1. **Iframe有哪些缺点**
2. 索引擎的检索程序无法解读这种页面，不利于SEO;
3. iframe会阻塞主页面的Onload事件；（onload 事件在所有 iframe 加载完毕后(包含里面的元素)才会触发。）
4. 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。

（在页面重要元素加载完成后，再通过 JavaScript 动态设置 iframe 的 SRC 可以避免2）3）情况。）

1. **网页验证码是干嘛的，为了解决什么问题**

1）区分用户是计算机还是人的公共全自动程序；

2）防止恶意破解密码，刷票，论坛灌水。

1. **在页面上实现一个圆形点击区域**

1）border-radius；

2）map+area或者svg

3）纯js，通过需要求一个点在不在圆上简单算法、获取鼠标坐标等等

1. **HTML5的form如何关闭自动完成功能？**

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

1. **浏览器是怎么对HTML5的离线储存资源进行管理和加载的**

1）在线的情况下，浏览器发现html头部有manifest属性，请求manifest文件，如果是第一次访问app，浏览器会根据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。

2）离线的情况下，浏览器就直接使用离线存储的资源。

1. **tite与h1的区别、b与strong的区别、i与em的区别？.**
2. 对搜索引擎来说，title是网页权重的最高点，但title不出现在正文中；title与h1是合作关系，可以达到双重优化的功能，一般一篇文章最好只用一个h1；
3. b和strong都是加粗，strong是给搜索引擎看到，i是给用户看到
4. i和em都是斜体，em是给搜索引擎看得，i是给用户看到
5. **常见的浏览器内核**

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）]

1. **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。

1. **jQuery与vue的区别**

jQuery：采用$操作dom，与原生js的区别仅在于更方便快捷；

vue：利用model与view双向绑定，是的dom操作不需要依赖去的都dom元素

1. **多标签页之间通信**
2. localstorage：利用storage事件、event.newValue和event.oldValue属性
3. cookie+setInterval：将要传递的信息存储在cookie中，每隔一定时间读取cookie信息，即可随时获取要传递的信息（document.cookie， getCookie）
4. **伪类与伪元素**

伪类：通过添加一个实际的类来达到；

伪元素：通过添加一个实际的元素达到。

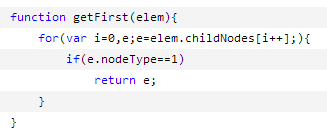
|  |  |
| --- | --- |
| 伪类 | :first-child，:visited，:link，:hover，:active，:focus |
| 伪元素 | :first-letter，:first-line，:before，:after |

Css3规定伪类用:表示，伪元素用::，但由于兼容性，目前大豆还是用一个冒号。

Css3新增伪类：nth-child，only-child等

（15）children与childNodes

childNodes：标准的，返回指定元素的子元素集合，包括HTML节点，所有属性，文本；nodeType==1时才是元素节点，2是属性节点，3是文本节点。



Children：非标准，它返回指定元素的子元素集合。经测试，它只返回HTML节点，甚至不返回文本节点。且在所有浏览器下表现惊人的一致（在ie中包含注释节点）

1. 轮播图

没有过渡动画

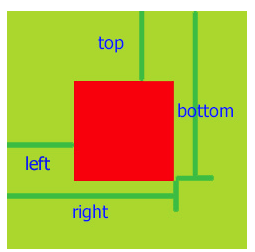
（13）getComputedStyle

获取当前元素所有最终使用的CSS属性值，支持ie9及以上

var style = window.getComputedStyle("元素", "伪类");



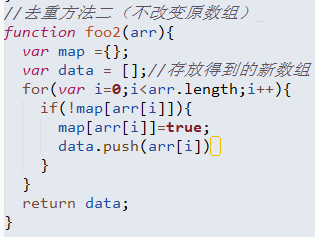
（14）getBoundingClientRect()





获取某个元素相对于视窗的位置集合（top\bottom\left\right），ie9及以上还有width、height

（15）数组去重



（16）跨浏览器事件对象



（17）关于HTML和body

背景色：在HTML和body均设置了背景色时，分别应用各自背景色；只有body设置了背景色时，HTML捕获body的背景色作为其背景色

Margin：HTML与浏览器之间没有margin，body与HTML之间有margin

Height：HTML与body的默认高度均为0，设置body的高度为100%时需要先设置HTML的高度为100%

**（18）创建ajax的过程**

1. 创建XMLHttpRequest对象，
2. 创建一个新的http请求，并指定其方法、url和是否异步发送请求的布尔值；
3. 设置响应http请求的状态改变函数
4. 发送http请求
5. 获取异步请求返回的数据
6. 利用js进行局部刷新

|  |  |
| --- | --- |
| readyState状态 |  |
| 0 | 未初始化：尚未调用open方法 |
| 1 | 启动：已经调用open方法，尚未调用send方法 |
| 2 | 发送：已经调用send方法，尚未接受到响应 |
| 3 | 接受：已经接受到部分响应数据 |
| 4 | 完成：已经接受到全部响应数据，而且已经可以在客户端使用 |

缺点：1）url不变，back键失去作用（如何解决？）；

2）手机等不能很好的支持ajax

（30）isprototype，getPrototypeOf，hasOwnPrototype

|  |  |
| --- | --- |
| isprototype | 确定对象之间是否存在关系，Person.prototype.isPrototypeOf(person1)//true |
| getPrototypeOf | 去的一个对象的原型，Object.getPrototypeOf(person1) == Person.prototype;//true |
| hasOwnPrototype | 检测一个属性是存在与实例中，person1.hasOwnPrototype("name");//若不是来自原型则返回true |

**（29）****attribute和property**

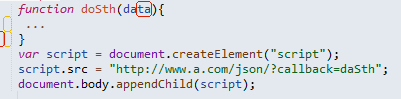
attribute是dom元素在文档中作为html标签拥有的属性；

property就是dom元素在js中作为对象拥有的属性。

所以：对于html的标准属性来说，attribute和property是同步的，是会自动更新的，但是对于自定义的属性来说，他们是不同步的，

【jsonp】（未实践）

**原理**：不使用XHR，采用动态script。在请求地址中指定回调函数



**优点**：简单易用，支持双向通信，可直接访问响应文本

**缺点**：1）jsonp是从其他域中加载代码执行，不能保证其他域是否安全；2）要确定jsonp请求失败不容易（可使用计时器检测指定时间内是否收到了响应）

【CORS】

**原理**：使用自定义的http头部让浏览器和服务器沟通，从而决定请求或响应是否应成功

A页面发送请求时，需要给其附加一个Origin的头部，包含A的地址信息（Origin：http://www.a.com）；B服务器在头部回发相同的地址信息（Access-Control-Allow-Origin:http://www.a.com），如果是公共资源，可以回发\*。若头部缺失或不匹配，浏览器都将驳回请求。

**优点**：由于使用的是XHR，所以错误处理方式比jsonp好；支持get，post等方法