜狗浏览器等
Gecko-火狐浏览器内核Mozilla :
Blink(Webkit的分支)-谷歌浏览器内核 :Safari,Chrome等
Presto，现为Blink-Opera浏览器内核:

07、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?

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。
增加的元素
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；
语义化更好的元素，比如article，footer，header，nav，section；
表单控件：calender，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doument.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

```
<!--[if it IE9]>
<script> src="http://html5shim.googlecode.com/svn/trunk/html5.js"</script>
<![endif]-->
```

如何区分HTML和HTML5：
DOCTYPE声明，新增的结构元素，功能元素

html5shim 可以让你的 IE9 或者更低版本的 IE 浏览器支持 HTML5。
使用方法：
head中加入

```
<!--[if lt IE 9]>
<script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->
```

08、如何区分 HTML 和 HTML5?

1、在文档类型声明上
HTML声明：<!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">

HTML5声明：<!doctype html>

上面的两种声明,HTML5声明简洁方便人们的记忆，HTML声明太长了并且很难记住这段代码。

2、在结构语义上

HTML:没有体现结构语义化的标签，通常都是这样来命名的 ，这样表示网站的头部。

HTML5:在语义上却有很大的优势，提供了一些新的HTML5标签比如: article、footer、header、nav、section，这些通俗易懂。

09、简述一下你对HTML语义化的理解?

1、什么是HTML语义化?

<基本上都是围绕着几个主要的标签，像标题（H1~H6）、列表（li）、强调（strong em）菜单menu 页脚footer 页头head 等等>

根据内容的结构化（内容语义化），选择合适的标签（代码语义化）便于开发者阅读和写出更优雅的代码的同时让浏览器的爬虫和机器很好地解析。

2、为什么要语义化?

为了在没有CSS的情况下，页面也能呈现出很好地内容结构、代码结构:为了裸奔时好看；
用户体验：例如title、alt用于解释名词或解释图片信息、label标签的活用；
有利于SEO：和搜索引擎建立良好沟通，有助于爬虫抓取更多的有效信息：爬虫依赖于标签来确定上下文和各个关键字的权重；
方便其他设备解析（如屏幕阅读器、盲人阅读器、移动设备）以意义的方式来渲染网页；
便于团队开发和维护，语义化更具可读性，是下一步吧网页的重要动向，遵循W3C标准的团队都遵循这个标准，可以减少差异化。

3、写HTML代码时应注意什么?

尽可能少的使用无语义的标签div和span；
在语义不明显时，既可以使用div或者p时，尽量用p, 因为p在默认情况下有上下间距，对兼容特殊终端有利；
不要使用纯样式标签，如：b、font、u等，改用css设置。
需要强调的文本，可以包含在strong或者em标签中（浏览器预设样式，能用CSS指定就不用他们），strong默认样式是加粗（不要用b），em是斜体（不用i）；
使用表格时，标题要用caption，表头用thead，主体部分用tbody包围，尾部用tfoot包围。表头和一般单元格要区分开，表头用th，单元格用td；
表单域要用fieldset标签包起来，并用legend标签说明表单的用途；
每个input标签对应的说明文本都需要使用label标签，并且通过为input设置id属性，在lable标签中设置for=someId来让说明文本和相对应的input关联起来。

4、HTML5新增了哪些语义标签，详述之。

<section></section>用于对网站或应用程序中页面上的内容进行分块。通常由内容及其标题组成。
<article></article>代表文档、页面或应用程序中独立的、完整的、可以独自被外部引用的内容。当我们描述一件具体的事物的时候，通常使用article来代替section。如一个帖子，一段用户评论等。
<aside></aside>表示当前页面或者文章的附属信息部分。如与当前页面或主要内容相关的引用、侧边栏、广告、nav元素组等。
<nav></nav>用作页面导航的链接组，其中可以包括
<p>元素等。
<time></time>表示某个时间或者某个日期。其中pubdate属性代表了文档的发布日期，可以用到time标签里。
<header></header>整个页面或者页面内容区块的标题，可以包含其他内容。
<footer></footer>页脚，页面底部或者版块的内容。
<hgroup></hgroup>页面上标题的组合，通常对h1~h6进行分组。
<address></address>文档作者或者文档维护者的联系信息。
<figure></figure>通常用于图片，统计图或代码示例，带有可选标题。将其从网页上移除后不会对网页上其他内容产生影响。
<figcaption></figcaption>表示figure的标题，从属于figure元素。
<mark></mark>页面中需要凸显出或者高亮显示的，对于当前用户具有参考作用的一段文字。

<progress></progress>代表一个任务完成的进度。
<details></details>描述文档或者用户要求得到并且可以得到的细节信息。与summary元素配合使用。
<summary></summary>给details元素提供标题或者图例。标题是可见的，用户点击标题时，会显示细节信息。
<datelist></datelist>选项列表。与input元素配合使用，来定义input可能的值。
<keygen></keygen>给表单添加一个公钥.
<menu></menu>菜单列表。HTML4中不推荐使用。

10、HTML5的离线储存怎么使用，工作原理能不能解释一下?

HTML5离线存储存储功能非常强大，它的作用是：在用户没有与因特网连接时，可以正常访问站点或应用，在用户与因特网连接时，自动更新缓存数据。

原理：
HTML5的离线存储是基于一个新建的.appcache文件的，通过这个文件上的解析清单离线存储资源，这些资源就会像cookie一样被存储了下来。之后当网络在处于离线状态下时，浏览器会通过被离线存储的数据进行页面展示。

怎么用：
首先，在html页面头部加入一个manifest的属性:

```
<!DOCTYPE HTML>
<html manifest = "cache.manifest">
...
</html>
```

然后书写cache.manifest文件：
CACHE MANIFEST
#v0.11
CACHE:
js/app.js
css/style.css
NETWORK:
resourse/logo.png
FALLBACK:
/ /offline.html

详解：
manifest （即 .appcache 文件）文件是简单的文本文件，可分为三个部分：

CACHE：
在此标题下列出的文件将在首次下载后进行缓存。（由于包含manifest文件的页面将被自动离线存储，所以不需要把页面自身也列出来）
NETWORK：
在此标题下列出的文件需要与服务器的连接，且不会被缓存，离线时无法使用。
可以使用 “*” 来指示所有其他资源/文件都需要因特网连接：
NETWORK: *
如果在CACHE和NETWORK中有一个相同的资源，那么这个资源还是会被离线存储，也就是说CACHE的优先级更高。
FALLBACK：
在此标题下列出的文件规定当页面无法访问时的回退页面。比如上面这个文件表示的就是如果访问根目录下任何一个资源失败了，那么就去访问offline.html。

如何使用---简单讲解：
1、页面头部像下面一样加入一个manifest的属性；
2、在cache.manifest文件的编写离线存储的资源；
CACHE MANIFEST

#v0.11
CACHE:
js/app.js
css/style.css
NETWORK:
resourse/logo.png
FALLBACK:
/ /offline.html
3、在离线状态时，操作window.applicationCache进行需求实现。
特别注意：**Web标准弃用window.applicationCache**
此功能已从 Web 标准中删除。尽管一些浏览器可能仍然支持它，但它正在被丢弃。避免使用它并尽可能更新现有的代码。请注意，此功能可能随时停止工作。

11、浏览器是怎么对HTML5的离线储存资源进行管理和加载的呢?

在线的情况下，浏览器发现html头部有manifest属性，它会请求manifest文件，如果是第一次访问app，那么浏览器就会根据manifest文件的内容下载相应的资源并且进行离线存储。如果已经访问过app并且资源已经离线存储了，那么浏览器就会使用离线的资源加载页面，然后浏览器会对比新的manifest文件与旧的manifest文件，如果文件没有发生改变，就不做任何操作，如果文件改变了，那么就会重新下载文件中的资源并进行离线存储。
离线的情况下，浏览器就直接使用离线存储的资源。

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

cookies，sessionStorage 和 localStorage 的区别?
cookie是网站为了标示用户身份而储存在用户本地终端（Client Side）上的数据（通常经过加密）。
cookie数据始终在同源的http请求中携带（即使不需要），记会在浏览器和服务器间来回传递。
sessionStorage和localStorage不会自动把数据发给服务器，仅在本地保存。

存储大小：
cookie数据大小不能超过4k。
sessionStorage和localStorage 虽然也有存储大小的限制，但比cookie大得多，可以达到5M或更大。

有期时间：
localStorage 存储持久数据，浏览器关闭后数据不丢失除非主动删除数据；
sessionStorage 数据在当前浏览器窗口关闭后自动删除。
cookie 设置的cookie过期时间之前一直有效，即使窗口或浏览器关闭

13、iframe有那些缺点?

