HTML

1. 谈谈你对 Web 标准的理解。

WEB 标准不是某一个标准，而是一系列标准的集合。

网页主要由三部分组成：

1．结构（Structure）结构化标准语言主要包括 XHTML 和 XML

2．表现（Presentation）表现标准语言主要包括 CSS

3．行为（Behavior）

行为标准主要包括对象模型（如 W3C DOM）、ECMAScript

并要求这三部分分离。

这些标准大部分由 W3C 起草和发布，也有一些是其他标准组织制订的标准，比如 ECMA（European Computer Manufacturers Association）的 ECMAScript 标准。

2. 浏览器标准模式和怪异模式之间的区别是什么？

盒子模型 渲染模式的不同

3. <img>标签上 title 与 alt 属性的区别是什么？

Alt 当图片不显示是 用文字代表。

Title 为该属性提供信息

4. 你如何对网站的文件和资源进行优化？

文件合并
文件最小化/文件压缩
使用 CDN 托管
缓存的使用

5.前端页面有哪三层构成，分别是什么？作用是什么？

结构层 Html 表示层 CSS 行为层 js

6.什么是语义化的 HTML？

直观的认识标签　对于搜索引擎的抓取有好处

7.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：

1、普通超级链接，语法为：文本

2、下载链接，即目标文档为下载资源，语法如：
　　下载

3、电子邮件链接，用于链接到 email，语法如：

联系我们

4、空链接，用于返回页面顶部，语法如：...

5、链接到 JavaScript，以实现特定的代码功能，语法如：

JS 功能

8.列举常用的结构标记，并描述其作用。

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

常用的结构标记有：

<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。

9.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，则当用户点击该标签（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。

10.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。当定义了锚点后，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>
```

11.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？

常见的浏览器内核有：

Trident：IE 浏览器；

Gecko：Mozilla 浏览器，比如 Firefox；

Webkit：Safari 浏览器，也是 Chrome 浏览器的内核原型；

Blink：Chrome 浏览器，Opera 浏览器。

12.写出几种 IE6 BUG 的解决方法

1.双边距 BUG float 引起的　使用 display

2.像素问题　使用 float 引起的　使用 dislpay:inline -3px

3.超链接 hover　点击后失效　使用正确的书写顺序 link visited hover active

4.Ie z-index 问题　给父级添加 position:relative

5.Png 透明 使用 js 代码 改

6.Min-height 最小高度 ！ Important 解决’
7.select 在 ie6 下遮盖 使用 iframe 嵌套

8.为什么没有办法定义 1px 左右的宽度容器（IE6 默认的行高造成的，使用 over:hidden,zoom:0.08 line-height:1px）

13.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。

14.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> <img> <input> <link> <meta>

鲜为人知的是：

<area> <base> <col> <command> <embed> <keygen> <param> <source> <track> <wbr>

15.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.

16.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；

17.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 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 {}
```

18.xhtml 和 html 有什么区别

HTML 是一种基本的 WEB 网页设计语言，XHTML 是一个基于 XML 的置标语言

最主要的不同：

XHTML 元素必须被正确地嵌套。

XHTML 元素必须被关闭。

标签名必须用小写字母。

XHTML 文档必须拥有根元素。

19.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；

* 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

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

20.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。

21.HTML5 新的 DocType 和 Charset 是什么？

```
<! DOCTYPE html>
<meta charset=”UTF-8”>
```

22.如何在 HTML5 页面中嵌入音频?

```
<audio controls=”controls”>
<source src=”...” type=”audio/mpeg”>audio
</audio>
```

23.如何在 HTML5 页面中嵌入视频？

```
<video width=550 height=550 controls=”controls”>
```

```
<source src=”...” type=”video/mp4”>video
</video>
```

24.除了音频和视频，HTML5 还支持其他什么新的媒体元素

```
<embed> 作为对外部应对容器
<track>为媒介规定外部文本轨道
<source>
```

25.HTML5 有哪些不同类型的存储，区别是什么？

HTML5 能够本地存储数据，在之前都是使用 cookies 使用的。HTML5 提供了下面两种本地存储方案：

localStorage 用于持久化的本地存储，数据永远不会过期，关闭浏览器也不会丢失。

sessionStorage 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

26.HTML5 引入什么新的表单属性？

```
Datalist
Datatime
Output
Keygen
Date
Month
Week
Time
Number
Range
Email
Url
```

27.HTML5 标准提供了哪些新的 API？

```
Media api
History api
```

28.什么是 HTML5？

HTML5 是最新的 html 标准，它主要目标是提供所有内容而不需要任何额外插件。

29.没有<!DOCTYPE HTML>,HTML5 还会工作吗？

不会，浏览器不能识别它是 HTML 文档；

30.哪些浏览器支持 HTML5?

所有浏览器都支持 HTML5；

31.HTML5 的页面结构

<header>代表 HTML 的头部数据
<nav>页面导航元素
<article>自包含的内容
<section>把分组内容放到区域里
<aside>代表页面的侧边栏内容
<footer>代表页面的脚部区域

32.HTML5 中的 datalist 是什么？

HTML5 中的 Datalist 元素有助于提供文本框自动完成特性

33.HTML5 中心的表单元素是什么？

```
<input type=”color”/>
<input type=”date” />
<input type=”datetime-local” />
<input type=”email” />含有 email 校验的 HTML 文本框;
<input type=”url” />url 校验;
<input type=”number” min=”1” max=”5” />
<input type=”range” min=”1” max=”5” step=”2” value=”3” />range (范围) step(步骤)  //step 指的是拉动的距离，value 指的是初始值;
<input type=”search” />搜索引擎;
<input type=”time” />只输入时间;
<input type=”tel” />只输入电话;
```

34.HTML5 中什么是输出元素?

```
Output 当你需要输出值的时候就是输出元素;
```

35.什么 SVG?

```
SVG 表示可缩放矢量图形;（跟 canvas 差不多）
```

36.HTML5 中 canvas 是什么?

```
Canvas 元素用于在网页上绘制图形，该元素标签强大之处在于可以直接在 HTML 上进行图形操作
```

37.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。
```

38.HTML5 应用程序缓存和浏览器缓存有什么区别?

应用程序缓存是 HTML5 的重要特性之一，提供了离线使用的功能，让应用程序可以获取本地的网站内容，例如 HTML、CSS、图片以及 JavaScript。这个特性可以提高网站性能，它的实现借助于 manifest 文件，如下：

?

