

前端面试题整理 by zuoshaohua

Part 1 HTML

1. Doctype作用？严格模式与混杂模式如何区分？它们有何意义？

Doctype的作用

Doctype是Document Type（文档类型）的缩写，`<!DOCTYPE`声明位于文档中的最前面，处于`<html>`标签之前。告诉浏览器以何种模式来渲染文档。`<!DOCTYPE`声明不是HTML标签，`<!DOCTYPE`声明没有结束标签，`<!DOCTYPE`声明对大小写不敏感。

Doctype文档类型声明种类

HTML文档声明的类型主要有3几种：过渡定义类型、严格定义类型、框架定义类型。

HTML 4.01规定了三种文档类型：Strict、Transitional以及Frameset；

XHTML 1.0规定了三种XML文档类型：Strict、Transitional以及Frameset。

严格模式与混杂模式区别

当浏览器厂商开始创建与标准兼容的浏览器时，他们希望确保向后兼容性。为了实现这一点，他们创建了两种呈现模式：标准模式（严格模式）和混杂模式（Quirks模式）。

- （1）在标准模式中，浏览器以其支持的最高标准呈现页面；
- （2）在混杂模式中，页面以一种比较宽松的向后兼容的方式显示。混杂模式通常模拟老式浏览器的行为以防止老站点无法工作。

模式触发

浏览器根据DOCTYPE是否存在以及使用的哪种DTD（Document Type Declaration文档类型声明）来选择要使用的呈现方法。

如果XHTML 1.0、HTML 4.01文档包含形式完整的DOCTYPE，那么它一般以标准模式呈现。

包含过渡DTD（文档类型声明）和URI的DOCTYPE也导致页面以标准模式呈现，但是有过渡DTD而没有URI会导致页面以混杂模式呈现。

DOCTYPE不存在或形式不正确会导致HTML和XHTML文档以混杂模式呈现。

HTML5没有DTD，也就没有严格模式与宽松模式的区别，HTML5有相对宽松的语法，实现时，已经尽可能大的实现了向后兼容。

2. HTML5 为什么只需要写 `<!DOCTYPE HTML>`？

HTML5不引用DTD

HTML 4.01中的doctype需要对DTD进行引用，因为HTML 4.01基于SGML（标准通用标记语言）。而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。早期的HTML版本是基于SGML，因此需要套用SGML的解析规则。DTD的作用在于定义SGML文档的文档类型以便于浏览器解析。

3. 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：span img input select a b 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：br hr img input link meta

鲜为人知的是：area base col command embed keygen param source track wbr

4. 页面导入样式时，使用link和@import有什么区别？

(1)最根本区别：link属于HTML标签，而@import是CSS提供的;link标签除了可以加载CSS外，还可以做很多其它的事情，比如定义RSS，定义rel连接属性等，@import就只能加载CSS了。

(2)加载顺序差别：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)兼容性差别：@import只在IE5以上才能识别，而link是HTML标签，无兼容问题;

(4)使用DOM控制样式时的差别：当使用JavaScript控制DOM去改变样式的时候，只能使用link标签，因为@import不是DOM可以控制的。

(5)link方式的样式的权重高于@import的权重. @import的写法一般有下列几种：@import 'style.css'

//Windows IE4/ NS4, Mac OS X IE5, Macintosh IE4/IE5/NS4不识别

@import "style.css" //Windows IE4/ NS4, Macintosh IE4/NS4不识别

@import url(style.css) //Windows NS4, Macintosh NS4不识别

@import url('style.css') //Windows NS4, Mac OS X IE5, Macintosh IE4/IE5/NS4不识别

@import url("style.css") //Windows NS4, Macintosh NS4不识别

由上分析知道，@import url(style.css)和@import url("style.css")是最优的选择，兼容的浏览器最多。从字节优化的角度来看@import url(style.css)最值得推荐。

5. 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。

6. 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)]

7. 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？

新特性

HTML5现在已经不是SGML的子集，拥有新的语义、图形、多媒体、定位、存储等新的元素。

语义化更好的内容标签（比如 article、footer、header、nav、section）；

用于绘画的 canvas 元素;