缺点：

- 框架结构中出现各种滚动条
- iframe 会阻塞主页面的 Onload 事件
- 搜索引擎的检索程序无法解读这种页面，不利于 SEO
- 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。

如果使用iframe，最好通过JavaScript动态的给iframe添加src属性值，这样可以绕过以上两个问题。

```
<iframe id="myframe" src="http://www.baidu.com">
  <p>您的浏览器不支持iframe。</p>
</iframe>
<br><br>
<input type="button" onclick="changeSrc()" value="改变地址">

<script>
  function changeSrc()
```

```
 {
   document.getElementById("myframe").src="http://google.com";
 }
</script>
```

优点：

- iframe 能够原封不动的把嵌入的网页展现出来。
- 如果有多个网页引用 iframe，那么你只需要修改 iframe 的内容，就可以实现调用的每一个页面内容的更改，方便快捷。
- 网页如果为了统一风格，头部和版本都是一样的，就可以写成一个页面，用 iframe 来嵌套，可以增加代码的可重用。
- 如果遇到加载缓慢的第三方内容如图标和广告，这些问题可以由 iframe 来解决。

14、Label的作用是什么？是怎么用的？（加 for 或 包裹）

label标签来定义表单控制间的关系,当用户选择该标签时，浏览器会自动将焦点转到和标签相关的表单控件上。

```
<label for="Name">Number:</label> <input type= “text “name="Name" id="Name"/>

<label>Date:<input type="text" name="B" /></label>
```

15、HTML5的form如何关闭自动完成功能？

给不想要提示的 form 或下某个input 设置为 autocomplete=off。

定义和用法

autocomplete 属性规定表单是否应该启用自动完成功能。

自动完成允许浏览器预测对字段的输入。当用户在字段开始键入时，浏览器基于之前键入过的值，应该显示出在字段中填写的选项。

注释：autocomplete 属性适用于 <form>，以及下面的 <input> 类型：text, search, url, telephone, email, password, datepickers, range 以及 color。

提示：在某些浏览器中，您可能需要手动启用自动完成功能。

16、如何实现浏览器内多个标签页之间的通信？(阿里)

调用localstorge、sessionStorage、cookies等本地存储方式

使用vue中vuex

17、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

18、页面可见性（Page Visibility）API 可以有哪些用途？

页面可见性： 就是对于用户来说，页面是显示还是隐藏, 所谓显示的页面，就是我们正在看的页面；隐藏的页面，就是我们没有看的页面。 因为，我们一次可以打开好多标签页面来回切换着，始终只有一个页面在我们眼前，其他页面就是隐藏的，还有一种就是.....，(把浏览器最小化，所有的页面就都不可见了)。

API 很简单，document.hidden 就返回一个布尔值，如果是true, 表示页面可见，false 则表示，页面隐藏。 不同页面之间来回切换，触发visibilitychange事件。 还有一个document.visibilityState, 表示页面所处的状态，取值：visible, hidden 等四个。

实例代码

```
document.addEventListener("visibilitychange", function(){
   if(document.hidden){
```

```
    document.title ="hidden";
  }else {
    document.title = "visibile";
  }
})
```

我们打开这个页面，然后再打开另一个页面，来回点击这两个页面，当我们看到这个页面时，标题显示visiable ,当我们看另一个页面时，标题显示hidden;

动画，视频，音频都可以在页面显示时打开，在页面隐藏时关闭。

19、如何在页面上实现一个圆形的可点击区域?

1、map+area或者svg
2、border-radius
3、纯js实现 需要求一个点在不在圆上简单算法、获取鼠标坐标等等

20、实现不使用 border 画出1px高的线，在不同浏览器的Quirksmode和CSSCompat模式下都能保持同一效果。

```
<div style="height:1px;overflow:hidden;background:red"></div>
```

21、网页验证码是干嘛的，是为了解决什么安全问题?

区分用户是计算机还是人的公共全自动程序。可以防止恶意破解密码、刷票、论坛灌水；
有效防止黑客对某一个特定注册用户用特定程序暴力破解方式进行不断的登陆尝试。

22、tite与h1的区别、b与strong的区别、i与em的区别?

title属性没有明确意义只表示是个标题，H1则表示层次明确的标题，对页面信息的抓取也有很大的影响；

strong是标明重点内容，有语气加强的含义，使用阅读设备阅读网络时：会重读，而<B>是展示强调内容。

i内容展示为斜体，em表示强调的文本；

Physical Style Elements -- 自然样式标签
b, i, u, s, pre
Semantic Style Elements -- 语义样式标签
strong, em, ins, del, code
应该准确使用语义样式标签, 但不能滥用, 如果不能确定时首选使用自然样式标签。

CSS常见题目

介绍一下标准的CSS的盒子模型？低版本IE的盒子模型有什么不同的?

（1）有两种， IE 盒子模型、W3C 盒子模型；
（2）盒模型： 内容(content)、填充(padding)、边界(margin)、 边框(border)；
（3）区 别： IE的content部分把 border 和 pading计算了进去;

CSS选择符有哪些？哪些属性可以继承?

* 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 ;

CSS优先级算法如何计算?

* 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 > 内联样式 > id > class > tag
! important 比 内联优先级高

CSS3新增伪类有那些?

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。

:after 在元素之前添加内容,也可以用来做**清除浮动。**
:before 在元素之后添加内容
: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？如何让绝对定位的div居中?

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%;
}
```

让绝对定位的div居中

```
position: absolute;
width: 1200px;
background: none;
margin: 0 auto;
top: 0;
left: 0;
bottom: 0;
right: 0;
```

---

display有哪些值？说明他们的作用。

block　　像块类型元素一样显示。
none　　缺省值。像行内元素类型一样显示。
inline-block　像行内元素一样显示，但其内容像块类型元素一样显示。
list-item　像块类型元素一样显示，并添加样式列表标记。
table　　此元素会作为块级表格来显示
inherit　　规定应该从父元素继承 display 属性的值

---

position的值relative和absolute定位原点是?

absolute
生成绝对定位的元素，相对于值不为 static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?

新增各种CSS选择器　(: not(.input)：所有 class 不是“input”的节点)
圆角　　(border-radius:8px)
多列布局　　(multi-column layout)
阴影和反射　　(Shadow\Reflect)
文字特效　　(text-shadow、)
文字渲染　　(Text-decoration)
线性渐变　　(gradient)
旋转　　(transform)
增加了旋转,缩放,定位,倾斜,动画，多背景
transform:\scale(0.85,0.90)\ translate(0px,-30px)\ skew(-9deg,0deg)\Animation:

请解释一下CSS3的Flexbox（弹性盒布局模型）,以及适用场景?

该布局模型的目的是提供一种更加高效的方式来对容器中的条目进行布局、对齐和分配空间，在传统的布局中，block布局是把块级元素在垂直方向从上向下一次排列的，而inline布局则是在水平方向来排列。弹性盒布局没有这样的内在限制，操作比较自由。
适用场景：垂直居中，处于中心，以及左中右三块布局

用纯CSS创建一个三角形的原理是什么?

把上、左、右三条边隐藏掉（颜色设为 transparent）

```
#demo {
  width: 0;
  height: 0;
  border-width: 20px;
  border-style: solid;
  border-color: transparent transparent red transparent;
}
```

学习例子：

```
<html><head>
<style>
#demo {
  width: 0;
  height: 0;
  border-width: 20px;
  border-style: solid;
  border-color: red  transparent red transparent;
}
</style>
</head>
<body>
<div id="demo"></div>
</body></html>
```

一个满屏 品 字布局 如何设计?

简单的方式:

上面的div宽100%，

下面的两个div分别宽50%，

然后用float或者inline使其不换行即可

经常遇到的浏览器的兼容性有哪些？原因，解决方法是什么，常用hack的技巧？

* 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游览器从所有情况中分离出来。
接着，再次使用“+”将IE8和IE7、IE6分离开来，这样IE8已经独立识别。