```
<!doctype html>
<html manifest=”example.appcache”>
…..
</html>
```

与传统浏览器缓存相比，它不强制用户访问的网站内容被缓存。

39.(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>
```

40.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。

41.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；

42.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

43.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

44.为什么 HTML5 里面我们不需要 DTD（Document Type Definition 文档类型定义）？

在 HTML 4.01 中，<!DOCTYPE> 声明引用 DTD，因为 HTML 4.01 基于 SGML。DTD 规定了标记语言的规则，这样浏览器才能正确地呈现内容。

HTML5 不基于 SGML，所以不需要引用 DTD，因此没有声明 DTD。

45.Canvas 和 SVG 图形之间的区别是什么？

Canvas

依赖分辨率

不支持事件处理器

弱的文本渲染能力

能够以 .png 或 .jpg 格式保存结果图像

最适合图像密集型的游戏，其中的许多对象会被频繁重绘

SVG

不依赖分辨率

支持事件处理器

最适合带有大型渲染区域的应用程序（比如谷歌地图）

复杂度高会减慢渲染速度（任何过度使用 DOM 的应用都不快）

不适合游戏应用

46.如何使用 Canvas 和 HTML5 中的 SVG 去画一个矩形？

自己复习 canvas 和 svg

47.简要描述 HTML5 中的本地存储

参考答案：

很多时候我们会存储用户本地信息到电脑上，例如：比方说用户有一个填充了一半的长表格，然后突然网络连接断开了，这样用户希望你能存储这些信息到本地，当网络恢复的时候，他想获取这些信息然后发送到服务器进行存储。

现代浏览器拥有的存储被叫做“Local Storage”，用于存储这些信息。

48.HTML5 应用缓存是什么

参考答案：

常用于实现用户的离线浏览。如果网络连接不可用，页面应该来自浏览器缓存，离线应用缓存可以帮助你达到这个目的。

应用缓存可以帮助你指定哪些文件需要缓存，哪些不需要。

49.什么是 Web Worker？为什么我们需要他们？

参考答案：

查看如下代码（模拟会执行上百万次的繁重代码）：

```
function test(){
 for(i=0;i< 10000000000 ; i ++){
     x = x + i;
 }
}
```

如果上述代码在 HTML 按钮点击以后执行，这种执行是同步的，即，浏览器必须等到此执行完毕之后才能进行其他操作。因为此操作耗时较长，那么这个操作会导致浏览器冻结并且没有响应，而且屏幕还会出现异常信息。

如果可以将这些繁重的代码移动到 Javascript 文件中，并采用异步的方式运行，就可以解决这个问题。这就是 webworker 的作用。Web Worker 用于异步执行 JavaScript 文件，提高浏览器的敏捷度。

CSS

50.谈谈你对浏览器兼容性问题的理解

浏览器的类型及版本的不同会造成 CSS 效果不尽相同，因此需要实现浏览器兼容，也可以针对不同的浏览器编写不同的 CSS。

目前，各主流浏览器的新版本，对于 W3C 的标准支持很好，因此，首先保证代码符合 W3C 的标准，这是解决浏览器兼容问题的前提。

其次，对于某些支持受限的属性，针对不同的浏览器添加相应的前缀，比如-webkit-、-o-、-moz-。

第三，对于 IE 的低版本，可以编写带有特定前缀的代码，实现版本识别。比如：

```
.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*/
}
```

另外，对于特定的兼容性问题，特殊解决。常见的特殊问题有：

1、使用 CSS reset：对于有些 HTML 标签，浏览器默认的 margin 和 padding 不同，可以使用 CSS 代码改写默认的样式效果，从而实现统一；

2、IE 低版本中，不能使用 auto 关键字实现块级元素居中显示，可以改用设置父元素的 text-align；

3、子元素设置上外边距时，父元素需要设置边框或者外边距；

4、外边距合并问题。

51.为什么建议设置背景图像的同时还设置背景颜色？

一般建议在使用背景图像的同时提供 background-color 属性，并且将其设置为和图像主要颜色类似的颜色。这样，如果正在加载页面，或者因为各种原因无法显示背景图像时，页面可以使用这种颜色作为背景色。

52.内联元素可以实现浮动吗？

在 CSS 中，任何元素都可以浮动。浮动元素会生成一个块级框，而不论它本身是何种元素。因此，对于内联元素，如果设置为浮动，会产生和块级框相同的效果。

53.什么情况下需要额外设置表格的显示规则？

默认情况下（不额外设置表格的显示规则时），表格按照自动表格布局进行显示，即，浏览器在显示表之前查看每一个单元格，然后基于所有单元格的设置计算表的大小，而列的宽度是由列单元格中没有折行的最宽的内容设定的。此时，单元格的大小会适应内容的大小。

自动表格布局的算法在表格复杂时会比较慢，这是由于它需要在确定最终的布局之前访问表格中所有的内容。在不能提前确定每一列的大小时，这种方式会非常适用。

如果额外设置表格的显示规则，即，设置 table-layout 属性的值为 fixed，则称为固定表格布局。在固定表格布局中，水平布局仅取决于表格宽度、列宽度、表格边框宽度、单元格间距，而与单元格的内容无关。浏览器将使用某列指定的宽度来计算布局（如果给了宽度的话），并使用该宽度计算该列中所有其他单元格的宽度。

固定表格布局与自动表格布局相比，允许浏览器更快地对表格进行布局。因为如果指定使用固定表格布局，浏览器在接收到第一行后就可以显示表格。如果表格庞大且已经指定了大小，则会加速表的显示。

54.简要描述 CSS 中 content 属性的作用。

content 属性与:before 及:after 伪元素配合使用，来插入生成内容，可以在元素之前或之后放置生成的内容。可以插入文本、图像、引号，并可以结合计数器为页面元素插入编号。比如，查看如下代码：

```
body {counter-reset:chapter;}

h1:before {content:“第”counter(chapter)"章";}

h1 {counter-increment:chapter;}
```

使用 content 属性，并结合:before 选择器和计数器 counter，可以在每个<h1> 元素前插入新的内容。

55.在设置文本的字体时，为什么建议设置替换字体？

可以使用 font-family 属性来指定文本的字体，代码如下所示： font-family:name/inherit;

此时，name 为首选字体的名称。如果字体名称有多个单词，即中间有空格，则需要将字体名称用一对单引号或者双引号包围起来。

但是，如果用户机器上并没有安装 name 所指定的字体，则会显示默认字体。因此，如果可以指定一种替代字体，替代字体可以和指定字体不完全相同，相似且不会影响页面的布局，就可以解决问题了。

我们可以为 font-family 属性指定多种字体，且多种字体之间用逗号隔开，这样可以为页面指定一个字体列表。如果用户机器没有第一种字体，则浏览器会查找字体列表中的下一种字体作为替代字体显示。如果找遍了字体列表还是没有可以使用的字体，浏览器才会使用默认字体显示页面。代码如下所示：

```
h1{font-family: Georgia, serif;}
```

此时，如果用户机器上没有安装 Georgia，但安装了 Times 字体（serif

字体系列中的一种字体），浏览器就可能对<h1> 元素使用 Times。尽管

Times 与 Georgia 并不完全匹配，但至少足够接近。

因此，我们建议在所有 font-family 规则中都提供一个通用字体系列。这样就提供了一条后路，在用户机器无法提供与规则匹配的特定字体时，就可以选择一个通用字体作为替换。

