

2016 年 web 前端 5 大类型面试题

汇总

<一>类型 1

1、常用那几种浏览器测试？有哪些内核 (Layout Engine)?

(Q1) 浏览器：IE，Chrome，FireFox，Safari，Opera。

(Q2) 内核：Trident，Gecko，Presto，Webkit。

2、说下行内元素和块级元素的区别？行内块元素的兼容性使用？（IE8 以下）

(Q1) 行内元素：会在水平方向排列，不能包含块级元素，设置 width 无效，height 无效 (可以设置 line-height)，margin 上下无效，padding 上下无效。块级元素：各占据一行，垂直方向排列。从新行开始结束接着一个断行。

(Q2) 兼容性：display:inline-block;*display:inline;*zoom:1;

3、清除浮动有哪些方式？比较好的方式是哪一种？

(Q1)

(1) 父级 div 定义 height。

(2) 结尾处加空 div 标签 clear:both。

(3) 父级 div 定义伪类 :after 和 zoom。

(4) 父级 div 定义 overflow:hidden。

(5) 父级 div 定义 overflow:auto。

(6) 父级 div 也浮动，需要定义宽度。

(7) 父级 div 定义 display:table。

(8) 结尾处加 br 标签 clear:both。

(Q2) 比较好的是第 3 种方式，好多网站都这么用。

4、box-sizing 常用的属性有哪些？分别有什么作用？

(Q1)box-sizing: content-box|border-box|inherit;

(Q2)content-box: 宽度和高度分别应用到元素的内容框。在宽度和高度之外绘制元素的内边距和边框 (元素默认效果)。

border-box: 元素指定的任何内边距和边框都将在已设定的宽度和高度内进行绘制。通过从已设定的宽度和高度分别减去边框和内边距才能得到内容的宽度和高度。

5、Doctype 作用？标准模式与兼容模式各有什么区别？

(Q1) <!DOCTYPE> 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呈现。

(Q2) 标准模式的排版和 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。

6、HTML5 为什么只需要写 `<!DOCTYPE HTML>` ？

HTML5 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）。

而 HTML4.01 基于 SGML，所以需要对 DTD 进行引用，才能告知浏览器文档所使用的文档类型。

7、 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。

8、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 引擎则：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。

9、html5 有哪些新特性？如何处理 HTML5 新标签的浏览器兼容问题？如何区分 HTML 和 HTML5 ？

(Q1)

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。

(1)绘画 canvas;

(2)用于媒介回放的 video 和 audio 元素；

(3)本地离线存储 localStorage 长期存储数据，浏览器关闭后数据不丢失；

(4)sessionStorage 的数据在浏览器关闭后自动删除；

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

(6)表单控件，calendar、date、time、email、url、search;

(7)新的技术 webworker, websocket, Geolocation;

(Q2)

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。当然也可以直接使用成熟的框架、比如 html5shim，

```
<!--[if lt IE 9]>
```

<![endif]>

10、简述一下你对 HTML 语义化的理解？

用正确的标签做正确的事情。

html 语义化让页面的内容结构化，结构更清晰，便于对浏览器、搜索引擎解析；即使
在没有样式 CSS 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
爬虫也依赖于 HTML 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。

JavaScript

1、介绍 js 的基本数据类型

Undefined 、 Null 、 Boolean 、 Number 、 String

2、js 有哪些内置对象？

数据封装类对象： Object 、 Array 、 Boolean 、 Number 和 String

其他对象： Function 、 Arguments 、 Math 、 Date 、 RegExp 、 Error

3、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 。

4、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 + ')')。

5、DOM 怎样添加、移除、移动、复制、创建和查找节点

// 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

// 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

// 查找

getElementsByName() //通过名称

getElementsByTagName() //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

6、`null` 和 `undefined` 的区别？

`null` 是一个表示 "无" 的对象，转为数值时为 `0`；`undefined` 是一个表示 "无" 的原始值，转为数值时为 `NaN`。

`undefined`：

- (1) 变量被声明了，但没有赋值时，就等于 `undefined`。
- (2) 调用函数时，应该提供的参数没有提供，该参数等于 `undefined`。
- (3) 对象没有赋值的属性，该属性的值为 `undefined`。
- (4) 函数没有返回值时，默认返回 `undefined`。

`null`：

- (1) 作为函数的参数，表示该函数的参数不是对象。
- (2) 作为对象原型链的终点。

7、`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`。

8、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。

格式：采用键值对，例如：`{'age':'12', 'name':'back'}`

9、`call()` 和 `apply()` 的区别和作用？

`apply()` 函数有两个参数：第一个参数是上下文，第二个参数是参数组成的数组。如果上下文是 `null`，则使用全局对象代替。

如：`function.apply(this,[1,2,3]);`

`call()` 的第一个参数是上下文，后续是实例传入的参数序列。

如：`function.call(this,1,2,3);`

10、如何获取 UA？

`function whatBrowser()`

`{ document.Browser.Name.value=navigator.app`

`Name;`

`document.Browser.Version.value=navigator.appVersion;`

`document.Browser.Code.value=navigator.appCodeName;`

```
document.Browser.Agent.value=navigator.userAgent;
```

```
}
```