```
css
    .bb{
```

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

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

* IE下,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 {}

---

li与li之间有看不见的空白间隔是什么原因引起的？有什么解决办法?

行框的排列会受到中间空白（回车\空格）等的影响，因为空格也属于字符,这些空白也会被应用样式，占据空间，所以会有间隔，把字符大小设为0，就没有空格了。

方法一：既然是因为<li>换行导致的，那就可以将<li>代码全部写在一排
但代码会不美观
方法二:
.wrap ul{font-size:0px;}
但随着而来的就是<ul>中的其他文字就不见了，因为其尺寸被设为0px了，我们只好将他们重新设定字符尺寸。
方法三:
本来以为方法二能够完全解决问题，但经测试，将li父级标签字符设置为0在Safari浏览器依然出现间隔空白；既然设置字符大小为0不行，那咱就将间隔消除了，将下面代码替换方法二的代码，目前测试完美解决。同样随来而来的问题是li内的字符间隔也被设置了，我们需要将li内的字符间隔设为默认。
.wrap ul{letter-spacing: -5px;}
之后记得设置li内字符间隔
.wrap ul li{letter-spacing: normal;}

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（搜索引擎优化）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： * {padding: 0; margin: 0;} （强烈不建议）

淘宝的样式初始化代码:
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(容器块)计算方式跟正常流有什么不同?

无论属于哪种，都要先找到其祖先元素中最近的 position 值不为 static 的元素，然后再判断：
1、若此元素为 inline 元素，则 containing block 为能够包含这个元素生成的第一个和最后一个 inline box 的 padding box (除 margin, border 外的区域) 的最小矩形；
2、否则,则由这个祖先元素的 padding box 构成。
如果都找不到，则为 initial containing block。

补充：
1. static(默认的)/relative：简单说就是它的父元素的内容框（即去掉padding的部分）
2. absolute: 向上找最近的定位为absolute/relative的元素
3. fixed: 它的containing block一律为根元素(html/body)，根元素也是initial containing block

---

CSS里的visibility属性有个collapse属性值是干嘛用的？在不同浏览器下以后什么区别?

visibility有如下属性值：
visible　默认值。元素是可见的。
hidden　元素是不可见的，相当于display：hidden；，但此时仍占用页面空间
collapse　当在表格元素中使用时，此值可删除一行或一列，但是它不会影响表格的布局。被行或列占据的空间会留给其他内容使用。如果此值被用在其他的元素上，会呈现为 “hidden”。
inherit　规定应该从父元素继承 visibility 属性的值。
visibility的第三种值collapse：
对于一般的元素，它的表现跟display:hidden是一样的。
但例外的是，如果这个元素是table相关的元素，例如table行，table group，table列，table column group，它的表现却跟display: none一样，也就是说，它们占用的空间也会释放。
在不同浏览器下的区别：

- 在谷歌浏览器里，使用collapse值和使用hidden值没有什么区别。
- 在火狐浏览器、Opera和IE11里，使用collapse值的效果就如它的字面意思：table的行会消失，它的下面一行会补充它的位置。

---

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

（1）如果display为none，这是就不产生盒子，所以会忽略position和float，否则
（2）如果position的值为absolute或fixed，float会被转换为none

设置值	计算值

inline-table	table
inline, run-in, table-row-group, table-column, table-column-group,table-header-group, table-footer-group, table-row, table-cell,table-caption, inline-block	block
其他，如list-item	同设定值

（3）如果position不是绝对布局，这时，如果float不为none或者该元素是根元素，则依旧按上表进行转换。

（4）否则如果，position不是绝对布局，并且float为none不浮动，且不是跟元素，display就取设置值。

对BFC规范(块级格式化上下文：block formatting context)的理解?

（W3C CSS 2.1 规范中的一个概念,它是一个独立容器，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。）

一个页面是由很多个 Box 组成的,元素的类型和 display 属性,决定了这个 Box 的类型。

不同类型的 Box,会参与不同的 Formatting Context（决定如何渲染文档的容器）,因此Box内的元素会以不同的方式渲染,也就是说BFC内部的元素和外部的元素不会互相影响。

BFC(Block formatting context)直译为"块级格式化上下文"。它是一个独立的渲染区域，只有Block-level box参与， 它规定了内部的Block-level Box如何布局，并且与这个区域外部毫不相干。

BFC有一些规则：

内部的Box会在垂直方向，一个接一个地放置。

Box垂直方向的距离由margin决定。属于同一个BFC的两个相邻Box的margin会发生重叠。

每个元素的margin box的左边， 与包含块border box的左边相接触(对于从左往右的格式化，否则相反)。即使存在浮动也是如此。

BFC的区域不会与float box重叠。

BFC就是页面上的一个隔离的独立容器，容器里面的子元素不会影响到外面的元素。反之也如此。

计算BFC的高度时，浮动元素也参与计算。

触发BFC

根元素

float属性不为none

position为absolute或fixed

display为inline-block, table-cell, table-caption, flex, inline-flex

overflow不为visible

BFC应用：

两列布局。

清除内部浮动，撑开父高度。

防止两个margin重叠。

---

css定义的权重

声明位于内联样式，权重为1000（始终）

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

请解释一下为什么会出现浮动和什么时候需要清除浮动？清除浮动的方式

为什么出现浮动：

原本的浮动是用来实现左边图片右边文字的样式，后来慢慢发展成布局的主要手段，但是这个浮动布局会破坏文档流。渐渐的现在也建议不使用，现在css3.0提供了伸缩盒模型（flex），慢慢的将成为主流的布局方式

什么时候需要清除浮动：

浮动的属性虽然方便使用，但是在使用这种属性时，也存在着一种弊端，那就是当子元素设置了float属性之后，且父元素的高度和宽度没有进行设置，而是由子元素支撑起8来，则会导致父元素的高度塌陷(原本的height后来被重置为0)

***清除浮动的5种方法：**

1、父级div定义overflow:hidden

原理：使用overflow:hidden时，浏览器会自动检查浮动区域的高度。

优点：简单，代码少，浏览器支持好。

缺点：必须定义width或zoom:1，不能和position配合使用，因为超出的尺寸的会被隐藏。

建议：只推荐没有使用position或对overflow:hidden理解的朋友使用。

2、结尾处加空div标签clear:both

```
<div class="box1">
<div class="child-1">child-1</div>
<div class="child-2">child-2</div>
<div style="clear: both;"></div>
</div>
```

原理：添加一个空div，利用css提高的clear:both清除浮动，让父级div能自动获取到高度。

优点：简单，代码少，浏览器支持好，不容易出现怪问题。

缺点：不少初学者不理解原理；如果页面浮动布局多，就要增加很多空div，让人感觉很不爽。

建议：此方法是以前主要使用的一种清除浮动方法。

3、父级div定义height

原理：父级div手动定义height，就解决了父级div无法自动获取到高度的问题。

优点：简单，代码少，容易掌握。

缺点：只适合高度固定的布局，要给出精确的高度，如果高度和父级div不一样时，会产生问题。

建议：不推荐使用，只建议高度固定的布局时使用。

4、父级div定义overflow:auto

```
.div1{background:#000080;border:1px solid red;width:98%;overflow:auto}
```

原理：同1，使用overflow:auto时，浏览器会自动检查浮动区域的高度。

优点：简单，代码少，浏览器支持好。

缺点：内部宽高超过父级div时，会出现滚动条。

建议：不推荐使用，如果你需要出现滚动条或者确保你的代码不会出现滚动条就使用吧。

5、父级div定义伪类:after和zoom

```
<style type="text/css">
 .div1{background:#000080;border:1px solid red;}     .left{float:left;width:20%;height:200px;background:#DDD}
.right{float:right;width:30%;height:80px;background:#DDD}
.clearfloat:after{display:block;clear:both;content:"";visibility:hidden;height:0}
.clearfloat{zoom:1}
</style>
<div class="div1 clearfloat">
<div class="left">Left</div>
```

Right

优点：浏览器支持好，不容易出现怪问题（目前：大型网站都有使用，如：腾迅，网易，新浪等等）。
缺点：代码多，不少初学者不理解原理，要两句代码结合使用，才能让主流浏览器都支持。
建议：推荐使用，建议定义公共类，以减少CSS代码。

移动端的布局用过媒体查询吗？

用过，即响应式布局
通过媒体查询可以为不同大小和尺寸的媒体定义不同的css，适合相应的设备显示；
@media screen and (min-width: 400px) and (max-width: 700px) { ... }
@media handheld and (min-width: 20em), screen and (min-width: 20em) { ... }

<!-- link元素中的CSS媒体查询 -->
当媒体查询为真时，相关的样式表或样式规则会按照正常的级联规被应用。
当媒体查询返回假， <link> 标签上带有媒体查询的样式表 仍将被下载 （只不过不会被应用）。

<link rel="stylesheet" media="(max-width: 800px)" href="example.css" />

<!-- 样式表中的CSS媒体查询 -->
包含了一个媒体类型和至少一个使用 宽度、高度和颜色等媒体属性来限制样式表范围的表达式。
CSS3加入的媒体查询使得无需修改内容便可以使样式应用于某些特定的设备范围。

```
<style>
   @media (min-width: 700px) and (orientation: landscape){
     .sidebar {
       display: none;
     }
   }
</style>
```

使用 CSS 预处理器吗？喜欢那个？

LESS(SASS、LESS没有本质区别，只因为之前团队前端都是用的LESS)

CSS优化、提高性能的方法有哪些？

- 避免后代选择符
- 避免过度的约束
- 缩写css代码
    - background属性
    - border属性
- 避免不必要的命名空间
- 避免使用！important
- 较大的站点，分离网页颜色和背景设置样式

总结—拓展
使用简写 例:margin border background
查找并删除未使用的 CSS -浏览器开发工具栏就可以实现
打开开发者工具，点击 Audits 栏位，点击 Run audits 后就开始分析结果。
使用颜色快捷方式

```
target { background-color: #ffffff; }
target { background: #fff; }
```

删除不必要的零和单位

使用纹理图集

由于协议开销的原因，加载多个小图片的效率很低。CSS 精灵将一系列小图片组合成一个大的 PNG 文件，然后通过 CSS 规则将其分解。**TexturePacker** 等程序大大简化了创建过程。