56.简要描述 CSS 中的定位机制。

CSS 中，除了默认的流定位方式以外，还有如下几种定位机制：浮动定位、相对定位、绝对定位和固定定位。

浮动定位是指将元素排除在普通流之外，并且将它放置在包含框的左边或者右边，但是依旧位于包含框之内。

相对定位将元素相对于它在普通流中的位置进行定位。

绝对定位是指将元素的内容从普通流中完全移除，并且可以使用偏移属性来固定该元素的位置。

固定定位是指将元素的内容固定在页面的某个位置。

57.哪些属性可以继承？

CSS 中可以继承的属性如下：

1）文本相关属性：font-family、font-size、font-style、font-variant、font-weight、font、letter-spacing、line-height、text-align、text-indent、text-transform、word-spacing、color；

2）列表相关属性：list-style-image、list-style-position、list-style-type、list-style；

3）表格相关属性：border-collapse、border-spacing、caption-side、table-layoute；

4）其他属性：Cursor、visibility。

58.如何理解 CSS 样式表的层叠性？

CSS 使用层叠（Cascade）的原则来考虑继承、层叠次序和优先级等重要特征，从而判断相互冲突的规则中哪个规则应该起作用。

继承性是指，许多 CSS 的样式规则不但影响选择器所定义的元素，而且会被这些元素的后代继承。

层叠性是指，当一个 Web 页面使用多个样式表，多个样式表中的样式可层叠为一个。在多个样式表之间所定义的样式没有冲突的时候，浏览器会显示所有的样式。

优先级是指，当发生样式定义冲突时，浏览器首先会按照不同样式规则的优先级来应用样式。CSS 样式的优先级如下所示（其中数字 3 拥有最高的优先权）：

1. 浏览器缺省设置；

2. 外部样式表（.css 文件）或者内部样式表（位于<head>元素内部）；

3. 内联样式（作为某个元素的 style 属性的值）。同等优先级下，以最后定义的样式为准，important 比内联高。

59.cellpadding 与 cellspacing 有何区别？

cellpadding：代表单元格边框到内容之间的距离（留白）

cellspacing：代表单元格间边框，以及和 table 边框之间的距离

60.列举常用的 CSS 选择器

ID、元素、class、伪类、组、包含、子等

61.display 与 visibility 有何异同？

可以使用 display 属性定义建立布局时元素生成的显示框类型。

如果将 display 属性设置为 block，可以让行内元素（比如<a>

素）表现得像块级元素一样；

如果将 display 属性设置为 inline，可以让块级元素（比如<p> 元

素）表现得像内联元素一样；

可以通过把 display 属性设置为 none，让生成的元素根本没有框。

这样的话，该框及其所有内容就不再显示，不占用文档中的空间。

在 DIV 设计中，使用 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，也即是说它仍具有高度、宽度等属性值。

62.怎么在网页中实现绝对定位？

将 position 设置为：absolute

63.table-layout、border-collapse 有何用途？

table-layout：设置表格是否自动调整宽高

border-collapse：表格与单元格及单元格间的边框是否融合在一起。

64.链接标记 target 属性的_top、_parent、_blank、main、left、top 各有何用处?

_top：在顶层 window 中打开链接；

_parent：在包含当前 frame 的父 window 中打开链接

_blank：在新 window 中打开链接

main、left、top 是由 Adobe Dreamweaver 生成的主、左、上框架集的框架默认名。

65.CSS 引入的方式有哪些?

内联 内嵌 外链 导入

66.CSS 中的选择器是什么？

选择器就是你想去应用一个样式的时候去帮你选择元素

67.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).

68.描述 css reset 的作用和用途。

Reset 重置浏览器的 css 默认属性 浏览器的品种不同，样式不同，然后重置，让他们统一

69.解释 css sprites，如何使用。

Css 精灵 把一堆小的图片整合到一张大的图片上，减轻服务器对图片的请求数量

70.清除浮动的几种方式，各自的优缺点 如何清除浮动元素带来的影响

1.使用空标签清除浮动 clear:both（理论上能清楚任何标签，，，增加无意义的标签）

2.使用 overflow:auto（空标签元素清除浮动而不得不增加无意代码的弊端,,使用 zoom:1 用于兼容 IE）

3.是用 afert 伪元素清除浮动(用于非 IE 浏览器)

浮动定位是指将元素排除在普通流之外，并且将它放置在包含框的左边或者右边，但是依旧位于包含框之内。也就是说，浮动的框可以向左或向右移动，直到它的外边缘碰到包含框或另一个浮动框的边框为止。

由于浮动框不在文档的普通流中，所以元素浮动之后，其原有位置不再保留，其他元素的位置会受到影响。

如果需要清除左侧或者右侧浮动元素带来的影响，则可以使用 clear 属性来设置。另外，包含框内的子元素浮动后，如果包含框没有设置具体的高度，则其高度会发生变化，此时，可以使用 overflow 属性来清除子元素浮动后带来的影响。

71.CSS 选择符有哪些？哪些属性可以继承？优先级算法如何计算？

* 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;
```

72.优先级为:

```
   !important >  id > class > tag

   important 比 内联优先级高
```

73.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。
```

74.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%;
}
```

75.列出 display 的值，说明他们的作用。position 的值， relative 和 absolute 定位原点是？

```
  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。

76.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

77.一个满屏 品 字布局 如何设计?

```
<!doctype>
<html>
    <head>
        <title>test</title>
    </head>
    <style>
    *{
        margin:0;
        padding: 0;
    }
    .main{
        width:100%;
        height: 60%;
    }
```

```
    .main .left, .main .right{
        width:50%;
        height: 100%;
        float:left;
        background: #a23;
    }
    .main .right{
        background: #e11;
    }
    .clear{
        clear:both;
    }
    .header{
        height:40%;
        background: #e33;
        width:100%;
    }
    </style>
    <body>
        <div class="header"></div>
        <div class="main">
            <div class="left"></div>
            <div class="right"></div>
            <div class="clear"></div>
        </div>

    </body>
</html>
```