其他

1、HTTP 状态码知道哪些？

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）。

2、你有哪些性能优化的方法？

（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。

3、 什么叫优雅降级和渐进增强？

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

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

4、哪些常见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）。

5、线程与进程的区别

一个程序至少有一个进程，一个进程至少有一个线程。

线程的划分尺度小于进程，使得多线程程序的并发性高。

另外，进程在执行过程中拥有独立的内存单元，而多个线程共享内存，从而极大地提高了程序的运行效率。

线程在执行过程中与进程还是有区别的。每个独立的线程有一个程序运行的入口、顺序执行序列和程序的出口。但是线程不能够独立执行，必须依存在应用程序中，由应用程序提供多个线程执行控制。

从逻辑角度来看，多线程的意义在于一个应用程序中，有多个执行部分可以同时执行。但操作系统并没有将多个线程看做多个独立的应用，来实现进程的调度和管理以及资源分配。这就是进程和线程的重要区别。

<二>类型 2

作者：青青

链接：<https://zhuanlan.zhihu.com/p/23842369>

来源：知乎 著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

1、http1.1 比 1.0 有哪些改进

(1)、支持持久链接

Connection:keep-alive;

(2)、支持虚拟主机

Host:http://tmooc.cn

一台服务器中可以同时存在多个网站，公用 IP 地址，但域名不同

(3)、支持代理链接

Proxy:xxx; 客户端可以通过中间代理服务器访问

目标服务器 2、提高一个网站访问性能

(1)、域名解析 尽可能减少域名解析的次数，减少跨站

外部资源的访问

(2)、创建链接 尽量减少创建链接次数，采用持久链

接避免重复链接

(3)、发送请求

尽力减少请求次数，资源合并，合理使用缓存

(4)、等待响应 提高服

务器端相应速度

(5)、接受响应 缩短相应消

息下载时间

3、谈一下对 Ajax 的理解

异步的 js 和 xml, 通过 Ajax，可以使用 js 来调用 XMLHttpRequest 对象，

直接与服务器进行通信。可以在页面不完全刷新的情况下，局部更新页面数据。

4、原生 Ajax 代码

```
var xhr = new XMLHttpRequest();
```

```
xhr.onreadystatechange =
```

```
function(){ if(xhr.readyState == 4){
```

```
if(xhr.status ==
```

```
200){ succ();
```

```
}else{
```

```
err();
```

```
}
```

```
complete();
```

```
}
```

```
}
```

```
xhr.open();
```

```
// 如果为 POST
```



```
xhr.open('POST','1.php',true);
```

```
// 设置请求主题的消息类型
```

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

```
beforeSend();
```

```
xhr.send();
```

5、跨域请求 浏览器从某个域名下的资源访问另一个域名下的资源，可能是协议，域名，端口号不同

1)、方案 1

给被跨域访问的资源添加相应消息头部，设定允许来至于那个域名下的页面访问当前页面

```
header("Access-Control-Allow-Origin:http://xx.com");
```

2)、方案 2

JSONP

客户端定义操作 JSON 数据的函数，开始请求资源。

被请求资源在设置 header 头部的时候，把 Json 改成 js 格式的类型，

返回客户端的时候，字符串改成 函数名 (json 参数)” 的形式

....