```
.download { width:80px; height:31px; background-position: -160px -160px}
```

删除注释 发布时删除

浏览器是怎样解析CSS选择器的?

按照从上到下，从右到左的顺序解析。例如：.list a {color:blue；}先解析到 a 标签，并将页面上所有 a 标签的字体颜色都按照 color:blue 进行渲染（蓝色），再解析到 .list ，将页面上所有 .list 类目下的 a 标签的字体渲染成蓝色。

在网页中的应该使用奇数还是偶数的字体？为什么呢?

偶数字号相对更容易和 web 设计的其他部分构成比例关系。

比如：当我用了 14 px 的正文字号，我可能会在一些地方用 14 × 0.5 = 7 px 的 margin，在另一些地方用 14 × 1.5 = 21 px 的标题字号。

---

margin和padding分别适合什么场景使用?

使用margin值的情况:

1.需要在border外侧添加空白时;

2.空白处不需要背景色时;

3.上下相连得两个盒子之间的空白需要相互抵消时，如15px+20px得margin，将得到20px的空白。（margin折叠）

4.需要使用负值对页面布局时（margin值可以取负值，padding不行）。

使用padding时的情况:

1.需要在border内侧添加空白时。

2.空白处需要背景（色）时。

上下相连的两个盒子之间的空白，希望等于两者之和时。如15px + 10px将得到25px的空白。

最后，需注意margin是用来隔开元素与元素的间距;padding是用来隔开元素与内容的间隔。margin用于布局分开元素使元素与元素互不干扰;padding用于元素与内容之间的间隔，让内容与元素之间有一段距离。

在怪异盒模型中，一个块的总宽度受margin影响但不受padding影响。

margin是用来隔开元素与元素的间距；padding是用来隔开元素与内容的间隔。

margin用于布局分开元素使元素与元素互不相干;

padding用于元素与内容之间的间隔，让内容（文字）与（包裹）元素之间有一段距离

抽离样式模块怎么写，说出思路，有无实践经验？[阿里航旅的面试题]

先看视觉稿，把可复用的组件找出来。然后把命名和结构确定下来

例如api的接口，利用axios，

按钮组件重写等

https://github.com/jayli/jayl...

元素竖向的百分比设定是相对于容器的高度吗?

不是，是容器宽度。

全屏滚动的原理是什么？用到了CSS的那些属性?

（1）原理：方法一是整体的元素一直排列下去，假设有5个需要展示的全屏页面，那么高度是500%

，只是展示100%，剩下的可以通过transform进行y轴定位，也可以通过margin-top实现

（2）overflow：hidden；

transition：all 1000ms ease；

什么是响应式设计？响应式设计的基本原理是什么？如何兼容低版本的IE？

响应式网站设计(Responsive Web design)是一个网站能够兼容多个终端，而不是为每一个终端做一个特定的版本。

基本原理是通过媒体查询检测不同的设备屏幕尺寸做处理。

页面头部必须有meta声明的viewport。

<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0, user-scalable=no"/>

兼容IE可以使用JS辅助一下来解决

---

*视差滚动效果，如何给每页做不同的动画？（回到顶部，向下滑动要再次出现，和只出现一次分别怎么做？）

视差滚动（Parallax Scrolling）通过在网页向下滚动的时候，控制背景的移动速度比前景的移动速度慢来创建出令人惊叹的3D效果。

CSS3实现

优点：开发时间短、性能和开发效率比较好，缺点是不能兼容到低版本的浏览器

jQuery实现

通过控制不同层滚动速度，计算每一层的时间，控制滚动效果。

优点：能兼容到各个版本的，效果可控性好

缺点：开发起来对制作者要求高

插件实现方式

例如：parallax-scrolling，兼容性十分好

::before 和 :after中双冒号和单冒号 有什么区别？解释一下这2个伪元素的作用。

单冒号(:)用于CSS3伪类，双冒号(::)用于CSS3伪元素。

::before就是以一个子元素的存在，定义在元素主体内容之前的一个伪元素。并不存在于dom之中，只存在在页面之中。

:before 和 :after 这两个伪元素，是在CSS2.1里新出现的。起初，伪元素的前缀使用的是单冒号语法，但随着Web的进化，在CSS3的规范里，伪元素的语法被修改成使用双冒号，成为::before ::after

---

如何修改chrome记住密码后自动填充表单的黄色背景？

开启浏览器自带填充表单功能

设置表单属性 autocomplete="on" 开启自动填充表单，自己实现记住密码（关闭off）

```
<!-- 对整个表单设置 -->
<form autocomplete="on" method=".." action="..">
<!-- 或对单一元素设置 -->
<input type="text" name="textboxname" autocomplete="on">
```

设置css

```
input:-webkit-autofill, textarea:-webkit-autofill, select:-webkit-autofill {
    background-color: rgb(250, 255, 189); /* #FAFFBD; */
    background-image: none;
    color: rgb(0, 0, 0);
}
```

你对line-height是如何理解的？

line-height定义行高，设置行间的距离。应用方式是：用line-height减去font-size，得到的差（称为行间距）除2，分别添加到文本的顶部和底部。可以包含这些内容的最小框就是行框。

CSS中起高度作用的是height和line-height，没有定义height属性，最终其表现作用一定是line-height。

单行文本垂直居中：把line-height值设置为height一样大小的值可以实现单行文字的垂直居中，其实也可以把height删除。

多行文本垂直居中：需要设置display属性为inline-block。

设置元素浮动后，该元素的display值是多少？（自动变成display:block）

自动变成了 display:block

可以用 window.getComputedStyle(element).display 检查，确实是 block

怎么让Chrome支持小于12px 的文字?

1、用图片：如果是内容固定不变情况下，使用将小于12px文字内容切出做图片，这样不影响兼容也不影响美观。

2、使用12px及12px以上字体大小：为了兼容各大主流浏览器，建议设计美工图时候设置大于或等于12px的字体大小，如果是接单的这个时候就需要给客户讲解小于12px浏览器不兼容等事宜。

3、继续使用小于12px字体大小样式设置：如果不考虑chrome可以不用考虑兼容，同时在设置小于12px对象设置-webkit-text-size-adjust:none，做到最大兼容考虑。

4、使用css3里的一个属性：transform：scale（） //缩放

p{font-size:10px;-webkit-transform:scale(0.8);}//0.8是缩放比例

让页面里的字体变清晰，变细用CSS怎么做?

（-webkit-font-smoothing: antialiased;）

-webkit-font-smoothing在window系统下没有起作用，但是在IOS设备上起作用-webkit-font-smoothing：antialiased是最佳的，灰度平滑。

---

font-style属性可以让它赋值为“oblique” oblique是什么意思?

倾斜的字体样式

与italic的区别：italic和oblique都是向右倾斜的文字, 但区别在于Italic是指斜体字，而Oblique是倾斜的文字，**对于没有斜体的字体应该使用Oblique属性值**来实现倾斜的文字效果。也就是说：Italic是使用文字的斜体，Oblique是让没有斜体属性的文字倾斜。

---

position:fixed;在Android下无效怎么处理?

<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0, user-scalable=no"/>

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)

有空格时候会有间隙 解决：移除空格

margin正值的时候 解决：margin使用负值

使用font-size时候 解决：font-size:0、letter-spacing、word-spacing

---

overflow: scroll时不能平滑滚动的问题怎么处理?

一：-webkit-overflow-scrolling: touch;

-webkit-overflow-scrolling 属性控制元素在移动设备上是否使用滚动回弹效果.

auto: 使用普通滚动, 当手指从触摸屏上移开，滚动会立即停止。

touch: 使用具有回弹效果的滚动, 当手指从触摸屏上移开，内容会继续保持一段时间的滚动效果。继续滚动的速度和持续的时间和滚动手势的强烈程度成正比。同时也会创建一个新的堆栈上下文。

二：

（1）高度尺寸不确定的时候，使用：overflow-y：scroll;

（2）高度尺寸确定的，要么没有滚动条，要么直接出现，不会出现跳动。

（3）css3计算calc和vw单位巧妙实现滚动条出现页面不跳动：

.wrap-outer {

margin-left: calc(100vw - 100%);
}
或.wrap-outer {
padding-left: calc(100vw - 100%);
}
首先，.wrap-outer指的是居中定宽主体的父级，如果没有，创建一个
然后，calc是css3的计算
100vw是浏览器的内部宽度，而100%是可用宽度，不含滚动条
calc（100vw-100%）是浏览器的滚动条的宽度

有一个高度自适应的div，里面有两个div，一个高度100px，希望另一个填满剩下的高度。

（1）height：calc（100%-100px）
（2）absolute positioning：外层position：relative；
百分百自适应元素 position: absolute; top: 100px; bottom: 0; left: 0
（3）flex 父元素display:flex; flex-direction: column; 下面的子元素的设置flex: 1;

---

png、jpg、gif 这些图片格式解释一下，分别什么时候用。有没有了解过webp?

1）png是便携式网络图片（Portable Network Graphics）是一种无损数据压缩位图文件格式.
优点是：压缩比高，色彩好。 大多数地方都可以用。
2）jpg是一种针对相片使用的一种失真压缩方法，是一种破坏性的压缩，在色调及颜色平滑变化做的不错。在www上，被用来储存和传输照片的格式。
3）gif是一种位图文件格式，以8位色重现真色彩的图像。可以实现动画效果.
4）webp格式是谷歌在2010年推出的图片格式，压缩率只有jpg的2/3，大小比png小了45%。缺点是压缩的时间更久了，兼容性不好，目前谷歌和opera支持。

什么是Cookie 隔离？（或者说：请求资源的时候不要让它带cookie怎么做）

如果静态文件都放在主域名下，那静态文件请求的时候都带有的cookie的数据提交给server的，非常浪费流量，
所以不如隔离开。
因为cookie有域的限制，因此不能跨域提交请求，故使用非主要域名的时候，请求头中就不会带有cookie数据，
这样可以降低请求头的大小，降低请求时间，从而达到降低整体请求延时的目的。
同时这种方式不会将cookie传入Web Server，也减少了Web Server对cookie的处理分析环节，
提高了webserver的http请求的解析速度。
实现方法：静态资源放 CDN 不放在自己的服务器中。

---

style标签写在body后与body前有什么区别?

写在head标签中利于浏览器逐步渲染（resources downloading->CSSOM+DOM->RenderTree(composite)->Layout->paint）。具体渲染过程请参考

写在body标签后由于浏览器以逐行方式对html文档进行解析，当解析到写在尾部的样式表（外联或写在style标签）会导致浏览器停止之前的渲染，等待加载且解析样式表完成之后重新渲染，在windows的IE下可能会出现FOUC现象（即样式失效导致的页面闪烁问题）

什么是CSS 预处理器 / 后处理器?

- 预处理器例如：LESS、Sass、Stylus，用来预编译Sass或less，**增强了css代码的复用性**，
还有层级、mixin、变量、循环、函数等，具有很方便的UI组件模块化开发能力，极大的提高工作效率。

- 后处理器例如：PostCSS，通常被视为在完成的样式表中根据CSS规范处理CSS，让其更有效；**目前最常做的是给CSS属性添加浏览器私有前缀，实现跨浏览器兼容性的问题。**

JavaScript

介绍js的基本数据类型。

Undefined、Null、Boolean、Number、String、
ECMAScript 2015 新增:Symbol(创建后独一无二且不可变的数据类型

介绍js有哪些内置对象?

Object 是 JavaScript 中所有对象的父对象
数据封装类对象：Object、Array、Boolean、Number 和 String
其他对象：Function、Arguments、Math、Date、RegExp、Error

说几条写JavaScript的基本规范?

1.减少全局污染，避免太多的全局变量
2.for循环规范 for(var i=0,max=myarray.length;i<max;i++)
3..请使用 ===/!==来比较true/false或者数值
4.For循环、If语句必须使用大括号
5..命名规则　构造器函数首字母大写，如function Person(){}
6.注意写注释（团队开发）
7.不要在同一个行内声明多个变量
8.Switch语句中必须带有default分支

JavaScript原型，原型链？有什么特点?

每个对象都会在其内部初始化一个属性，就是prototype(原型)，当我们访问一个对象的属性时，
如果这个对象内部不存在这个属性，那么他就会去prototype里找这个属性，这个prototype又会有自己的prototype，
于是就这样一直找下去，也就是我们平时所说的原型链的概念。
关系：instance.constructor.prototype = instance.__proto__
特点:
JavaScript对象是通过引用来传递的，我们创建的每个新对象实体中并没有一份属于自己的原型副本。当我们修改原型时，与之相关的对象也会继承这一改变。

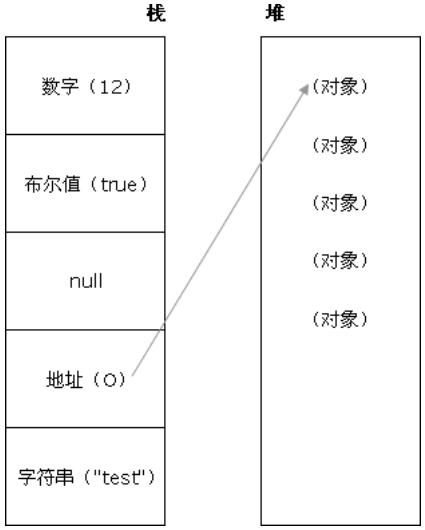
当我们需要一个属性的时，Javascript引擎会先看当前对象中是否有这个属性， 如果没有的话，
就会查找他的Prototype对象是否有这个属性，如此递推下去，一直检索到 Object 内建对象。
例子:

```
function Func(){}
Func.prototype.name = "Sean";
Func.prototype.getInfo = function() {
  return this.name;
}
var person = new Func();//现在可以参考var person = Object.create(oldObject);
console.log(person.getInfo());//它拥有了Func的属性和方法
//"Sean"
console.log(Func.prototype);
// Func { name="Sean", getInfo=function()}
```

JavaScript有几种类型的值？，你能画一下他们的内存图吗?

栈：原始数据类型（Undefined，Null，Boolean，Number、String）
堆：引用数据类型（对象、数组和函数）
两种类型的区别是：存储位置不同；
原始数据类型直接存储在栈(stack)中的简单数据段，占据空间小、大小固定，属于被频繁使用数据，所以放入栈中存储；

引用数据类型存储在堆(heap)中的对象,占据空间大、大小不固定,如果存储在栈中，将会影响程序运行的性能；引用数据类型在栈中存储了指针，该指针指向堆中该实体的起始地址。当解释器寻找引用值时，会首先检索其
在栈中的地址，取得地址后从堆中获得实体



如何将字符串转化为数字，例如'12.3b'?

* parseFloat('12.3b'); parseFloat（) 函数可解析一个字符串，并返回一个浮点数
* 正则表达式，'12.3b'.match(/(\d)+(\.)?(\d)+/g)[0] * 1, 但是这个不太靠谱，提供一种思路而已。

如何将浮点数点左边的数每三位添加一个逗号，如12000000.11转化为『12,000,000.11』?

```
function commafy(num){
    return num && num
        .toString()
        .replace(/(\d)(?=(\d{3})+\.)/g, function($1, $2){
            return $2 + ',';
        });
}
```

如何实现数组的随机排序?

方法一:

```
var arr = [1,2,3,4,5,6,7,8,9,10];
function randSort1(arr){
    for(var i = 0,len = arr.length;i < len; i++ ){
        var rand = parseInt(Math.random()*len);
        var temp = arr[rand];
        arr[rand] = arr[i];
        arr[i] = temp;
    }
    return arr;
}
console.log(randSort1(arr));
```

方法二:

```
var arr = [1,2,3,4,5,6,7,8,9,10];
function randSort2(arr){
    var mixedArray = [];
```

```
        while(arr.length > 0){
            var randomIndex = parseInt(Math.random()*arr.length);
            mixedArray.push(arr[randomIndex]);
            arr.splice(randomIndex, 1);
        }
        return mixedArray;
    }
    console.log(randSort2(arr));
```

方法三：

```
    var arr = [1,2,3,4,5,6,7,8,9,10];
    arr.sort(function(){
        return Math.random() - 0.5;
    })
    console.log(arr);
```

Javascript如何实现继承?

1、构造继承
2、原型继承
3、实例继承
4、拷贝继承
原型prototype机制或apply和call方法去实现较简单，建议使用构造函数与原型混合方式。

```
function Parent(){
    this.name = 'wang';
}

function Child(){
    this.age = 28;
}
Child.prototype = new Parent();//继承了Parent，通过原型

var demo = new Child();
alert(demo.age);
alert(demo.name);//得到被继承的属性
}
```

JavaScript继承的几种实现方式?

参考：
构造函数的继承：http://www.ruanyifeng.com/blog/2010/05/object-oriented_javascript_inheritance.html
非构造函数的继承：http://www.ruanyifeng.com/blog/2010/05/object-oriented_javascript_inheritance_continued.html

---

javascript创建对象的几种方式?

javascript创建对象简单的说,无非就是使用内置对象或各种自定义对象，当然还可以用JSON；但写法有很多种，也能混合使用。

1、对象字面量的方式

```
person={firstname:"Mark",lastname:"Yun",age:25,eyecolor:"black"};
```

2、用function来模拟无参的构造函数

```
function Person(){}
var person=new Person();//定义一个function，如果使用new"实例化",该function可以看作是一个Class
person.name="Mark";
person.age="25";
```

```
person.work=function(){
alert(person.name+" hello...");
}
person.work();
```

3、用function来模拟参构造函数来实现（用this关键字定义构造的上下文属性）

```
function Pet(name,age,hobby){
  this.name=name;//this作用域：当前对象
  this.age=age;
  this.hobby=hobby;
  this.eat=function(){
    alert("我叫"+this.name+",我喜欢"+this.hobby+",是个程序员");
  }
}
var maidou =new Pet("麦兜",25,"coding");//实例化、创建对象
maidou.eat();//调用eat方法
```

4、用工厂方式来创建（内置对象）

```
 var wcDog =new Object();
 wcDog.name="旺财";
 wcDog.age=3;
 wcDog.work=function(){
   alert("我是"+wcDog.name+",汪汪汪.....");
 }
 wcDog.work();
```

5、用原型方式来创建

```
function Dog(){
 }
 Dog.prototype.name="旺财";
 Dog.prototype.eat=function(){
 alert(this.name+"是个吃货");
 }
 var wangcai =new Dog();
 wangcai.eat();
```

6、用混合方式来创建

```
function Car(name,price){
  this.name=name;
  this.price=price;
}
 Car.prototype.sell=function(){
   alert("我是"+this.name+"，我现在卖"+this.price+"万元");
  }
var camry =new Car("凯美瑞",27);
camry.sell();
```

Javascript作用链域?

全局函数无法查看局部函数的内部细节，但局部函数可以查看其上层的函数细节，直至全局细节。
当需要从局部函数查找某一属性或方法时，如果当前作用域没有找到，就会上溯到上层作用域查找，
直至全局函数，这种组织形式就是作用域链。

谈谈This对象的理解。

this总是指向函数的直接调用者（而非间接调用者）；

如果有new关键字，this指向new出来的那个对象；
在事件中，this指向触发这个事件的对象，特殊的是，IE中的attachEvent中的this总是指向全局对象Window；

---

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）。
由JSON字符串转换为JSON对象的时候可以用eval，var obj =eval('('+ str +')');

什么是window对象? 什么是document对象?

window对象是指浏览器打开的窗口。
document对象是Documentd对象（HTML 文档对象）的一个只读引用，window对象的一个属性。

---

null，undefined 的区别?

null　　表示一个对象是“没有值”的值，也就是值为“空”；
undefined　　表示一个变量声明了没有初始化(赋值)；
undefined不是一个有效的JSON，而null是；
undefined的类型(typeof)是undefined；
null的类型(typeof)是object；
Javascript将未赋值的变量默认值设为undefined；
Javascript从来不会将变量设为null。它是用来让程序员表明某个用var声明的变量时没有值的。
typeof undefined
　　//"undefined"
　　undefined :是一个表示"无"的原始值或者说表示"缺少值"，就是此处应该有一个值，但是还没有定义。当尝试读取时会返回undefined；
　　例如变量被声明了，但没有赋值时，就等于undefined
typeof null
　　//"object"
　　null : 是一个对象(空对象, 没有任何属性和方法)；
　　例如作为函数的参数，表示该函数的参数不是对象；
注意：
　　在验证null时，一定要使用　===，因为 == 无法分别 null 和　undefined
　　null == undefined // true
　　null === undefined // false

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.removeEventListener) {
        element.removeEventListener(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;
    }
};
```