78.经常遇到的 CSS 的兼容性有哪些？原因，解决方法是什么？

打开：http://www.cnblogs.com/lgmcolin/archive/2013/02/12/2910179.html

79.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; }
```

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

一个 block-level element ('display' 属性值为 'block', 'list-item' 或是 ‘table’) 会生成一个 block-level box，这样的盒子会参与到 block-formatting context (一种布局的方式) 中。
block formatting context
在这种布局方式下，盒子们自所在的 containing block 顶部起一个接一个垂直排列，水平方向上撑满整个宽度 (除非内部的盒子自己内部建立了新的 BFC)。
containing block
一般来说，盒子本身就为其子孙建立了 containing block，用来计算内部盒子的位置、大小，而对内部的盒子，具体采用哪个 containing block 来计算，需要分情况来讨论：

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；
否则则由这个祖先元素的 padding box 构成。
根元素所在的 containing block 被称为 initial containing block，在我们常用的浏览器环境下，指的是原点与 canvas 重合，大小和 viewport 相同的矩形；
对于 position 为 static 或 relative 的元素，其 containing block 为祖先元素中最近的 block container box 的 content box (除 margin, border, padding 外的区域)；
对于 position:fixed 的元素，其 containing block 由 viewport 建立；
对于 position:absolute 的元素，则是先找到其祖先元素中最近的 position 属性非 static 的元素，然后判断：
如果都找不到，则为 initial containing block。

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

这个题目应该能拆成几个点来回答：1、'display'、'position' 和 'float' 的相互关系；2、position 跟 display、overflow、float 下的 margin collapse。
一、'display'、'position' 和 'float' 的相互关系
首先我们先来看下这 3 个属性。
display 属性规定元素应该生成的框的类型。 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（display 还有很多其他值设置，读者自行查找）。
position 属性规定元素的定位类型。 absolute 表示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）。
Float 也是是一种布局方式，它定义元素在哪个方向浮动。以往这个属性总应用于图像，使文本围绕在图像周围，不过在 CSS 中，任何元素都可以浮动。浮动元素会生成一个块级框，而不论它本身是何种元素。 在布局过程中也经常会使用它来达到左右并排布局的效果。
那么这三种布局和框形成的关键特性(display)之间有什么关系呢，请看下面流程图：

转换对应表：
设定值 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 同设定值
总的来说，可以把它看作是一个类似优先级的机制， "position:absolute" 和 "position:fixed" 优先级最高，有它存在的时候，浮动不起作用，'display' 的值也需要调整； 其次，元素的 'float' 特性的值不是 "none" 的时候或者它是根元素的时候，调整 'display' 的值； 最后，非根元素，并且非浮动元素，并且非绝对定位的元素，'display' 特性值同设置值。
这从另一个侧面说明了一个问题:浮动或绝对定位的元素,只能是块元素或表格。
1. 'display' 的值为 'none'
如果 'display' 的值为 'none'，那么 'position' 和 'float' 不起作用。在这种情况下，元素不产生框。因此浮动和定位无效。
2. 'position' 的值是 'absolute' 或 'fixed'
否则，如果 'position' 的值是 'absolute' 或 'fixed'，框就是绝对定位的，'float' 计算后的值应该是 'none'，并且，'display' 会被按照上表设置。 框的位置将由 'top'，'right'，'bottom' 和 'left' 属性和该框的包含块确定。
也就是说，当元素是绝对定位时，浮动失效，'display' 会被按规则重置。
示例代码：
```
<script type="text/javascript">function getStyle(obj, style) {
var _style = (style == "float") ? "styleFloat" : style; return
document.defaultView ? document.defaultView.getComputedStyle(obj, null).getPropertyValue(style) : obj.currentStyle[_style.replace(/-[a-z]/g, function() {  return arguments[0].charAt(1).toUpperCase(); })];}window.onload = function() { document.getElementById("info").innerHTML = "float : " + getStyle(document.getElementById("test"), "float") +        "<br/>display : " + getStyle(document.getElementById("test"), "display");}</script><div id="test" style="position:absolute; float:left;display:inline;"></div><div id="info"></div>
```

上面代码中有一个既是绝对定位又是浮动的元素，以上代码可取出其 'display' 和 'float' 的计算值。
IE 中，'float' 值和 'display' 的特性值未发生变化，还是 "float: left; display: inline"。
其他浏览器中计算后的结果是："float: none; display: block"。
3. 'float' 的值不是 "none"

82.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。）

83.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

84.解释下浮动和它的工作原理？清除浮动的技巧

浮动的框可以左右移动，直至它的外边缘遇到包含框或者另一个浮动框的边缘。浮动框不属于文档中的普通流，当一个元素浮动之后，不会影响到 块级框的布局而只会影响内联框（通常是文本）的排列，文档中的普通流就会表现得和浮动框不存在一样，当浮动框高度超出包含框的时候，也就会出现包含框不会 自动伸高来闭合浮动元素（“高度塌陷”现象）。顾名思义，就是漂浮于普通流之上，像浮云一样，但是只能左右浮动。

```
clear:both;
overflow:hidden;
```

前后置内容生成

85.用过媒体查询，针对移动端的布局吗？

(1).根据媒体特性执行执行不同的 CSS 文件

例如：`<link media="screen and (min-width:768px) and (max-width:991px)" rel="" href="">`

(2). 在某段 CSS 样式声明中，只针对部分样式使用媒体查询

例如：

```
@media screen and (min-width:768px) and (max-width:991px)
{
选择器 {
样式设定....
}
}
```

86.使用 CSS 预处理器吗？喜欢那个？

Sass，less…

87.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

88.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

89.超链接点击过后 hover 样式就不出现的问题？

被点击访问过的超链接样式不再具有 hover 和 active 样式了,解决方法是改变 CSS 属性的排列顺序: L-V-H-A

90.IE6 的 margin 双倍边距 bug 问题

例如：

```
<style type=”text/css”>
body {margin:0;}
div {float:left; margin-left:10px; width:200px; height:200px; b
order:1px solid red;}
</style>
```

浮动后本来外边距 10px,但 IE 解释为 20px,解决办法是加上 display:inline;

91.为什么中火狐浏览器下文本无法撑开容器的高度？

标准浏览器中固定高度值的容器是不会象 IE6 里那样被撑开的,那我又想固定高度，又想能被撑开需要怎样设置呢？办法就是去掉 height 设置 min-height:200px; 这里为了照顾不认识 min-height 的 IE6 可以这样定义：

```
div { height:auto!important; height:200px; min-height:200px; }
```

92.过渡与动画的区别是什么？

过渡属性 transition 可以在一定的时间内实现元素的状态过渡为最终状态，用于模拟一种过渡动画效果，但是功能有限，只能用于制作简单的动画效果；而动画属性 animation 可以制作类似 Flash 动画，通过关键帧控制动画的每一步，控制更为精确，从而可以制作更为复杂的动画。

JavaScript

93.简要描述 JavaScript 中的作用域链

参考答案：

任何一段 JavaScript 代码都对应一个作用域链，作用域链中存放一系列对象，代码中声明的变量将作为对象的属性存放。

在 JavaScript 的顶层代码中，作用域链由一个全局对象组成；当定义一个函数时，它保存一个作用域链，作用域链上有两个对象，一个是函数对象，一个是全局对象。

每当一个函数被调用时，会创建一个活动对象（也叫上下文对象），函数中的局部变量将作为该对象的属性存放。

当需要使用一个变量时，将从作用域链中逐个查找对象的属性。比如：要使用变量 a，将先查找作用域中的第一个对象是否有属性 a，如果有就使用；如果没有就查找作用域链中

下一个对象的属性，以此类推。如果作用域链上没有任何一个对象含有属性 x，则认为这段代码的作用域链上不存在 x，将抛出引用错误异常。

当函数调用完成后，如果没有其他引用指向为此次调用所创建的上下文对象，该对象将被回收。

94.简述 JavaScript 中创建自定义对象的方式

参考答案：

自定义对象（user-defined object）指由用户创建的对象，兼容性问题需要由编写者注意。创建自定义对象的方式有：

1、对象直接量

2、new Object()

3、function 对象模板

4、Object.create()

95.JavaScript 中，this 关键字的作用是什么？

参考答案：

笼统的说，关键字 this 指向当前对象。比如，顶级代码中的 this 指向全局对象；在指定元素事件的时候，this 指定当前发生事件的元素对象。 对于嵌套函数，如果嵌套函数作为方法被调用，其 this 指向调用它的对象；如果作为函数调用，this 是全局对象或者为 undefined（严格模式下）。

96.简要描述 JavaScript 中的匿名函数

参考答案：

匿名函数是指在定义时没有指定名字的函数，且定义后往往直接调用。如：