6、 HTML5 新增了那些标签，废除了那些标签

新的表单元素

datalist 、 progress 、 meter 、 output

7、 HTML5 新增了那些标签属性，废除了那些标签属性

新的 input type

(1)、number 显示一个数字输入框

(2)、email 显示一个邮件地址输入框

(3)、url 显示一个 url 地址输入框

(4)、tel 显示一个电话号码输入框

(5)、search 显示搜索输入框

(6)、range 显示一个范围选择滑块

(7)、color 显示一个颜色选择框

(8)、date 显示一个日期选择框

(9)、month 显示一个月份选择器

(10)、week 显示一个星期 /周选择器

表单新属性

(1)、placeholder : 站位消息

(2)、autoplay 自动获得焦点

(3)、multiple: 多选，用于 email 和 file

(4)、autocomplete: 自动完成

(5)、form: 自定当前输入域所属的表单

(6)、required: 必填项

(7)、min : 最小值

(8)、max: 最大值

(9)、min-length / max-length : 最小/最大长度

(10)、pattern : 定义正则表达式

PS: 总结 6、7

(1) 新的 input type —— 10 个

number 、 email 、 url 、 tel 、 search 、 range 、 color 、 date 、 month 、 week

(2) 新的表单元素 ——4 个

datalist 、 progress 、 meter 、 output

(3) 表单元素新的属性 —— 11 个

placeholder 、 autofocus 、 autocomplete 、 multiple 、 form

----- 验证相关

required 、 min 、 max 、 minlength 、 maxlength 、 pattern

8 、 SVG 和 Canvas 的比较

SVG Canvas

绘制的是矢量图 绘制的是位图

每个图形 / 图像都是独立的标签 只有画布是标签 容易为每个图形添加事

件监听 不能为每个图形添加独立的事件监听 可

以无限缩放，适合地图 不能无限放大，适合表现颜色细节，例如游戏

9 、描述一下浏览器的线程模型

Program: 程序，编写好的代码经过处理，可以在计算机上执行，放在文件系统上

Progress: 进程 / 任务，程序从磁盘调入内存，分配必须可执行代码空间、数据空间，随时 准备被 CPU 执行

thread: 线程，进程内部执行代码的基本单位

进程和线程的关系：

进程是操作系统分配内存的基本单位

线程是执行代码的基本单位

线程处于进程内部， 一个进程内部必须至少有一个线程， 也可以有多个线程， 这些线程之间可以没有影响，并发执行（宏观上是同时执行，微观上看是交替执行）

10 、JS 内置对象

String 、 Number 、 Boolean---- 原始类型

Array Math Date RegExp Error Function Object Global(在浏览器中被替换 window) ；

11 、检索关键词的位置

```
while((i=str.indexOf(" 关键词 ",fromi))!=-
```

```
1){ i // 关键词的位置
```

```
};
```

```
while((i=str.lastIndexOf(" 关键词 ",fromi))!=-
```

```
1){ i // 关键词的位置
```

```
}
```

12 、Error 专门封装错误信息的对象

6 种

SyntaxError ReferenceError TypeError RangeError URIError EvalError

13 、try catch 中的 return

如果 finally 中有 return 则使用 finally 中的 return 返回

如果 finally 中没有 return ,try 中的 return ，会先获得要返回的值，但挂起，暂不返回，要等 finally 中的代码执行完，才返回

14 、检索 reg.exec(str);

```
while((arr=reg.exec(str))!=null){
```

```
arr[0] //本次找到的关键词
```

```
arr[i] // 第 i 个分组的子内容
```

```
arr.index //本次找打的关键词位置
```

```
}
```

15 、 Date API

FullYear Month Date Day

Hours Minutes Seconds Milliseconds

Day 没有 set 方法 其他都有 get set 方法

<三>类型 3

链接：<https://zhuanlan.zhihu.com/p/23906368> 来源：知乎

1、什么是盒子模型？

在网页中，一个元素占有空间的大小由几个部分构成，其中包括元素的内容（ **content** ），元素的内边距（ **padding** ），元素的边框（ **border** ），元素的外边距（ **margin** ）四个部分。这四个部分占有的空间中， 有的部分可以显示相应的内容， 而有的部分只用来分隔相邻的区域或区域。 4 个部分一起构成了 **css** 中元素的盒模型。

2、行内元素有哪些？块级元素有哪些？ 空 (void) 元素有那些？

行内元素： **a** 、 **b** 、 **span** 、 **img** 、 **input** 、 **strong** 、 **select** 、 **label** 、 **em** 、 **button** 、 **textarea**

块级元素： **div** 、 **ul** 、 **li** 、 **dl** 、 **dt** 、 **dd** 、 **p** 、 **h1-h6** 、 **blockquote**

空元素：即系没有内容的 **HTML** 元素，例如： **br** 、 **meta** 、 **hr** 、 **link** 、 **input** 、 **img**

3、CSS 实现垂直水平居中 一道经典的问题，实现方法有很

多种，以下是其中一种实现：

HTML 结构：

```
<div class="wrapper">
```

```
<div class="content"></div>
```

```
</div>
```

CSS：

```
.wrapper{position:relative;}
```

```
.content{

background-color:#6699FF;

width:200px;

height:200px;

position: absolute; //父元素需要相对定位

top: 50%;

left: 50%;

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

margin-left: -100px;

}
```

4、简述一下 `src` 与 `href` 的区别

`href` 是指向网络资源所在位置，建立和当前元素（锚点）或当前文档（链接）之间的链接，用于超链接。

`src` 是指向外部资源的位置，指向的内容将会嵌入到文档中当前标签所在位置；在请求资源时会将其指向的资源下载并应用到文档内，例如 `js` 脚本，`img` 图片和 `frame` 等元素。当浏览器解析到该元素时，会暂停其他资源的下载和处理，直到将该资源加载、编译、执行完毕，图片和框架等元素也如此，类似于将所指向资源嵌入当前标签内。这也是为什么将 `js` 脚本放在底部而不是头部。

5、什么是 CSS Hack?