["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。
详细解析：http://blog.csdn.net/justjavac/article/details/19473199

重写parseInt函数测试一下是否符合上面的规则。

```
function parseInt(str, radix) {
    return str+'-'+radix;
};
var a=["1", "2", "3"];
a.map(parseInt);  // ["1-0", "2-1", "3-2"] 不能大于radix
```

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、Firefox同时支持两种事件模型，也就是：捕获型事件和冒泡型事件；
3. ev.stopPropagation();（旧ie的方法 ev.cancelBubble = true;）

声明一个变量加var，不加var区别。

"var a = 'aa' " 和 "a = 'aa' " 都是全局变量
加上'var'就可以作用域相关了，不加'var'始终都是在为'window'对象动态添加属性

```
var a = 'aa';
delete window.a; // false
a = 'aa';
delete window.a; // true
```

通过'var'声明的全局变量，其实际上是为'window'对象增加了一个不可配置的属性, 而不加'var'声明的全局变量，其实际上是为'window'对象增加了一个可以配置的属性
delete 不可以删除那些可配置性为false的属性"，某些内置对象的属性是不可配置的，比如通过变量声明或者函数声明创建的全局对象的属性，以下代码为证

```
delete Object.prototype; // false 不可删除，该属性是不可配置的
var a = 'aa';
delete window.a;//false 不可删除，该属性是不可配置的
function test(){};
delete window.test;//false 不可删除，该属性是不可配置的
```

什么是闭包（closure），为什么要用它?

闭包是指有权访问另一个函数作用域中变量的函数，**创建闭包的最常见的方式**就是在一个函数内创建另一个函数，通过另一个函数访问这个函数的局部变量,利用闭包可以突破作用链域，将函数内部的变量和方法传递到外部。

闭包的特性:

1.函数内再嵌套函数
2.内部函数可以引用外层的参数和变量
3.参数和变量不会被垃圾回收机制回收

```
//li节点的onclick事件都能正确的弹出当前被点击的li索引
 <ul id="testUL">
    <li> index = 0</li>
    <li> index = 1</li>
    <li> index = 2</li>
    <li> index = 3</li>
</ul>
<script type="text/javascript">
   var nodes = document.getElementsByTagName("li");
   for(i = 0;i<nodes.length;i+= 1){
      nodes[i].onclick = function(){
         console.log(i+1);//不用闭包的话，值每次都是4
      }(i);
   }
</script>
```

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
```

---

javascript 代码中的"use strict";是什么意思？使用它区别是什么?

use strict是一种ECMAscript 5 添加的（严格）运行模式,这种模式使得 Javascript 在更严格的条件下运行,
使JS编码更加规范化的模式,消除Javascript语法的一些不合理、不严谨之处，减少一些怪异行为。
默认支持的糟糕特性都会被禁用，比如不能用with，也不能在意外的情况下给全局变量赋值;
全局变量的显示声明,函数必须声明在顶层，不允许在非函数代码块内声明函数,arguments.callee也不允许使用;
消除代码运行的一些不安全之处，保证代码运行的安全,限制函数中的arguments修改，严格模式下的eval函数的行为和非严格模式的也不相同;
提高编译器效率，增加运行速度;
为未来新版本的Javascript标准化做铺垫。

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);
```

用原生JavaScript的实现过什么功能吗?

表单提交验证等

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?

hasOwnPropert

javaScript中hasOwnProperty函数方法是返回一个布尔值，指出一个对象是否具有指定名称的属性。此方法无法检查该对象的原型链中是否具有该属性；该属性必须是对象本身的一个成员。

使用方法：

object.hasOwnProperty(proName)

其中参数object是必选项。一个对象的实例。

proName是必选项。一个属性名称的字符串值。

如果 object 具有指定名称的属性，那么JavaScript中hasOwnProperty函数方法返回 true，反之则返回 false。

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

如：{"age":"12", "name":"back"}

JSON字符串转换为JSON对象:

```
var obj =eval('('+ str +')');
var obj = str.parseJSON();
var obj = JSON.parse(str);
```

JSON对象转换为JSON字符串：

```
var last=obj.toJSONString();
var last=JSON.stringify(obj);
```

[].forEach.call($$("*"),function(a){a.style.outline="1px solid #"+(~~(Math.random()*(1<<24))).toString(16)}) 能解释一下这段代码的意思吗?

1.call:call(thisObj,arg1,arg2,arg3)

[].forEach.call($$("*"),

    就是用$$('a')来替代[],

好 那么到了第二个问题$$('a')是什么意思

2.$$('a')

你可以在自己的浏览器上面运行一下，就是页面上所有的a标签

然后再继续

3.function(a){}

无疑就是$$('a')组成的数组要进行的回调函数了

4.~~
看在浏览器上面的运行
var a=12.233
~~a　　//12
所以~~的作用就相当于parseInt
5.1<<24
也就是1向左移24位
也就是2的24次方
6.toString(16)
就是把数字转换成16进制的字符串

---

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? 如何创建一个Ajax?

ajax的全称：Asynchronous Javascript And XML。
异步传输+js+xml。
所谓异步，在这里简单地解释就是：向服务器发送请求的时候，我们不必等待结果，而是可以同时做其他的事情，等到有了结果它自己会根据设定进行后续操作，与此同时，页面是不会发生整页刷新的，提高了用户体验。

(1)创建XMLHttpRequest对象,也就是创建一个异步调用对象
(2)创建一个新的HTTP请求,并指定该HTTP请求的方法、URL及验证信息
(3)设置响应HTTP请求状态变化的函数
(4)发送HTTP请求
(5)获取异步调用返回的数据
(6)使用JavaScript和DOM实现局部刷新

Ajax 解决浏览器缓存问题?

1、在ajax发送请求前加上 anyAjaxObj.setRequestHeader("If-Modified-Since","0")。
2、在ajax发送请求前加上 anyAjaxObj.setRequestHeader("Cache-Control","no-cache")。
3、在URL后面加上一个随机数： "fresh=" + Math.random();。
4、在URL后面加上时间戳："nowtime=" + new Date().getTime();。
5、如果是使用jQuery，直接这样就可以了 $.ajaxSetup({cache:false})。这样页面的所有ajax都会执行这条语句就是不需要保存缓存记录。

同步和异步的区别?

同步的概念应该是来自于OS中关于同步的概念:不同进程为协同完成某项工作而在先后次序上调整(通过阻塞,唤醒等方式).同步强调的是顺序性.谁先谁后.异步则不存在这种顺序性.
同步：浏览器访问服务器请求，用户看得到页面刷新，重新发请求,等请求完，页面刷新，新内容出现，用户看到新内容,进行下一步操作。
异步：浏览器访问服务器请求，用户正常操作，浏览器后端进行请求。等请求完，页面不刷新，新内容也会出现，用户看到新内容。
（待完善）

如何解决跨域问题?

jsonp、 iframe、window.name、window.postMessage、服务器上设置代理页面
xios 跨域设置
前端
axios.defaults.withCredentials = true; //配置为true

页面编码和被请求的资源编码如果不一致如何处理?

对于ajax请求传递的参数，如果是get请求方式，参数如果传递中文，在有些浏览器会乱码，不同的浏览器对参数编码的处理方式不同，所以对于get请求的参数需要使用 encodeURIComponent函数对参数进行编码处理，后台开发语言都有相应的解码api。对于post请求不需要进行编码

另外一种：

a.html 的编码是gbk或gb2312的。 而引入的js编码为utf-8的 ，那就需要在引入的时候

<script src="http://www.xxx.com/test.js" charset="utf-8"></script>

同理，如果你的页面是utf-8的，引入的js是gbk的，那么就需要加上charset="gbk".

服务器代理转发时，该如何处理cookie?

nginx

---

nginx模块化开发怎么做?

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
        };
    })();
```

（待完善）

AMD（Modules/Asynchronous-Definition）、CMD（Common Module Definition）规范区别?

AMD 规范在这里：https://github.com/amdjs/amdjs-api/wiki/AMD

CMD 规范在这里：https://github.com/seajs/seajs/issues/242

Asynchronous Module Definition，异步模块定义，所有的模块将被异步加载，模块加载不影响后面语句运行。所有依赖某些模块的语句均放置在回调函数中。

区别：

1. 对于依赖的模块，AMD 是提前执行，CMD 是延迟执行。不过 RequireJS 从 2.0 开始，也改成可以延迟执行（根据写法不同，处理方式不同）。CMD 推崇 as lazy as possible.
2. CMD 推崇依赖就近，AMD 推崇依赖前置。看代码：

```
// CMD
define(function(require, exports, module) {
    var a = require('./a')
    a.doSomething()
    // 此处略去 100 行
    var b = require('./b') // 依赖可以就近书写
    b.doSomething()
```

```
    // ...
})

// AMD 默认推荐
define(['./a', './b'], function(a, b) { // 依赖必须一开始就写好
    a.doSomething()
    // 此处略去 100 行
    b.doSomething()
    // ...
})
```

requireJS的核心原理是什么？（如何动态加载的？如何避免多次加载的？如何缓存的？）

谈一谈你对ECMAScript6的了解?

ECMAScript6 怎么写class么，为什么会出现class这种东西?

异步加载JS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

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

DOM操作——怎样添加、移除、移动、复制、创建和查找节点?

（1）创建新节点
createDocumentFragment()　//创建一个DOM片段
createElement()　//创建一个具体的元素
createTextNode()　//创建一个文本节点
（2）添加、移除、替换、插入
appendChild()
removeChild()
replaceChild()
insertBefore() //在已有的子节点前插入一个新的子节点
（3）查找
getElementsByTagName()　//通过标签名称
getElementsByName()　//通过元素的Name属性的值(IE容错能力较强，会得到一个数组，其中包括id等于name值的)
getElementById()　//通过元素Id，唯一性

.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);
```

数组和对象有哪些原生方法，列举一下?