```
function(num1, num2){
 console.log( num1 + num2 );
}
```

这种方式所定义的匿名函数，往往需要直接调用，如：

```
(function (num1, num2) {
   console.log(num1 + num2);
})(10,20);
```

匿名函数常用于定义不需要重复使用的函数，用完即释放。另外，对于直接调用的匿名函数而言，可以看成是一个临时的命名空间，其区域内定义的所有变量，不会污染到全局命名空间。

97.什么是正则表达式？在 JavaScript 中，如何应用正则表达式？

参考答案：

正则表达式(Regular Expression) 本身就是一个字符串，由一些普通字符和特殊字符组成的，用以描述一种特定的字符规则的表达式。

正则表达式常用于在一段文本中搜索、匹配或替换特定形式的文本。如：词语出现频率统计、验证字符串是否符合邮箱格式、屏蔽一篇帖子中的限制性词语等。许多程序设计语言都支持利用正则表达式进行字符串操作。

在 JavaScript 中，正则表达式的应用分为两种：

1、结合 String 对象的 replace、search 和 match 方法，实现对字符串的替换、查找和匹配；

2、定义正则表达式对象，实现对字符串的复杂匹配操作.

98.编写函数，实现对身份证号码最后一位的验证

二代身份证号码为 18 位，其最后一位（第 18 位）的计算方法为：

将前面的身份证号码 17 位数分别乘以不同的系数。从第一位到第十七位的系数分别为：

7－9－10－5－8－4－2－1－6－3－7－9－10－5－8－4－2

将这 17 位数字和系数相乘的结果相加

3、用加出来和除以 11，看余数是多少？

4、余数只可能有 0—1—2—3—4—5—6—7—8—9—10 这 11 个数字。每个数字所对应的最后一位身份证的号码为:1—0—X—9—8—7—6—5—4—3—2。即，如果余数是 2，就会在身份证的第 18 位数字上出现罗马数字的 X。如果余数是 10，身份证的最后一位号码就是 2。

例如：某男性的身份证号码是 34052419800101001X。验证其最后一位是否正确时，首先需要得出前 17 位的乘积和是 189，然后用 189 除以 11 得出的结果是 17+2/11，也就是说其余数是 2。最后通过对应规则就可以知道余数 2 对应的数字是 x。所以，可以判定此身份证号码的最后一位是合格的。

参考答案：

编写验证方法如下：

```
//验证方法
function verifyCode(id){
  if(id.length !=18 )
    return false;
  /*1、从第一位到第十七位的系数分别为：
        7,9,10,5,8,4,2,1,6,3,7,9,10,5,8,4,2
        将这 17 位数字和系数相乘的结果相加。    */
  var arr = [7,9,10,5,8,4,2,1,6,3,7,9,10,5,8,4,2];
  var sum = 0;
  for(var i=0; i<arr.length; i++){
    sum += parseInt(id.charAt(i)) * arr[i];
  }
  //2、用加出来和除以 11，看余数，
  var c = sum%11;
  //3、分别对应的最后一位身份证的号码为：1-0-X-9-8-7-6-5-4-3-2
  var ch = ['1', '0', 'X', '9', '8', '7', '6', '5', '4', '3', '2'];
  var code = ch[c];
  var last = id.charAt(17);
  last = last=='x' ? 'X': last;
  return last == code;
}
```

测试该方法：

99.编写函数，实现冒泡排序

参考答案：

使用 JavaScript 编写的冒泡排序函数如下所示：

```
function bubbleSort(arr) {
    for (var i = 0; i < arr.length; i++) {
        for (var j = 0; j < arr.length - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                var temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}
```

测试函数 bubbleSort，代码如下：

```
var arr = [12, 4, 9, 21, 43, 3];
bubbleSort(arr);
console.log(arr);
```

上述代码运行时，将输出排序后的结果：[3, 4, 9, 12, 21, 43]。

100.简要描述 JavaScript 中定义函数的几种方式

参考答案：

JavaScript 中，有三种定义函数的方式：

1、函数语句：即使用 function 关键字显式定义函数。如：

```
function f(x){
    return x+1;
}
```

2、函数定义表达式：也称为“函数直接量”。形如：

```
var f = function(x){return x+1;};
```

3、使用 Function() 构造函数定义，形如：

101.简述 arguments 对象的作用

参考答案：

在函数代码中，使用特殊对象 arguments 可以访问函数的参数。即，开发者在定义函数时，无需明确的为方法声明参数，也可以在方法体中使用 arguments 来访问参数。这是因为，arguments 是一种特殊对象，在函数代码中，表示函数的参数数组。

正因为 arguments 表示参数组成的数组，因此，首先可以使用 arguments.length 检测函数的参数个数，其次，可以通过下标（arguments[index]）来访问某个参数。这样，可以用 arguments 对象判断传递给函数的参数个数并获取参数，适用于函数参数无法确定个数的情况下。

102.什么是“逻辑短路”？

参考答案：

逻辑短路是对于逻辑运算而言，是指，仅计算逻辑表达式中的一部分便能确定结果，而不对整个表达式进行计算的现象。

对于“&&”运算符，当第一个操作数为 false 时，将不会判断第二个操作数，因为此时无论第二个操作数为何，最后的运算结果一定是 false；

对于“||”运算符，当第一个操作数为 true 时，将不会判断第二个操作数，因为此时无论第二个操作数为何，最后的运算结果一定是 true。

103.列举几个 JavaScript 中常用的全局函数，并描述其作用

参考答案

JavaScript 中常用的全局函数及其作用如下：

parseInt：解析一个字符串并返回一个整数；

2. parseFloat：解析一个字符串并返回一个浮点数；

3. isNaN：检查某个值是否是数字，返回 true 或者 false；

4. encodeURI ：把字符串作为 URI 进行编码；

5. decodeURI ：对 encodeURI() 函数编码过的 URI 进行解码；

6. eval：计算某个字符串，以得到结果，或者用于执行其中 JavaScript 代码。

104.解释一下 JavaScript 中的局部变量与全局变量的区别

参考答案：

全局变量拥有全局作用域，在 JavaScript 代码的任何地方都可以访问；在函数内声明的变量只在函数体内有定义，即为局部变量，其作用域是局部性的。

需要注意的是，在函数体内声明局部变量时，如果不使用 var 关键字，则将声明全局变量。

105.简要描述 JavaScript 的数据类型？

参考答案：

JavaScript 的数据类型可以分为原始类型和对象类型。

原始类型包括 string、number 和 boolean 三种。其中，字符串是使用一对单引号或者一对双引号括起来的任意文本；而数值类型都采用 64 位浮点格式存储，不区分整数和小数；布尔（逻辑）只能有两个值：true 或 false。

复杂类型指其他对象，如 Array、Date、Object 等。

除此之外，JavaScript 中还有两个特殊的原始值：null（空）和 undefined（未定义），它们代表了各自特殊类型的唯一成员。

106.如何阻止表单提交？

在 onsubmit 事件中返回 false

107.如何动态操作表格？

可以象普通 dom 一样操作，但是因为表格的 dom 比较复杂，所以我通常是使用 table 的 insertRow、deleteRow 及 tr 对象的 insetCell、deleteCell 操作。

108.String.match 与 RegExp.exec 有何区别？

match 只会返回没有分组的全部匹配结果或者有分组的第一次匹配结果；

而 exec 可以利用循环返回全部匹配结果。

109.正则的 i 标记与 g 标记各有何用途？

i：不区分大小写；g：全局匹配，如果没有此标记，正则不会第二个之后的结果。

110.为 String 添加 trim()方法。