一般来说是针对不同的浏览器写不同的 CSS, 就是 CSS Hack 。

IE 浏览器 Hack 一般又分为三种，条件 Hack、属性级 Hack、选择符 Hack（详细参考 CSS 文档：css 文档）。例如：

// 1、条件 Hack

```
<!--[if IE]>
```

```
<style>
```

```
.test{color:red;}
```

```
</style>
```

```
<![endif]-->
```

// 2、属性 Hack

```
.test{ color:#
```

```
0909;
```

```
*color:#f00;
```

```
_color:#ff0;
```

```
}
```

// 3、选择符 Hack

```
* html .test{color:#090;}
```

```
* + html .test{color:#ff0;}
```

6、简述同步和异步的区别 同步是阻塞

模式，异步是非阻塞模式。

同步就是指一个进程在执行某个请求的时候，若该请求需要一段时间才能返回信息，那么这个进程将会一直等待下去，直到收到返回信息才继续执行下去；

异步是指进程不需要一直等下去，而是继续执行下面的操作，不管其他进程的状态。当有消息返回时系统会通知进程进行处理，这样可以提高执行的效率。

7、px 和 em 的区别

px 和 em 都是长度单位，区别是，px 的值是固定的，指定是多少就是多少，计算比较容易。

em 得值不是固定的，并且 em 会继承父级元素的字体大小。

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

8、什么叫优雅降级和渐进增强？ 渐进增

强 **progressive enhancement** :

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

优雅降级 **graceful degradation** : 一开始就构建完整的功

能，然后再针对低版本浏览器进行兼容。 区别：

a. 优雅降级是从复杂的现状开始，并试图减少用户体验的供给

b. 渐进增强则是从一个非常基础的，能够起作用的版本开始，并不断扩充，以适应未来环境的需要

c. 降级（功能衰减）意味着往回看；而渐进增强则意味着朝前看，同时保证其根基处于安全地带

9、浏览器的内核分别是什么 ？

IE: trident 内核

Firefox : gecko 内核

Safari : webkit 内核

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

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

JavaScript 部分

1、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()` //添加

`removeChild()` // 移 除

`replaceChild()` // 替 换

`insertBefore()` // 插 入

3) 查找

`getElementsByTagName()` //通过标签名称

`getElementsByName()` //通过元素的 `Name` 属性的值

`getElementById()` //通过元素 `Id`，唯一性

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

```
/**
```

```
* 对象克隆
```

```
* 支持基本数据类型及对象
```

```
* 递归方法
```

```
*/
```

```
function clone(obj)
```

```
    { var o;
```

```
    switch (typeof obj)
```

```
    { case "undefined":
```

```
        break;
```

```
        case "string":
```

```
o = obj + "";
```

```
break;
```

```
case "number":
```

```
o = obj - 0;
```

```
break;
```

```
case "boolean":
```

```
o = obj;
```

```
break;
```

```
case "object": // object 分为两种情况 对象（ Object ）或数组（ Array ）
```

```
if (obj === null)
```

```
{ o = null;
```

```
} else {
```

```
if (Object.prototype.toString.call(obj).slice(8, -1) === "Array")
```

```
{ o = [];
```

```
for (var i = 0; i < obj.length; i++)
```

```
{ o.push(clone(obj[i]));
```

```
}
```

```
} else
```

```
{ o = {};
```

```
for (var k in obj)
```

```
{ o[k] =
```

```
clone(obj[k]);
```

```
}
```

```
}  
  
}  
  
break;  
  
default:  
  
o = obj;  
  
break;  
  
}  
  
return o;  
  
}
```

3、如何消除一个数组里面重复的元素？

// 方法一：

```
var arr1 =[1,2,2,2,3,3,3,4,5,6],  
  
arr2 = [];  
  
for(var i = 0,len = arr1.length; i < len;  
  
i++){ if(arr2.indexOf(arr1[i]) <  
  
0){ arr2.push(arr1[i]);  
  
}  
  
}  
  
document.write(arr2); // 1,2,3,4,5,6
```

4、想实现一个对页面某个节点的拖曳？如何做？

使用原生 JS

5、在 Javascript 中什么是伪数组？如何将伪数组转化为标准数组？

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

```
function log(){

var args = Array.prototype.slice.call(arguments);

//为了使用 unshift 数组方法，将 argument 转化为真正的数组

args.unshift('(app)');

console.log.apply(console, args);

};
```