音频、视频API(audio, video)；

地理定位(Geolocation) API；

本地离线存储API：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；

新的表单控件：calendar、date、time、email、url、search；

其他新的技术：拖放，Application Cache, webworker, websocket, SSE；

移除的元素

纯表现的元素：acronym, applet, basefont, big, dir, center, font, strike, tt;

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

如何处理兼容性

在旧浏览器上兼容性很差，如IE8-，通过document.createElement方法产生的标签，可以利用这一特性让这些浏览器支持HTML5新标签，浏览器支持新标签后，还需要添加标签默认的风格。

当然最好的方式是直接使用成熟的框架、使用最多的是html5shim框架。

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

```
<script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
```

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

如何区分HTML和HTML5

DOCTYPE声明的方式是区分HTML和HTML5标志的一个重要因素，此外，还可以根据新增的结构、功能元素来加以区分。

8. 简述一下你对HTML语义化的理解？

HTML标签可分为有语义的标签和无语义的标签。比如有语义的标签中table表示表格，form表示表单，a标签表示超链接，strong标签表强调。无语义标签典型的有div和span。

HTML语义化就是根据页面内容的结构（结构语义化），选择合适的语义标签（标签语义化）来书写html页面。

语义化的好处：

- （1）为了在没有CSS的情况下，页面也能呈现出很好的内容结构、代码结构；
- （2）有利于SEO：爬虫依赖于标签来确定上下文和各个关键字的权重，有助于爬虫抓取更多的有效信息；
- （3）方便其他设备解析：使屏幕阅读器、盲人阅读器、移动设备等以一种有意义的方式来渲染网页；
- （4）便于团队开发和维护：语义化代码更具可读性，遵循这个标准可以减少差异化。

写HTML代码时应注意什么？

- （1）尽量使用有语义的标签，如标题（h1~h6）、列表（ul、ol、li、dd、dl、dt）、强调（strong、em）、段落（p）等；
- （2）尽可能少使用无语义的标签div和span；
- （3）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；
- （4）不要使用纯样式标签，如：b、font、u等，改用css设置；
- （5）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）；
- （6）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；
- （7）表单域要用fieldset标签包起来，并用legend标签说明表单的用途；

(8) 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。

9. 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；

```
<html manifest="demo.appcache">
```

(2) 在demo.appcache文件中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。

10. 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。

11. cookies

中文叫做小型文本文件，指某些网站为了辨别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是一小段文本信息，伴随着用户请求和页面在Web服务器和浏览器之间传递。Cookie包含每次用户访问站点时Web应用程序都可以读取的信息。

Cookie原意是小甜饼，是服务器保存在浏览器的一小段文本信息，属于其中一种互联网存储机制。

很干练的对cookie，web storage的介绍

特点：

每个Cookie大小一般不超过4KB；

Cookie保存的信息包括Cookie名、Cookie值、到期时间、所属域名和生效路径；

浏览器每次向服务器发出请求，会在HttpRequest头上带上Cookie信息。

优点：极高的扩展性和可用性

- 1) 数据持久性。
- 2) 不需要任何服务器资源。Cookie存储在客户端并在发送后由服务器读取。
- 3) 可配置到期规则。控制cookie的生命期，使之不会永远有效。偷盗者很可能拿到一个过期的cookie。
- 4) 简单性。基于文本的轻量结构。
- 5) 通过良好的编程，控制保存在cookie中的session对象的大小。
- 6) 通过加密和安全传输技术（SSL），减少cookie被破解的可能性。
- 7) 只在cookie中存放不敏感数据，即使被盗也不会有重大损失。

缺点：

- 1) Cookie数量和长度的限制。

数量：每个域的cookie总数有限。

- a) IE6或更低版本最多20个cookie
- b) IE7和之后的版本最后可以有50个cookie
- c) Firefox 最多 50个cookie
- d) chrome和Safari没有做硬性限制