```
String.prototype.trim = function() {
        return this.replace(/^ +| +$/g,"");
}
```

111.简述 COOKIE。在 JS 中如何操作 Cookie?

使用 document.cookie 操作。

112.谈谈 javascript 数组排序方法 sort()的使用，重点介绍 sort()参数的使用及其内部机制。

调用默认排序：[].sort();

调用自定义排序：

```
[].sort(function(){
    return 0; //返回 0，正数，负数分别代表与比较的数相等，比它大或小。
});
```

113.谈谈 innerHTML outerHTML innerText 之间的区别。

innerHTML 是 w3c 的 html dom 定义的方法，而后两者是 IE 独有的方法；

innerHTML 代表一个元素节点内由所有子节点，不包括当前节点组成的 html 代码；

outerHTML 代表一个元素节点内由所有子节点和当前节点组成的 html 代码；

innerText 代表一个元素节点内由所有子文本节点内容组成的文本；

114.谈谈 innerHTML、nodeValue 与 textContent 之间的区别

参考答案：

innerHTML 属性读取或设置节点起始和结束标签中的 HTML 内容；

nodeValue 属性读取或设置指定节点的文本内容，适用于文本类型的节点；

textContent 属性读取或设置指定节点的文本内容，对于元素节点而言，会返回所包含的所有子节点中的文本内容的组合。

115.你在 js 中用过 array 吗？如果用过，array 中添加数据用什么方法？

用过。

在尾部添加使用 push()；

在头部添加使用 unshift()；

在任意位置添加使用 splice()，但要注意把它的删除个数设置为 0。

116.简述 javascript 的优缺点。

优点：简单易用，与 Java 有类似的语法，可以使用任何文本编辑工具编写，

只需要浏览器就可执行程序，并且事先不用编译，逐行执行，无需进行严格的变量声明，而且内置大量现成对象，编写少量程序可以完成目标；

缺点：不适合开发大型应用程序；

117.Javascript 有哪些内置对象?

只有 Math 和 Global（在浏览器环境中，Global 就是 Window）

118.列举 Javascript 的本地对象。

Object、Function、Array、String、Boolean、Number、Date、RegExp、Error、EvalError

、RangeError、ReferenceError、SyntaxError、TypeError、URIError

119.什么是 Ajax？

客户端向服务器端发送请求,而无需刷新页面的技术

Ajax 包含的技术

* HTML、CSS、JavaScript、XML 和 XMLHttpRequest 等

120.解释 XMLHttpRequest 是什么？

XMLHttpRequest，是我们得以实现异步通讯的根本。最早在 IE 5 中以 ActiveX 组件实现；

最近，Mozilla 1.0 和 Safari 1.2 中实现为本地对象。XMLHttpRequest 虽然不是 W3C 标准，但却得到了 FireFox、Safari、Opera、Konqueror、IE 等绝大多数浏览器的支持。

121.谈谈你对 Ajax 的理解。你在项目中如何使用 Ajax？

Ajax（Asynchronous JavaScript + XML），即异步 JavaScript + XML 的缩写，主要用来页面异步刷新，也是构建 RIA 的一种基础技术。

因为它涉及浏览器兼容、跨域等问题，在项目中一般会使用一些基础类库辅助实现，如 jQuery 等。

122.谈谈你对 JSON 的理解。

JSON(JavaScript Object Notation) 是一种轻量级的数据交换格式。 易于人阅读和编写。同时也易于机器解析和生成。 它基于 JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999 的一个子集。 JSON 采用完全独立于语言的文本格式，但是也使用了类似于 C 语言家族的习惯（包括 C, C++, C#, Java, JavaScript, Perl, Python 等）。 这些特性使 JSON 成为理想的数据交换语言。

所以它往往在 AJAX 中替代 XML，交换数据。

123.你的项目中有使用到跨域吗？你在项目中是如何处理 JS 跨域问题的？

有。

一个网站，主要是使用其它网站提供的 javascrip api 如 QQ。使用 script 的 src 可以直接读取跨域资源。

当然跨域还有其它处理方式：如代理服务器、改变 domain、JSONP 等。

124.你在项目中有使用到网页到服务器的即时通信吗？说说你都采用什么手段处理以及你所知道的处理办法？

没有用到，但我知道 html 的 websockets、flash 的 socket、ajax 长轮询等都可以实现。

125.你在 AJAX 中有遇到乱码吗？如果遇到，你是如何解决的？

遇到过。

一般我首先统一页面和服务器编码，对请求和响应的 Content-Type 设置正确编码；对请求参数进行编码处理。

126.你使用过 jQuery 吗？如果有，你为什么要使用 jQuery 呢？

用过。

如果使用原生 javascript 开发的话，会面临很多问题，如浏览器兼容、Ajax 数据解析、Dom、事件注册操作等都非常烦锁，而 jQuery 正好解决了这些问题。

当然 jQuery 还有非常有用的其它特性，如为 dom 对象绑定数据、动画、等。并且 jQuery 还非常容易扩展，在它的基础上开发非常灵活，也有众多的插件可用，如 jQueryUI、easyUI 等。

127.在 jQuery 中如何注册事件？

使用 bind()方法注册事件，但通常我们使用与事件同名的方法注册更方便，如：click()、hover()等。

128.如何获取 Html 内容？如何获取文本内容？如何获取属性值？如何获取 input 值？如何创建新的节点？

可以使用 html()获取 html 内容。

使用 text()获取文本内容。

使用 attr()可以获取属性值，使用 css()可以获取样式属性值。

通过 val()便可以获取 input 的值

把节点元素名加上<、>作为参数调用 jQuery 方法便可创建新节点，如：jQuery (“ ”)。

129.如何向页面插入节点？

调用 append 方法，将新节点作为参数。

130.ajax、get、post、ajaxSetup、getJSON 各有何用途？

ajax：jQuery 对 ajax 执行的核心方法。其它 ajax 方法都是使用该方法实现。

get：专门用于发送 get 请求的便捷方法。

post：专门用于发送 post 请求的便捷方法。

ajaxSetup：设置调用 ajax 方法时的黑认值。

getJSON：专门用于向服务器请求 json 格式数据的便捷方法。

131.如何使用从服务器获取一个复杂数据（对象）？

通常会把这个数据转换为通用的数据交换格式，如 xml 或 json。由于 xml 解析比较麻烦，所以使用 json 比较多。

在 jQuery 中有专门的获取服务器 json 数据的方法，getJSON()，在回调中，jQuery 会自动将 json 转换为 javascript 对象。

132.addClass、css 有何用途？

addClass：为元素设置 class 属性，如果该元素已经存在 class 属性，则在其值后添加空格及新的 class 值。

css：操作元素的 style 属性的方法。

133.如何获取一个元素的实际位置？

使用 position()或 offset()都可以。

134.bind()、unbind()、hover()有何用途？

bind()：注册特定事件

unbind()：删除特定事件

hover()：同时注册鼠标移入、移出事件

135.你知道 jQuery 插件吗？你了解 jQuery 执行原理和插件机制吗？你都用过哪些 jQuery 插件？

知道 jQuery 插件。

其原理是扩展 jQuery 本身及其核心函数的原型实现。可以调用其 extend 实现对它的扩展。

jQuery 插件有很多，常见的有：jQueryUI、jQuery-Cookie、jQuery-Timer 等。

136.Array 的 join、push、splice、slice 各有何用途，splice 与 slice 有何异同？

join：使用指定间隔符连接所有元素为字符串

push：在尾部添加元素并维护 array 实例的 length

splice 与 slice 都是截取一部分元素。不同的在于：slice 返回截取后的新实例，splice 在原 array 实例上操作。

137.什么是 Bom 什么是 Dom?你如何理解 Dom?

Bom 指浏览器对象，Dom 指文档模型中定义的对象。

138.DOM 操作中，如何获取元素的属性值？

参考答案：

对于元素节点，获取其某属性的值有多种方式，如下所示：

1、element.attributes[下标].value

2、element.attributes['属性名'].value

3、element.getAttributeNode('属性名').value

4、element.getAttribute('属性名')

139.简要描述 DOM 操作中查找元素的方式

参考答案：

通过 HTML 中的信息选取元素，比如：

a) getElementById()方法：根据元素的 id 属性值查询单个节点；

b) getElementsByTagName()方法：根据元素标签的名称查询点；

c) getElementsByName()方法：根据元素 name 属性的值查询点。

2、通过 CSS 类选取元素

a) getElementsByClassName('className')方法：根据 class 名称选取元素；

b) querySelector('selector')和 querySelectorAll('selector')方法：根据 CSS 选择器选取元素。

3、通过 document 对象选取，如 document.all、document.body 等；

4、通过节点遍历选取节点，如 parentNode、firstChild 等。

140.如何判断一个变量的值是否为数字？以及有哪些手段判断变量值的数据类型？

全局函数 isNaN 可以判断一个变量的值是否为数字。

可以使用运算符 type、instanceof 判断变量值的数据类型。

141.javascript 的 typeof 返回哪些数据类型

Object number function boolean underfind

142.例举 3 种强制类型转换和 2 种隐式类型转换？

强制（parseInt,parseFloat,number）

隐式（== - ===）

143.split() join() 的区别

前者是切割成数组的形式，后者是将数组转换成字符串

144.数组方法 pop() push() unshift() shift()

Push()尾部添加 pop()尾部删除

Unshift()头部添加 shift()头部删除

145.IE 和 DOM 事件流的区别

1.执行顺序不一样、

2.参数不一样

3.事件加不加 on

4.this 指向问题

146.IE 和标准下有哪些兼容性的写法