6、Javascript 中 `callee` 和 `caller` 的作用？

`caller` 是返回一个对函数的引用，该函数调用了当前函数；

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

7、请描述一下 `cookies`，`sessionStorage` 和 `localStorage` 的区别

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

`web storage` 和 `cookie` 的区别

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

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

8、手写数组快速排序 关于快排算法的详细说明，可以参考阮

一峰老师的文章快速排序

快速排序 ”的思想很简单，整个排序过程只需要三步：

（1）在数据集之中，选择一个元素作为 “基准”（ pivot ）。

（2）所有小于 “基准”的元素，都移到 “基准”的左边；所有大于 “基准”的元素，都移到 “基准” 的右边。

（3）对 “基准”左边和右边的两个子集，不断重复第一步和第二步，直到所有子集只剩下一个元素为止。

9、统计字符串 ” aaaabbbccccdddfgh 中字母个数或统计最多字母数。

```
var str = "aaaabbbccccdddfgh";
```

```
var obj = {};
```

```
for(var
```

```
i=0;istr.length;i++){ var v =
```

```
str.charAt(i);
```

```
if(obj[v] & obj[v].value ==
```

```
v){ obj[v].count = ++
```

```
obj[v].count;
```

```
}else{ obj
```

```
[v] = {};
```

```
obj[v].count =
```

```
1; obj[v].value
```

```
=
```

```
v;
```

```
}
```

```

}

for(key in obj){

document.write(obj[key].value    +'='+obj[key].count+' '); // a=4 b=3 c=4 d=2 f=1 g=1 h=1

}

```

10、写一个 function ，清除字符串前后的空格。（兼容所有浏览器）

```

function trim(str) {

if (str & typeof str === "string") {

return str.replace(/(^s*)(s*)$/g,"");    //去除前后空白符

}

}

```

其他部分

1、一次完整的 HTTP 事务是怎样的一个过程？

基本流程：

- a. 域名解析
- b. 发起 TCP 的 3 次握手
- c. 建立 TCP 连接后发起 http 请求
- d. 服务器端响应 http 请求，浏览器得到 html 代码 e.

浏览器解析 html 代码，并请求 html 代码中的资源

f. 浏览器对页面进行渲染呈现给用户 2、对前端工程师这

个职位你是怎么样理解的？

- a. 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
- b. 参与项目，快速高质量完成实现效果图，精确到 1px ；
- c. 与团队成员， UI 设计，产品经理的沟通；

d. 做好的页面结构，页面重构和用户体验；

e. 处理 **hack**，兼容、写出优美的代码格式；

f. 针对服务器的优化、拥抱最新前端技术。

<四>类型 4

作者：mdcc

链接：<https://zhuanlan.zhihu.com/p/23516211>

来源：知乎 著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

HTML、CSS 部分

要点：对 Web 标准的理解、浏览器差异、CSS 基本功：布局、盒子模型、选择器优先级及使用、HTML5、CSS3、移动端开发技术等

1.Doctype 作用？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？

(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。

2.行内元素有哪些？块级元素有哪些？空(void) 元素有那些？

(1) CSS 规范规定，每个元素都有 display 属性，确定该元素的类型，每个元素都有默认的 display 值，比如 div 默认 display 属性值为 “block”，成为 “块级”元素；span 默认 display 属性值为 “inline”，是 “行内”元素。

(2) 行内元素有：a b span img input select strong (强调的语气) 块级元素有：div ul ol li dl dt dd h1 h2 h3 h4 p

(3) 知名的空元素：
 <hr> <input> <link> <meta>
鲜为人知的是：<area> <base> <col> <command> <embed> <keygen> <param>
<source> <track> <wbr>

3.CSS 的盒子模型？

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

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

4.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。

5.CSS 选择符有哪些？哪些属性可以继承？优先级算法如何计算？
些？

CSS3 新增伪类有那

* 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： font-size font-family color, UL LI DL DD DT;

* 不可继承： border padding margin width height ;

* 优先级就近原则，样式定义最近者为准；

* 载入样式以最后载入的定位为准；

优先级为：

!important > id > class > tag

important 比 内联优先级高

CSS3 新增伪类举例：

p:first-of-type 选择属于其父元素的首个 <p> 元素的每个 <p>元素。

p:last-of-type 选择属于其父元素的最后 <p> 元素的每个 <p> 元素。

p:only-of-type 选择属于其父元素唯一的 <p> 元素的每个 <p> 元素。

p:only-child 选择属于其父元素的唯一子元素的每个 <p> 元素。 p:nth-

child(2) 选择属于其父元素的第二个子元素的每个 <p> 元素。

:enabled 、 :disabled 控制表单控件的禁用状态。

:checked ，单选框或复选框被选中。

6. 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%; }
```

7. 浏览器的内核分别是什么 ? 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 ?

* IE 浏览器的内核 Trident 、 Mozilla 的 Gecko 、 google 的 WebKit 、 Opera 内核 Presto ；

* png24 为的图片在 ie6 浏览器上出现背景，解决方案是做成 PNG8.

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

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

浮动 ie 产生的双倍距离 `#box{ float:left; width:10px; margin:0 0 0 100px;}`

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

渐进识别的方式，从总体中逐渐排除局部。

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