JS 怎么实现一个类。怎么实例化这个类

JavaScript中的作用域与变量声明提升?

如何编写高性能的Javascript?

那些操作会造成内存泄漏?

jQuery的源码看过吗？能不能简单概况一下它的实现原理?

jQuery.fn的init方法返回的this指的是什么对象？为什么要返回this?

jquery中如何将数组转化为json字符串，然后再转化回来?

jQuery 的属性拷贝(extend)的实现原理是什么，如何实现深拷贝?

jquery.extend 与 jquery.fn.extend的区别?

jQuery 的队列是如何实现的？队列可以用在哪些地方?

谈一下Jquery中的bind(),live(),delegate(),on()的区别?

JQuery一个对象可以同时绑定多个事件，这是如何实现的?

是否知道自定义事件。jQuery里的fire函数是什么意思，什么时候用?

jQuery 是通过哪个方法和 Sizzle 选择器结合的？ （jQuery.fn.find()进入Sizzle）

针对 jQuery性能的优化方法?

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：

```
$.fn.stringifyArray = function(array) {
    return JSON.stringify(array)
}

$.fn.parseArray = function(array) {
    return JSON.parse(array)
}
```

然后调用：

```
$("").stringifyArray(array)
```

jQuery和Zepto的区别？各自的使用场景？

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++) {}

Zepto的点透问题如何解决？

jQueryUI如何自定义组件？

需求：实现一个页面操作不会整页刷新的网站，并且能在浏览器前进、后退时正确响应。给出你的技术实现方案？

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
移动端最小触控区域是多大？

jQuery 的 slideUp动画 ，如果目标元素是被外部事件驱动, 当鼠标快速地连续触发外部元素事件, 动画会滞后的反复执行，该如何处理呢？

把 Script 标签 放在页面的最底部的body封闭之前 和封闭之后有什么区别？浏览器会如何解析它们？

移动端的点击事件的有延迟，时间是多久，为什么会有？ 怎么解决这个延时？（click 有 300ms 延迟,为了实现safari的双击事件的设计，浏览器要知道你是不是要双击操作。）

知道各种JS框架(Angular, Backbone, Ember, React, Meteor, Knockout...)么? 能讲出他们各自的优点和缺点么？

Underscore 对哪些 JS 原生对象进行了扩展以及提供了哪些好用的函数方法？

解释JavaScript中的作用域与变量声明提升？

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）
JQuery一个对象可以同时绑定多个事件，这是如何实现的?

Node.js的适用场景?

(如果会用node)知道route, middleware, cluster, nodemon, pm2, server-side rendering么?

解释一下 Backbone 的 MVC 实现方式?

什么是“前端路由”?什么时候适合使用“前端路由”？“前端路由”有哪些优点和缺点?

知道什么是webkit么? 知道怎么用浏览器的各种工具来调试和debug代码么?

如何测试前端代码么? 知道BDD, TDD, Unit Test么? 知道怎么测试你的前端工程么(mocha, sinon, jasmin, qUnit..)?

前端templating(Mustache, underscore, handlebars)是干嘛的, 怎么用?

简述一下 Handlebars 的基本用法?

简述一下 Handlerbars 的对模板的基本处理流程， 如何编译的？如何缓存的?

用js实现千位分隔符?(来源：前端农民工，提示：正则+replace)

```
function commafy(num) {
    num = num + '';
    var reg = /(-?d+)(d{3})/;

   if(reg.test(num)){
    num = num.replace(reg, '$1,$2');
   }
   return num;
}
```

检测浏览器版本版本有哪些方式?

功能检测、userAgent特征检测

比如：navigator.userAgent
//"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_2) AppleWebKit/537.36
（KHTML, like Gecko) Chrome/41.0.2272.101 Safari/537.36"
What is a Polyfill?

polyfill 是“在旧版浏览器上复制标准 API 的 JavaScript 补充”,可以动态地加载 JavaScript 代码或库，在不支持这些标准 API 的浏览器中模拟它们。
例如，geolocation（地理位置）polyfill 可以在 navigator 对象上添加全局的 geolocation 对象，还能添加 getCurrentPosition 函数以及“坐标”回调对象，
所有这些都是 W3C 地理位置 API 定义的对象和函数。因为 polyfill 模拟标准 API，所以能够以一种面向所有浏览器未来的方式针对这些 API 进行开发，
一旦对这些 API 的支持变成绝对大多数，则可以方便地去掉 polyfill，无需做任何额外工作。
做的项目中，有没有用过或自己实现一些 polyfill 方案（兼容性处理方案）？

比如： html5shiv、Geolocation、Placeholder

我们给一个dom同时绑定两个点击事件，一个用捕获，一个用冒泡。会执行几次事件，会先执行冒泡还是捕获?

ECMAScript6 相关
Object.is() 与原来的比较操作符“===”、“==”的区别?

两等号判等，会在比较时进行类型转换;
三等号判等(判断严格)，比较时不进行隐式类型转换，(类型不同则会返回false)；

Object.is 在三等号判等的基础上特别处理了 NaN 、-0 和 +0 ，保证 -0 和 +0 不再相同，
但 Object.is(NaN, NaN) 会返回 true.

Object.is 应被认为有其特殊的用途，而不能用它认为它比其它的相等对比更宽松或严格。
前端框架相关
react-router 路由系统的实现原理?

React中如何解决第三方类库的问题?

其他问题
原来公司工作流程是怎么样的，如何与其他人协作的?如何夸部门合作的?

你遇到过比较难的技术问题是?你是如何解决的?

设计模式 知道什么是singleton, factory, strategy, decrator么?

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?

页面重构怎么操作?

网站重构：在不改变外部行为的前提下，简化结构、添加可读性，而在网站前端保持一致的行为。
也就是说是在不改变UI的情况下，对网站进行优化，在扩展的同时保持一致的UI。

对于传统的网站来说重构通常是:

表格(table)布局改为DIV+CSS
使网站前端兼容于现代浏览器(针对于不合规范的CSS、如对IE6有效的)
对于移动平台的优化
针对于SEO进行优化
深层次的网站重构应该考虑的方面

减少代码间的耦合
让代码保持弹性
严格按规范编写代码
设计可扩展的API
代替旧有的框架、语言(如VB)
增强用户体验
通常来说对于速度的优化也包含在重构中

压缩JS、CSS、image等前端资源(通常是由服务器来解决)
程序的性能优化(如数据读写)
采用CDN来加速资源加载
对于JS DOM的优化
HTTP服务器的文件缓存

列举IE与其他浏览器不一样的特性?

1、事件不同之处:

触发事件的元素被认为是目标（target）。而在 IE 中，目标包含在 event 对象的 srcElement 属性;

获取字符代码、如果按键代表一个字符（shift、ctrl、alt除外），IE 的 keyCode 会返回字符代码（Unicode），DOM 中按键的代码和字符是分离的，要获取字符代码，需要使用 charCode 属性;

阻止某个事件的默认行为，IE 中阻止某个事件的默认行为，必须将 returnValue 属性设置为 false，Mozilla 中，需要调用 preventDefault() 方法;

停止事件冒泡，IE 中阻止事件进一步冒泡，需要设置 cancelBubble 为 true，Mozzilla 中，需要调用 stopPropagation();
99%的网站都需要被重构是那本书上写的?

网站重构：应用web标准进行设计（第2版）
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，则代码会针对旧版本的IE进行降级处理了,使之在旧式浏览器上以某种形式降级体验却不至于完全不能用。
如：border-shadow

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版本浏览器才支持的功能,向页面增加不影响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。
如：默认使用flash上传，但如果浏览器支持 HTML5 的文件上传功能，则使用HTML5实现更好的体验;
是否了解公钥加密和私钥加密。

一般情况下是指私钥用于对数据进行签名，公钥用于对签名进行验证;
HTTP网站在浏览器端用公钥加密敏感数据，然后在服务器端再用私钥解密。
WEB应用从服务器主动推送Data到客户端有那些方式?

html5提供的Websocket
不可见的iframe
WebSocket通过Flash
XHR长时间连接
XHR Multipart Streaming
<script>标签的长时间连接(可跨域)
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。
你有用过哪些前端性能优化的方法?