长度：每个cookie长度不超过4KB（4096B），否则会被截掉。

- 2) 潜在的安全风险。Cookie可能被拦截、篡改。如果cookie被拦截，就有可能取得所有的session信息。
- 3) 用户配置为禁用。有些用户禁用了浏览器或客户端设备接受cookie的能力，因此限制了这一功能。
- 4) 有些状态不可能保存在客户端。例如，为了防止重复提交表单，我们需要在服务器端保存一个计数器。如果我们把这个计数器保存在客户端，那么它起不到任何作用。

12.浏览器本地存储（Web Storage）

Web Storage是HTML5提供的一种新的浏览器端数据储存机制，它提供两种数据存储的对象：

localStorage：该对象存储的数据没有时间限制；

sessionStorage：该对象存储的数据仅用于浏览器的一次对话，当对话结束（通常是窗口关闭），数据被清空；

特性：

- 1) localStorage和sessionStorage对象都是构造函数Storage的实例，拥有同样的属性和方法，二者唯一的区别是存储数据的期限不一。
- 2) localStorage和sessionStorage这两个对象相较于cookie的存储机制，能够提供更大的数据存储空间，但是存储空间根据浏览器类型而定，一般Chrome是2.5MB、Firefox和Opera是5MB、IE是10MB；
- 3) 目前现代浏览器均支持web存储，但IE7-是不支持的，依旧采用传统的cookie保持信息。检测浏览器是否支持web存储的方法有：

```
//method 1
window.localStorage && window.localStorage.getItem
//method 2
if (typeof(Storage) !== 'undefined') {
    //support Web Storage
} else {
}
```

- 4) 使用Web Storage的API进行数据存储时，只能存储为字符串数据，因此所以数据在写入时会隐式调用toString方法转换为字符串。

优势与局限性

优势

存储空间大：与cookie的4k存储空间相比，Web Storage虽然不同浏览器的标准可能不一样，主流的一般都在5~10M；

页面性能优化：本地存储的数据不会被发到服务器，与cookie相比，节省带宽，加快响应速度；接口更加易用。

局限性

server端无法获取本地数据，如有需求，可通过post/get方法；

只能同源共享数据；

接口

由于localStorage和sessionStorage对象的属性和方法是基本一致，下面以localStorage为例，讲解一下localStorage对象的接口；

存储数据 setItem

读取数据 getItem

删除数据 removeItem

获取特定索引 key

13. 请描述一下 cookies，sessionStorage 和 localStorage 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。

浏览器的支持除了IE7及以下不支持外，其他标准浏览器都完全支持(ie及FF需在web服务器里运行)，值得一提的是IE总是办好事，例如IE7、IE6中的UserData其实就是javascript本地存储的解决方案。通过简单的代码封装可以统一到所有的浏览器都支持web storage。

localStorage和sessionStorage都具有相同的操作方法，例如setItem、getItem和removeItem等。

特性 cookie Web Storage

数据生命周期：服务器生成的话，为指定失效时间；浏览器端生成的话默认为关闭浏览器localStorage永久有效，除非使用localStorage.clear()清空；sessionStorage为关闭浏览器后；

存储空间：一般为4K 一般5MB~10MB；

与服务器通信：每次携带在HTTP头中 仅在客户端，如需通信，可通过get或post方法；

应用场景：用户登录身份验证（结合HttpOnly相对安全性高）保存用户购物车信息以及HTML5游戏的本地存储；

14. iframe有那些缺点？

iframe用于在网页内显示网页。 `<iframe src="URL"></iframe>`

优点：

- (1) iframe能够原封不动的把嵌入的网页展现出来。
- (2) 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。
- (3) 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。

(4) 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。

(5) 重载页面时不需要重载整个页面, 只需要重载页面中的一个框架页(减少了数据的传输, 增加了网页下载速度)。

(6) 方便制作导航栏。

缺点:

(1) 会产生很多页面, 不容易管理。

(2) 不容易打印。

(3) 浏览器的后退按钮无效。

(4) 代码复杂, 无法被一些搜索引擎索引到, 这一点很关键, 现在的搜索引擎爬虫还不能很好的处理iframe中的内容, 所以使用iframe会不利于搜索引擎优化。