```
css
.bb{
background-color:#f1ee18;
.background-color:#00deff\9;
+background-color:#a200ff;
_background-color:#1e0bd1;

}
```

* IE 下,可以使用获取常规属性的方法来获取自定义属性 ,
也可以使用 `getAttribute()` 获取自定义属性 ;

Firefox 下,只能使用 `getAttribute()` 获取自定义属性 .
解决方法 :统一通过 `getAttribute()` 获取自定义属性 .

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

* (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 {}`

8.html5\CSS3 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,地理定位等功能的增加。

* 绘画 canvas 元素

用于媒介回放的 video 和 audio 元素

本地离线存储 localStorage 长期存储数据,浏览器关闭后数据不丢失;
sessionStorage 的数据在浏览器关闭后自动删除 语义化更好的内容元素,

比如 article 、 footer 、 header 、 nav 、 section

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

CSS3 实现圆角,阴影,对文字加特效,增加了更多的 CSS 选择器 多背景 rgba

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 ；

* 是 IE8/IE7/IE6 支持通过 document.createElement 方法产生的标签，
可以利用这一特性让这些浏览器支持 HTML5 新标签，

浏览器支持新标签后，还需要添加标签默认的风格：

* 当然最好的方式是直接使用成熟的框架、使用最多的是 html5shim 框架
<!--[if lt IE 9]>

<![endif]-->

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

* 首先划分成头部、 body 、脚部； 。 。 。 。 。

*
实现效果图是最基本的工作，精确到 2px ；

与设计师，产品经理的沟通和项目的参与 做好

的页面结构，页面重构和用户体验

处理 hack ，兼容、写出优美的代码格式

针对服务器的优化、拥抱 HTML5 。

10. 常使用的库有哪些？常用的前端开发工具？开发过什么应用或组件？

* 使用率较高的框架有 jQuery 、 YUI 、 Prototype 、 Dojo 、 Ext.js 、 Mootools 等。尤其是 jQuery ，
超过 91% 。

轻量级框架有 Modernizr 、 underscore.js 、 backbone.js 、 Raphael.js 等。
(理解这些框架的功能、性能、设计原理)

* Sublime Text 、 Eclipse 、 Notepad 、 Firebug 、 HttpWatch 、 Yslow 。

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

JavaScript 原型，原型链 ？ 有什么特点？

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

* 原型链是由一些用来继承和共享属性的对象组成的（有限的）对象链。

* JavaScript 的数据对象有那些属性值？

`writable` ：这个属性的值是否可以改。

`configurable` ：这个属性的配置是否可以删除，修改。

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

`value` ：属性值。

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

function clone(proto)

{ function

Dummy() {}

Dummy.prototype = proto;

Dummy.prototype.constructor = Dummy;

return new Dummy(); //等价于 Object.create(Person);

}

function object(old)

{ function

F() {};

F.prototype = old;

return new F();

}

var newObj = object(oldObject);

12. 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。向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。

13. 页面重构怎么操作？

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

14. 语义化的理解？

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。

15. **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。

16. 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; }
```

17.(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>
语义 HTML 具有以下特性：
```

文字包裹在元素中，用以反映内容。例如：

段落包含在 <p> 元素中。 顺序表包含在 元素中。

从其他来源引用的大型文字块包含在 <blockquote> 元素中。

HTML 元素不能用作语义用途以外的其他目的。例如：

<h1> 包含标题，但并非用于放大文本。

<blockquote> 包含大段引述，但并非用于文本缩进。

空白段落元素 (<p></p>) 并非用于跳行。 文本并不直接包含任何样式信息。例如：

不使用 或 <center> 等格式标记。

类或 ID 中不引用颜色或位置。

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

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

20. 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。）

21.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。

22.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：

```
div{  
}
```

```
.class1{  
}
```

```
#id1{  
}
```

```
#id1 div{  
}
```

```
.class1 div{  
}
```

```
.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

23.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；

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）。

23. 写一个通用的事件侦听器函数

// event(事件)工具集，来源：<https://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(
nt('on' + type, function()
{ handler.call(eleme

nt);
});
} 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.detachEvent)
{ element.detach
Event('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.preventDefa
ult();

```