```
Var ev = ev || window.event
document.documentElement.clientWidth || document.body.clientWidth
Var target = ev.srcElement||ev.target
```

147.ajax 请求的时候 get 和 post 方式的区别

一个在 url 后面 一个放在虚拟载体里面

有大小限制

安全问题

应用不同 一个是论坛等只需要请求的，一个是类似修改密码的

148.ajax 请求时，如何解释 json 数据

使用 eval parse 鉴于安全性考虑 使用 parse 更靠谱

149.事件委托是什么

让利用事件冒泡的原理，让自己的所触发的事件，让他的父元素代替执行！

150.如何阻止事件冒泡和默认事件

```
canceBubble   return false
```

151.添加 删除 替换 插入到某个接点的方法

```
obj.appendChidl()
obj.innersetBefore
obj.replaceChild
obj.removeChild
```

152.解释 jsonp 的原理，以及为什么不是真正的 ajax

动态创建 script 标签，回调函数

Ajax 是页面无刷新请求数据操作

153.javascript 的本地对象，内置对象和宿主对象

本地对象为 array obj regexp 等可以 new 实例化

内置对象为 gload Math 等不可以实例化的

宿主为浏览器自带的 document,window 等

154.document load 和 document ready 的区别

Document.onload 是在结构和样式加载完才执行 js

Document.ready 原生种没有这个方法，jquery 中有 $().ready(function)

155.“==”和“===”的不同

前者会自动转换类型

后者不会

156.javascript 的同源策略

一段脚本只能读取来自于同一来源的窗口和文档的属性，这里的同一来源指的是主机名、协议和端口号的组合

157.编写一个数组去重的方法

```
function oSort(arr)

        {

            var result ={};

            var newArr=[];

            for(var i=0;i

            {

                if(!result[arr[i]])

                {

                    newArr.push(arr[i])

                    result[arr[i]]=1

                }
```

```
            }

            return newArr

        }
```
复制代码

158.JavaScript 原型，原型链 ？有什么特点？

原型(prototype)：保存所有子对象共有成员的对象

原型链：由各级对象的__proto__逐级向上引用形成的多级继承关系

特点：1.原型和原型链是 JS 实现继承的一种模型。
2.原型链的形成是真正是靠__proto__ 而非 prototype

159.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）。

160.null，undefined 的区别？

Null: 这是一个对象，但是为空。因为是对象，所以 typeof null 返回 'object' 。

Undefined: undefined 是全局对象（window）的一个特殊属性，其值是未定义的。

161.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, fals
e);
            } 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 e
vent;
        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;
        }
    };
```

162.Node.js 的适用场景？

高并发、聊天、实时消息推送

163.介绍 js 的基本数据类型。

number,string,boolean,object,undefined

164.Javascript 如何实现继承？

通过原型和构造器

165.["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)

166.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) }
```

```
}
```

167.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

168.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();

169.什么是闭包（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
```

170."use strict";是什么意思 ？ 使用它的好处和坏处分别是什么？

除了正常运行模式，ECMAscript 5 添加了第二种运行模式："严格模式"（strict mode）。顾名思义，这种模式使得 Javascript 在更严格的条件下运行。

好处：消除 Javascript 语法的一些不合理、不严谨之处，减少一些怪异行为;

- 消除代码运行的一些不安全之处，保证代码运行的安全；
- 提高编译器效率，增加运行速度；
- 为未来新版本的 Javascript 做好铺垫。

坏处：同样的代码，在"严格模式"中，可能会有不一样的运行结果；一些在"正常模式"下可以运行的语句，在"严格模式"下将不能运行

171.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');
}
```

172.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);
```

173.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

174.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

175.同步和异步的区别?