(5) 多数小型的移动设备(PDA 手机) 无法完全显示框架, 设备兼容性差。

(6) 多框架的页面会增加服务器的http请求, 对于大型网站是不可取的。

*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。

15. 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>
```

16. 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。

17. 如何实现浏览器内多个标签页之间的通信? (阿里)

调用localStorage、cookies等本地存储方式

方法一: 使用localStorage

使用localStorage.setItem(key,value);添加内容使用storage事件监听添加、修改、删除的动作:

方法二、使用cookie+setInterval

如何实现浏览器内多个标签页之间的通信

18. websocket如何兼容低浏览器? (阿里)

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

19. 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;

20.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 要求一个点在不在圆上简单算法、获取鼠标坐标等等

21.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>
```

22.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。

23.title与h1的区别、b与strong的区别、i与em的区别？

title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，strong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。

24.HTML与XHTML的区别？

XHTML简介：

XHTML指可扩展超文本标签语言（EXtensible HyperText Markup Language）。XHTML的目标是取代HTML。
XHTML与HTML 4.01几乎是相同的。
XHTML是更严格更纯净的HTML版本。
XHTML是作为一种XML应用被重新定义的HTML。
XHTML是一个W3C标准。

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规范定义：标签名和属性对大小写敏感。）
XHTML文档必须拥有根元素。

其他区别：

属性名称必须小写。
属性值必须加引号。
属性不能简写。
用id属性代替name属性。

XHTML DTD定义了强制使用的HTML元素。（所有XHTML文档必须进行文件类型声明（DOCTYPE declaration）。在XHTML文档中必须存在html、head、body元素，而title元素必须位于在head元素中。）xhtml中的XML namespace属性是强制性的。

25.除了audio和video，HTML5还有哪些媒体标签？

HTML5对于多媒体提供了强有力的支持，除了audio和video标签外，还支持以下标签：

`<embed>`标签定义嵌入的内容，比如插件。

```
<embed type="video/quicktime" src="Fishing.mov">
```

`<source>`对于定义多个数据源很有用。

```
<video width="450" height="340" controls>
```

```
<source src="jamshed.mp4" type="video/mp4">
```

```
<source src="jamshed.ogg" type="video/ogg">
```

```
</video>
```

`<track>`标签为诸如video元素之类的媒介规定外部文本轨道。用于规定字幕文件或其他包含文本的文件，当媒介播放时，这些文件是可见的。

```
<video width="450" height="340" controls>
```

```
<source src="jamshed.mp4" type="video/mp4">
```

```
<source src="jamshed.ogg" type="video/ogg">
```

```
<track kind="subtitles" label="English" src="jamshed_en.vtt" srclang="en" default></track>
```

```
<track kind="subtitles" label="Arabic" src="jamshed_ar.vtt" srclang="ar"></track>
```

```
</video>
```

26.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。

27.你如何对网站的文件和资源进行优化？

期待的解决方案包括：

文件合并

文件最小化/文件压缩

使用 CDN 托管

缓存的使用（多个域名来提供缓存）

其他

雅虎十四条

1. 尽可能的减少 HTTP 的请求数 [content]
2. 使用 CDN (Content Delivery Network) [server]
3. 添加 Expires 头(或者 Cache-control) [server]
4. Gzip 组件 [server]
5. 将 CSS 样式放在页面的上方 [css]
6. 将脚本移动到底部（包括内联的） [javascript]
7. 避免使用 CSS 中的 Expressions [css]

8. 将 JavaScript 和 CSS 独立成外部文件 [javascript] [css]
9. 减少 DNS 查询 [content]
10. 压缩 JavaScript 和 CSS (包括内联的) [javascript] [css]
11. 避免重定向 [server]
12. 移除重复的脚本 [javascript]
13. 配置实体标签 (ETags) [css]
14. 使 AJAX 缓存

28.请说出三种减少页面加载时间的方法。

1.优化图片

2.图像格式的选择 (GIF : 提供的颜色较少, 可用在一些对颜色要求不高的地方)

3.优化CSS (压缩合并css, 如margin-top,margin-left...)