```

} 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;
};

```

24.99% 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）

25. 什么叫优雅降级和渐进增强？

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

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

26.Node.js 的适用场景

高并发、聊天、实时消息推送

27.WEB 应用从服务器主动推送 Data 到客户端有那些方式？

html5 websocket

WebSocket 通过 Flash

XHR 长时间连接

XHR Multipart Streaming

不可见的 `iframe`

`<script>` 标签的长时间连接（可跨域）

JavaScript 部分

要点：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、异步装载回调、模板引擎、`Nodejs` 等。

js 的几种数据类型：`number,string,boolean,object,undefined`

js 的常见内置对象类：`Date,Array,Math` 、`Number` 、`Boolean` 、`String` 、`Array` 、`RegExp` 、`Function...`

通常可以做一些小练习来判断 TA 的水平，js 虽然很灵活，但是具体的代码和实现方式能体现一个人的全局观，随着代码规模的增长，复杂度增加，如何合理划分模块实现功能和接口的能力比较重要。（下面例题）

```
["1", "2", "3"].map(parseInt)
```

```
[typeof null, null instanceof Object][[3,2,1].reduce(Math.pow), [].reduce(Math.pow)] ]var val = 'smtg';
```

```
console.log('Value is ' + (val === 'smtg') ? 'Something' : 'Nothing');
```

1.创建一个对象

```
function Person(name, age)
{ this.name = name;
  this.age = age;
  this.sing = function() { alert(this.name) }
}
```

2. 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。

但是总有一个原则，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

3. 事件、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();`

4.什么是闭包（`closure`），为什么要用？

待完善

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
```

5. 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');  
}
```

6. `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);
```

7. JSON 的了解

JSON(JavaScript Object Notation) 是一种轻量级的数据交换格式。它是基于 JavaScript 的一个子集。数据格式简单，易于读写，占用带宽小。

```
{'age':'12', 'name':'back'}
```

8. 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

9. ajax 是什么？ajax 的交互模型？同步和异步的区别？如何解决跨域问题？

待完善

1. 通过异步模式，提升了用户体验

2. 优化了浏览器和服务端之间的传输，减少不必要的数据往返，减少了带宽占用

3. **Ajax** 在客户端运行，承担了一部分本来由服务器承担的工作，减少了大用户量下的服务器负载。

2. **Ajax** 的最大的特点是什么。

Ajax 可以实现动态不刷新（局部刷新）

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

ajax 的缺点

1、**ajax** 不支持浏览器 **back** 按钮。

2、安全问题 **AJAX** 暴露了与服务器交互的细节。

3、对搜索引擎的支持比较弱。

4、破坏了程序的异常机制。

5、不容易调试。

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

10. 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
    };
})();
```

11.对 **Node** 的优点和缺点提出了自己的看法：

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

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

12. 异步加载的方式

(1) **defer**，只支持 **IE**

(2) **async**：

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

document.write 和 **innerHTML** 的区别

document.write 只能重绘整个页面

innerHTML 可以重绘页面的一部分

13. 告诉我答案是多少？

```
(function(x){
  delete x;
  alert(x);
})(1+5);
```

函数参数无法 **delete** 删除，**delete** 只能删除通过 **for in** 访问的属性。
当然，删除失败也不会报错，所以代码运行会弹出 “1”

14. JS 中的 **call()** 和 **apply()** 方法的区别？

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);
```

15. 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

16.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)
```

17.JavaScript 中的作用域与变量声明提升？

其他部分

（HTTP、正则、优化、重构、响应式、移动端、团队协作、SEO、UED、职业生涯）

*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++) {}
```

前端开发的优化问题（看雅虎 14 条性能优化原则）。

（1）减少 http 请求次数：CSS Sprites, JS、CSS 源码压缩、图片大小控制合适；网页 Gzip，CDN 托管，data 缓存，图片服务器。

（2）前端模板 JS+ 数据，减少由于 HTML 标签导致的带宽浪费，前端用变量保存 AJAX 请求结果，每次操作本地变量，不用请求，减少请求次数

（3）用 innerHTML 代替 DOM 操作，减少 DOM 操作次数，优化 javascript 性能。

（4）当需要设置的样式很多时设置 className 而不是直接操作 style。

（5）少用全局变量、缓存 DOM 节点查找的结果。减少 IO 读取操作。

(6) 避免使用 CSS Expression (css 表达式) 又称 Dynamic properties(动态属性)。

(7) 图片预加载，将样式表放在顶部，将脚本放在底部 加上时间戳。

(8) 避免在页面的主体布局中使用 table，table 要等其中的内容完全下载之后才会显示出来，显示比 div+css 布局慢。

http 状态码有那些？分别代表是什么意思？

100-199 用于指定客户端应相应的某些动作。

200-299 用于表示请求成功。

300-399 用于已经移动的文件并且常被包含在定位头信息中指定新的地址信息。

400-499 用于指出客户端的错误。 400 1、语义有误，当前请求无法被服务器理解。 401 当前请求需要用户验证 403 服务器已经理解请求，但是拒绝执行它。