同步：向服务器端发送请求，到服务器端返回响应，这个过程需要等待

异步：向服务器端发送请求，到服务器端返回响应，这个过程无需等待

176.如何解决跨域问题?

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

177.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
        };
    })();
```

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

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

3. AMD 的 API 默认是一个当多个用，CMD 的 API 严格区分，推崇职责单一。

179.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

180.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

181.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);
```

182.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

183.JQuery 的源码看过吗？能不能简单说一下它的实现原理？

看过，就是对 DOM 进行的封装

184.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)
```

185.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++) {}

186.JavaScript 中的作用域与变量声明提前？

作用域：表示变量或函数起作用的区域，指代了它们在什么样的上下文中执行，亦即上下文执行环境。Javascript 的作用域只有两种：全局作用域和本地作用域，本地作用域是按照函数来区分的。

变量声明提前：变量提到当前作用域的最前，赋值保留在原地

187.如何编写高性能的 Javascript？

1 使用 DocumentFragment 优化多次 append

2 通过模板元素 clone ，替代 createElement

3 使用 firstChild 和 nextSibling 代替 childNodes 遍历 dom 元素

4 使用三目运算符替代条件分支

5 需要不断执行的时候，优先考虑使用 setInterval

…….

188.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）

189.JQuery 一个对象可以同时绑定多个事件，这是如何实现的？

bind() 方法为被选元素添加一个或多个事件处理程序，并规定事件发生时运行的函数。
on() 方法事件处理程序到当前选定的 jQuery 对象中的元素。
delegate() 方法为指定的元素（属于被选元素的子元素）添加一个或多个事件处理程序，并规定当这些事件发生时运行的函数。
live() 方法为被选元素附加一个或多个事件处理程序，并规定当这些事件发生时运行的函数

190.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

191.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。

192.什么是 web worker?为什么我们需要他们?

Web worker 帮助我们用异步执行 javascript 文件;

193.Web worker 线程的限制是什么?

Web worker 线程不能修改 HTML 元素，你可以自由使用 javascript 数据类型。

194.如何在 javascript 中创建一个 worker 线程?

```
Var worker=new Worker(... .js);
```

195.如何终止 web worker?

```
W.terminate();  //terminate(终止);
```

196.为页面动态添加按钮

使用 js 代码为页面动态添加 5 个按钮，每个按钮上的文本“button1”、“button2”...“button5”。单击每个按钮时，分别弹出数字 1、2...5。代码如下：

```
for (var i = 1; i < 6; i++) {
    var input = document.createElement('input');
              (空白处)
    document.body.appendChild(input);
}
```

为空白处填上代码，实现所需功能

参考答案：

代码如下所示：

197.总结可以实现页面跳转和刷新的方法

参考答案：

使用超级链接，代码如：

```
<a href=”url”></a>
```

2、表单提交，代码如：

```
<form action=”url”></form>
```

3、JS 代码，代码如：

```
location.href=”url”;
location.assign(‘url’);
location.replace();
location.reload();
window.open(‘url’);
history.go();
```

198.body 中的 onload()函数和 jQuery 中的$(document).ready()有什么区别

参考答案：

onload（）和 document.ready（）的区别如下：

可以在页面中使用多个 document.ready()，但只能使用一次 onload()；

document.ready()函数在页面 DOM 元素加载完成以后就会被调用，而 onload()函数则要在所有的关联资源（包括图像，音频）加载完毕才会调用。

199.jQuery 中有哪几种类型的选择器

参考答案：

有 3 种类型的选择器，如下：

基本选择器：直接根据 id，css 类名，元素名返回 dom 元素；

层次选择器：也叫做路径选择器，可以根据路径层次来选择相应的 dom 元素；

过滤选择器：在前面的基础上过滤相关条件，得到匹配的 dom 元素。

200.jQuery 的美元符号$有什么作用？

参考答案：

美元符号$是'jQuery'的别名，它是 jQuery 选择器，查看如下代码：

```
$(document).ready(function(){});
```

也可以用 jQuery 来替代

201.jQuery 中的 Delegate（）函数有什么作用？

参考答案：

delegate()会在以下两个情况下使用到：

当需要给父元素其下的子元素添加事件时，代码如下：

```
Html 代码
$("ul").delegate("li", "click", function(){
$(this).hide();
});
```

当元素在当前页面中不可用时，可以使用 delegate()

202.请写出至少 5 种常见的 http 状态码以及代表的意义

参考答案：

5 种常见的 http 状态码以及代表的意义如下：

200（OK）：请求已成功，请求所希望的响应头或数据体将随此响应返回。

303（See Other）：告知客户端使用另一个 URL 来获取资源。

400（Bad Request）：请求格式错误。1)语义有误，当前请求无法被服务器理解。除非进行修改，否则客户端不应该重复提交这个请求；2)

请求参数有误。

404（Not Found）：请求失败，请求所希望得到的资源未被在服务器上发现。

500（Internal Server Error）：服务器遇到了一个未曾预料的状况，导致了它无法完成对请求的处理。

203.AJAX 应用和传统 Web 应用有什么不同

参考答案：

在传统的 Javascript 编程中，如果想得到服务器端数据库或文件上的信息，或者发送客户端信息到服务器，需要建立一个 HTML form 然后 GET 或者 POST 数据到服务器端。用户需要点击“Submit”按钮来发送或者接受数据信息，然后等待服务器响应请求，页面重新加载。因为服务器每次都会返回一个新的页面，所以传统的 web 应用有可能很慢而且用户交互较差。

使用 AJAX 技术，就可以使 Javascript 通过 XMLHttpRequest 对象直接与服务器进行交互。通过 HTTP Request，一个 web 页面可以发送一个请求到 web 服务器并且接受 web 服务器返回的信息(不用重新加载页面)，展示给用户的还是同一个页面，但是只实现页面的局部刷新，从而提高体验度。

204.AJAX 都有哪些优点和缺点

参考答案：

优点：

页面局部刷新，提高用户体验度；

使用异步方式与服务器通信，具有更加迅速的响应能力；

减轻服务器负担；

基于标准化的并被广泛支持的技术，不需要下载插件或者小程序。

缺点：

不支持浏览器 back 按钮；

安全问题；

对搜索引擎的支持比较弱。

其它问题

一、 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。

二、 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 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

三、 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）

}

四、 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

五、 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

六、 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、

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？

页面重构怎么操作？

列举 IE 与其他浏览器不一样的特性？

99%的网站都需要被重构是那本书上写的？

什么叫优雅降级和渐进增强？

WEB 应用从服务器主动推送 Data 到客户端有那些方式？

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。

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。

如何设计突发大规模并发架构？

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
```

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？

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？

面试题：你还有什么问题要问吗?

回答提示：

企业的这个问题看上去可有可无，其实很关键，企业不喜欢说"没问题"的人，因为其很注重员工的个性和创新能力。企业不喜欢求职者问个人福利之类的问题，如果有人这样问：贵公司对新入公司的员工有没有什么培训项目，我可以参加吗?或者说贵公司的晋升机制是什么样的?企业将很欢迎，因为体现出你对学习的热情和对公司的忠以及你的上进心。

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

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：

- 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。