4.网址后加斜杠 (如www.campr.com/目录, 会判断这个“目录是什么文件类型, 或者是目录。)

5.标明高度和宽度 (如果浏览器没有找到这两个参数, 它需要一边下载图片一边计算大小, 如果图片很多, 浏览器需要不断地调整页面。这不但影响速度, 也影响浏览体验。

当浏览器知道了高度和宽度参数后, 即使图片暂时无法显示, 页面上也会腾出图片的空位, 然后继续加载后面的内容。从而加载时间快了, 浏览体验也更好了。)

6.减少http请求 (合并文件, 合并图片)。

29.你都使用哪些工具来测试代码的性能 ?

WebPagetest:WebPagetest 是性能测试的黄金标准, 根据用户浏览器真实的连接速度, 在全球范围内进行网页速度测试, 并提供详细的优化建议;

Yslow:Yslow 基于 Yahoo 的高性能网页教条, 分析网页的性能并给出响应缓慢的原因;

Google PageSpeed:PageSpeed 根据网页最佳实践分析和优化测试的网页;

简书: [前端性能优化和测试工具总结](#)

30.什么是 FOUC (无样式内容闪烁) ? 你如何来避免 FOUC ?

FOUC - Flash Of Unstyled Content 文档样式闪烁

```
<style type="text/css" media="all">@import "../fouc.css";</style>
```

而引用CSS文件的@import就是造成这个问题的罪魁祸首。IE会先加载整个HTML文档的DOM, 然后再去导入外部的CSS文件, 因此, 在页面DOM加载完成到CSS导入完成中间会有一段时间页面上的内容是没有样式的, 这段时间的长短跟网速, 电脑速度都有关系。解决方法简单的出奇, 只要在之间加入一个<link>或者<script>元素就可以了。

31.img的alt与title有何异同 ? strong与em的异同 ?

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

title(tool tip):该属性为设置该属性的元素提供建议性的信息。

strong:粗体强调标签, 强调, 表示内容的重要性

em:斜体强调标签, 更强烈强调, 表示内容的强调点

32.你能描述一下渐进增强和优雅降级之间的不同吗?

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

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

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

33.为什么利用多个域名来存储网站资源会更有效？

CDN缓存更方便

突破浏览器并发限制

节约cookie带宽

节约主域名的连接数，优化页面响应速度

防止不必要的安全问题

34.请谈一下你对网页标准和标准制定机构重要性的理解。

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

35.简述一下src与href的区别。

src用于替换当前元素，href用于在当前文档和引用资源之间确立联系。

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

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

href是Hypertext Reference的缩写，指向网络资源所在位置，建立和当前元素（锚点）或当前文档（链接）之间的链接，如果我们在文档中添加`<link href="common.css" rel="stylesheet"/>` 那么浏览器会识别该文档为css文件，就会并行下载资源并且不会停止对当前文档的处理。这也是为什么建议使用link方式来加载css，而不是使用@import方式。

36.知道什么是微格式吗？谈谈理解。在前端构建中应该考虑微格式吗？

微格式（Microformats）是一种让机器可读的语义化XHTML词汇的集合，是结构化数据的开放标准。是为特殊应用而制定的特殊格式。优点：将智能数据添加到网页上，让网站内容在搜索引擎结果界面可以显示额外的提示。

37.谈谈以前端角度出发做好SEO需要考虑什么？

- 了解搜索引擎如何抓取网页和如何索引网页

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

- Meta标签优化

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

- 如何选取关键词并在网页中放置关键词

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

- 了解主要的搜索引擎

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

- 主要的互联网目录

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

- 按点击付费的搜索引擎

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

- 搜索引擎登录

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

- 链接交换和链接广泛度（Link Popularity）

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

- 合理的标签使用

38. Canvas vs SVG

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