500-599 用于支持服务器错误。 503 - 服务不可用

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

你所知道的页面性能优化方法有那些？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？

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

18. 谈谈你认为怎样做能是项目做的更好？

19. 你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？

20. 加班的看法

加班就像借钱，原则应当是 ----- 救急不救穷

21. 平时如何管理你的项目，如何设计突发大规模并发架构？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
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）

23. 你说你热爱前端，那么应该 WEB 行业的发展很关注吧？ 说说最近最流行的一些东西吧？

Node.js 、Mongodb 、npmM 、MVVM 、MEAN

24. 你有了解我们公司吗？说说你的认识？

因为我想去阿里，所以我针对阿里的说

最羡慕就是在双十一购物节， 350.19 亿元，每分钟支付 79 万笔。海量数据，居然无一漏单、无一故障。太厉害了。

25. 移动端（比如： Android IOS ）怎么做好用户体验 ？

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有：

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。

<五>类型 5

Doctype 作用？标准模式与兼容模式各有什么区别？

(1) `<!DOCTYPE>` 声明位于 HTML 文档中的第一行，处于 `<html>` 标签之前。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呈现。

(2) 标准模式的排版和 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。

HTML5 为什么只需要写 `<!DOCTYPE HTML>`？

HTML5 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）；

而 HTML4.01 基于 SGML，所以需要对 DTD 进行引用，才能告知浏览器文档所使用的文档类型。

行内元素有哪些？块级元素有哪些？空 (void) 元素有那些？

首先：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，如 div 的 display 默认值为 “block” 则为 “块级” 元素；span 默认 display 属性值为 “inline”，是 “行内” 元素。

(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

(3) 常见的空元素：

`
` `<hr>` `` `<input>` `<link>` `<meta>`

鲜为人知的是：

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

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；

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 引擎则：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。

常见的浏览器内核有哪些？

Trident 内核：IE,MaxThon,TT,The World,360, 搜狗浏览器等。 [又称 MSHTML]Gecko 内核：Netscape6 及以上版本，FF,MozillaSuite/SeaMonkey 等

Presto 内核：Opera7 及以上。 [Opera 内核原为：Presto，现为：Blink;]Webkit

内核：Safari,Chrome 等。 [Chrome 的：Blink (WebKit 的分支)]

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, websocket, Geolocation;

移除的元素：

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

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

* 支持 HTML5 新标签：

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

可以利用这一特性让这些浏览器支持 HTML5 新标签，

浏览器支持新标签后，还需要添加标签默认的样式。当

然也可以直接使用成熟的框架、比如 html5shim;

```
<!--[if lt IE 9]>
```

```
<script> src="http://html5shim.googlecode.com/svn/trunk/html5.js"</script>
```

```
<![endif]-->
```

* 如何区分 HTML5：DOCTYPE 声明 新增的结构元素 功能元素

简述一下你对 HTML 语义化的理解？

用正确的标签做正确的事情。

html 语义化让页面的内容结构化，结构更清晰，便于对浏览器、搜索引擎解析；

即使在没有样式 CSS 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

爬虫也依赖于 HTML 标记来确定上下文和各个关键字的权重，利于 SEO; 使阅读

源代码的人对网站更容易将网站分块，便于阅读维护理解。

HTML5 的离线储存怎么使用，工作原理能不能解释一下？

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，更新用户机器上的缓存文件。

原理：HTML5 的离线存储是基于一个新建的 `.appcache` 文件的缓存机制（不是存储技术），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。

如何使用：

- 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:
```

```
resource/logo.png
```

```
FALLBACK:
```

```
//offline.html
```

- 3、在离线状态时，操作 `window.applicationCache` 进行需求实现。
- 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。

请描述一下 `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
iframe 有那些缺点？

***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。

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，这样可以绕开以上两个问题。

Label 的作用是什么？是怎么用的？

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"/>
```

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

HTML5 的 **form** 如何关闭自动完成功能？

给不想要提示的 **form** 或某个 **input** 设置为 **autocomplete=off** 。
如何实现浏览器内多个标签页之间的通信？(阿里)

WebSocket 、**SharedWorker** ；
也可以调用 **localStorage** 、**cookies** 等本地存储方式；

localStorage 另一个浏览上下文里被添加、修改或删除时，它都会触发一个事件，我们通过监听事件，控制它的值来进行页面信息通信；
注意 **quirks**：Safari 在无痕模式下设置 **localStorage** 值时会抛出 **QuotaExceededError** 的异常；
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**
页面可见性（ **Page Visibility API** ） 可以有哪些用途？

通过 **visibilityState** 的值检测页面当前是否可见，以及打开网页的时间等在页面被切换到其他后台进程的时候，自动暂停音乐或视频的播放；

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

实现不使用 border 画出 1px 高的线，在不同浏览器的标准模式与怪异模式下都能保持一致的效果。

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

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。

title 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 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。