（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。

对普通的网站有一个统一的思路，就是尽量向前端优化、减少数据库操作、减少磁盘IO。向前端优化指的是，在不影响功能和体验的情况下，能在浏览器执行的不要在服务端执行，能在缓存服务器上直接返回的不要到应用服务器，程序能直接取得的结果不要到外部取得，本机内能取得的数据不要到远程取，内存能取到的不要到磁盘取，缓存中有的不要去数据库查询。减少数据库操作指减少更新次数、缓存结果减少查询次数、将数据库执行的操作尽可能的让你的程序完成（例如join查询），减少磁盘IO指尽量不使用文件系统作为缓存、减少读写文件次数等。程序优化永远要优化慢的部分，换语言是无法“优化”的。

http状态码有那些？分别代表是什么意思？

简单版

[

100 Continue 继续，一般在发送post请求时，已发送了http header之后服务端将返回此信息，表示确认，之后发送具体参数信息

200 OK 正常返回信息

201 Created 请求成功并且服务器创建了新的资源

202 Accepted 服务器已接受请求，但尚未处理

301 Moved Permanently 请求的网页已永久移动到新位置。

302 Found 临时性重定向。

303 See Other 临时性重定向，且总是使用 GET 请求新的 URI。

304 Not Modified 自从上次请求后，请求的网页未修改过。

400 Bad Request 服务器无法理解请求的格式，客户端不应当尝试再次使用相同的内容发起请求。

401 Unauthorized 请求未授权。

403 Forbidden 禁止访问。

404 Not Found 找不到如何与 URI 相匹配的资源。

500 Internal Server Error 最常见的服务器端错误。

503 Service Unavailable 服务器端暂时无法处理请求（可能是过载或维护）。

]

完整版

1**(信息类)：表示接收到请求并且继续处理

100——客户必须继续发出请求

101——客户要求服务器根据请求转换HTTP协议版本

2**(响应成功)：表示动作被成功接收、理解和接受

200——表明该请求被成功地完成，所请求的资源发送回客户端

201——提示知道新文件的URL

202——接受和处理、但处理未完成

203——返回信息不确定或不完整

204——请求收到，但返回信息为空

205——服务器完成了请求，用户代理必须复位当前已经浏览过的文件

206——服务器已经完成了部分用户的GET请求

3**(重定向类)：为了完成指定的动作，必须接受进一步处理

300——请求的资源可在多处得到

301——本网页被永久性转移到另一个URL

302——请求的网页被转移到一个新的地址，但客户访问仍继续通过原始URL地址，重定向，新的URL会在response中的Location中返回，浏览器将会使用新的URL发出新的Request。

303——建议客户访问其他URL或访问方式

304——自从上次请求后，请求的网页未修改过，服务器返回此响应时，不会返回网页内容，代表上次的文档已经被缓存了，还可以继续使用

305——请求的资源必须从服务器指定的地址得到

306——前一版本HTTP中使用的代码，现行版本中不再使用

307——申明请求的资源临时性删除

4**(客户端错误类)：请求包含错误语法或不能正确执行

400——客户端请求有语法错误，不能被服务器所理解

401——请求未经授权，这个状态代码必须和WWW-Authenticate报头域一起使用

HTTP 401.1 - 未授权：登录失败

HTTP 401.2 - 未授权：服务器配置问题导致登录失败

HTTP 401.3 - ACL 禁止访问资源

HTTP 401.4 - 未授权：授权被筛选器拒绝

HTTP 401.5 - 未授权：ISAPI 或 CGI 授权失败

402——保留有效ChargeTo头响应

403——禁止访问，服务器收到请求，但是拒绝提供服务

HTTP 403.1 禁止访问：禁止可执行访问

HTTP 403.2 - 禁止访问：禁止读访问

HTTP 403.3 - 禁止访问：禁止写访问

HTTP 403.4 - 禁止访问：要求 SSL

HTTP 403.5 - 禁止访问：要求 SSL 128

HTTP 403.6 - 禁止访问：IP 地址被拒绝

HTTP 403.7 - 禁止访问：要求客户证书

HTTP 403.8 - 禁止访问：禁止站点访问

HTTP 403.9 - 禁止访问：连接的用户过多

HTTP 403.10 - 禁止访问：配置无效

HTTP 403.11 - 禁止访问：密码更改

HTTP 403.12 - 禁止访问：映射器拒绝访问

HTTP 403.13 - 禁止访问：客户证书已被吊销

HTTP 403.15 - 禁止访问：客户访问许可过多

HTTP 403.16 - 禁止访问：客户证书不可信或者无效

HTTP 403.17 - 禁止访问：客户证书已经到期或者尚未生效

404——一个404错误表明可连接服务器，但服务器无法取得所请求的网页，请求资源不存在。eg：输入了错误的URL

405——用户在Request-Line字段定义的方法不允许

406——根据用户发送的Accept拖，请求资源不可访问

407——类似401，用户必须首先在代理服务器上得到授权

408——客户端没有在用户指定的饿时间内完成请求

409——对当前资源状态，请求不能完成

410——服务器上不再有此资源且无进一步的参考地址

411——服务器拒绝用户定义的Content-Length属性请求

412——一个或多个请求头字段在当前请求中错误

413——请求的资源大于服务器允许的大小

414——请求的资源URL长于服务器允许的长度

415——请求资源不支持请求项目格式

416——请求中包含Range请求头字段，在当前请求资源范围内没有range指示值，请求也不包含If-Range请求头字段
417——服务器不满足请求Expect头字段指定的期望值，如果是代理服务器，可能是下一级服务器不能满足请求长。

5**(服务端错误类)：服务器不能正确执行一个正确的请求
HTTP 500 - 服务器遇到错误，无法完成请求
HTTP 500.100 - 内部服务器错误 - ASP 错误
HTTP 500-11 服务器关闭
HTTP 500-12 应用程序重新启动
HTTP 500-13 - 服务器太忙
HTTP 500-14 - 应用程序无效
HTTP 500-15 - 不允许请求 global.asa
Error 501 - 未实现
HTTP 502 - 网关错误
HTTP 503：由于超载或停机维护，服务器目前无法使用，一段时间后可能恢复正常
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）

注：这题胜在区分度高，知识点覆盖广，再不懂的人，也能答出几句，
而高手可以根据自己擅长的领域自由发挥，从URL规范、HTTP协议、DNS、CDN、数据库查询、
到浏览器流式解析、CSS规则构建、layout、paint、onload/domready、JS执行、JS API绑定等等；

详细版：
1、浏览器会开启一个线程来处理这个请求，对 URL 分析判断如果是 http 协议就按照 Web 方式来处理;
2、调用浏览器内核中的对应方法，比如 WebView 中的 loadUrl 方法;
3、通过DNS解析获取网址的IP地址，设置 UA 等信息发出第二个GET请求;
4、进行HTTP协议会话，客户端发送报头(请求报头);
5、进入到web服务器上的 Web Server，如 Apache、Tomcat、Node.JS 等服务器;
6、进入部署好的后端应用，如 PHP、Java、JavaScript、Python 等，找到对应的请求处理;
7、处理结束回馈报头，此处如果浏览器访问过，缓存上有对应资源，会与服务器最后修改时间对比，一致则返回304;
8、浏览器开始下载html文档(响应报头，状态码200)，同时使用缓存;
9、文档树建立，根据标记请求所需指定MIME类型的文件（比如css、js）,同时设置了cookie;
10、页面开始渲染DOM，JS根据DOM API操作DOM,执行事件绑定等，页面显示完成。

简洁版：
浏览器根据请求的URL交给DNS域名解析，找到真实IP，向服务器发起请求；
服务器交给后台处理完成后返回数据，浏览器接收文件（HTML、JS、CSS、图象等）；
浏览器对加载到的资源（HTML、JS、CSS等）进行语法解析，建立相应的内部数据结构（如HTML的DOM）；
载入解析到的资源文件，渲染页面，完成。
部分地区用户反应网站很卡，请问有哪些可能性的原因，以及解决方法?

从打开app到刷新出内容，整个过程中都发生了什么，如果感觉慢，怎么定位问题，怎么解决?

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?

你用的得心应手用的熟练地编辑器&开发环境是什么样子?

Sublime Text 3 + 相关插件编写前端代码
Google chrome 、Mozilla Firefox浏览器 +firebug 兼容测试和预览页面UI、动画效果和交互功能
Node.js+Gulp
git 用于版本控制和Code Review
对前端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。
你怎么看待Web App 、hybrid App、Native App?

你移动端前端开发的理解？（和 Web 前端开发的主要区别是什么？）

你对加班的看法?

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
如何设计突发大规模并发架构?

当团队人手不足，把功能代码写完已经需要加班的情况下，你会做前端代码的测试吗?

说说最近最流行的一些东西吧？常去哪些网站?

ES6\WebAssembly\Node\MVVM\Web Components\React\React Native\Webpack 组件化
知道什么是SEO并且怎么优化么? 知道各种meta data的含义么?

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、
简单描述一下你做过的移动APP项目研发流程?

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?

你有自己的技术博客吗，用了哪些技术?

对前端安全有什么看法?

是否了解Web注入攻击，说下原理，最常见的两种攻击（XSS 和 CSRF）了解到什么程度?

项目中遇到国哪些印象深刻的技术难题，具体是什么问题，怎么解决？。

最近在学什么东西?

你的优点是什么？缺点是什么?

如何管理前端团队?

最近在学什么？能谈谈你未来3，5年给自己的规划吗?

前端学习网站推荐
1. 极客标签：　http://www.gbtags.com/

2. 码农周刊：　http://weekly.manong.io/issues/

3. 前端周刊：　http://www.feweekly.com/issues

4. 慕课网：　http://www.imooc.com/

5. div.io：　http://div.io

6. Hacker News：　https://news.ycombinator.com/news

7. InfoQ：　http://www.infoq.com/

8. w3cplus：　http://www.w3cplus.com/

9. Stack Overflow：　http://stackoverflow.com/

10.w3school：　http://www.w3school.com.cn/

11.mozilla：　https://developer.mozilla.org/zh-CN/docs/Web/JavaScript

以上内容摘抄于：

https://www.cnblogs.com/king18181753985/p/6510756.htm