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。

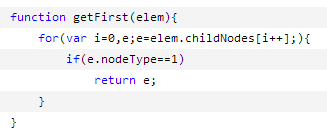
|  |  |
| --- | --- |
| 伪类 | :first-child，:visited，:link，:hover，:active，:focus |
| 伪元素 | :first-letter，:first-line，:before，:after |

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

（12）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）

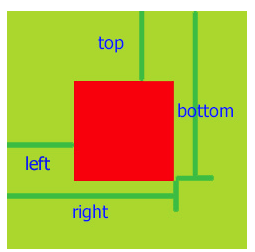
（13）getComputedStyle

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");

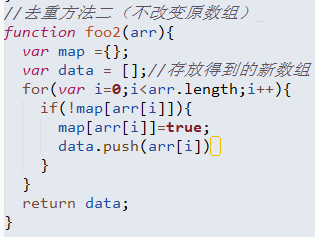


（14）getBoundingClientRect()



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）轮播图

没有过渡动画

（17）跨浏览器事件对象



（18）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%

**（19）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

（20）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 |

**（21）****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

**（22）absolute与float**

float：浮动到包含块的内边界，

absolute：以外边界进行定位（即border界定的位置），不过只设置absolute但不设置left等值时，元素脱离正常流但渲染的位置是它本应该在的位置（即与设置top：0，left：0不一样）

（23）常用的css3特性

1）Box-shadow：

2）text-overflow：（剪切、省略号、用给定字符串代替修剪部分）



3）box-sizing：

4）word-wrap：break-word；允许长单词或者url进行换行

word-break：规定自动换行的处理方式（break-all：允许在单词内换行，keep-all：只能在半角空格或连字符处换行）

white-space：属性设置如何处理元素内的空白（nowrap文本不换行，直到遇到br标签）

Word-spacing，letter-spacing

5）transition: （linear/ease/ease-in/ease-out/ease-in-out）

（在属性里定义原本的属性值大小和transition，在另一处定义运动后要达到的属性值大小，hover或点击等触发另一处时显示过渡效果）

6）border-radius：水平方向/垂直方向（左上，右上，右下，左下）

7）opacity：不透明度，从 0.0 （完全透明）到 1.0（完全不透明）

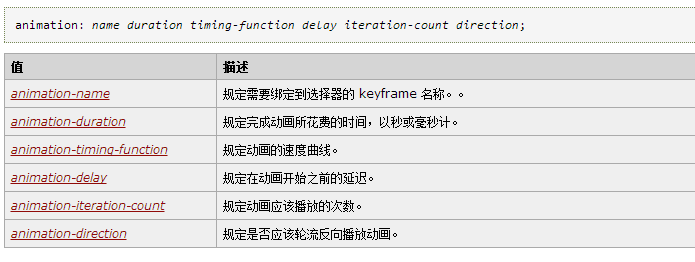
8）媒体查询

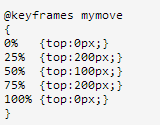
9）更强大的选择器

10）Text-indent：规定文本首行缩进

11）background-size：（cover将图片扩展到足够大，以试完全覆盖区域，contain使高度和宽度完全适应区域）

13）animation和keyframe（页面显示时就自动发生）



（此处的百分比为时长的百分比）

**（24）304缓存的原理**

客户端请求一个页面（A）。 服务器返回页面A，并在给A加上一个ETag。 客户端展现该页面，并将页面连同ETag一起缓存。 客户再次请求页面A，并将上次请求时服务器返回的ETag一起传递给服务器。 服务器检查该ETag，并判断出该页面自上次客户端请求之后还未被修改，直接返回响应304（未修改——Not Modified）和一个空的响应体。

**（25）模块化**

目前，通行的JavaScript模块规范主要有两种：CommonJS**和**AMD**；**

【CommonJS】

特点：1）一个单独的文件就是一个模块，拥有其单独的作用域；2）加载模块使用require方法，只有一个出口，module.exports对象

缺点：require是同步的，会阻塞加载，无法非阻塞的并行加载多个模块（因此不适用于客户端，适用于服务端）

【AMD】：Asynchronous Module Definition异步模块定义

对应的库函数：RequireJS

RequireJS使用define来定义模块，使用require函数加载模块；require()函数在加载依赖的函数的时候是异步加载的，这样浏览器不会失去响应，它指定的回调函数，只有前面的模块都加载成功后，才会运行，解决了依赖性的问题。

【CMD】Common Module Definition通用模块定义

对应的库函数：SeaJS

**AMD和CMD都是异步加载，最大的区别是对依赖模块的执行时机处理不同，注意不是加载的时机或者方式不同**

**（26）BFC**

【触发BFC的情况】

1. 根元素
2. float属性不为none
3. position为absolute或fixed
4. display为inline-block, table-cell, table-caption, flex, inline-flex
5. overflow不为visible

【BFC布局规则】

1. 内部的Box会在垂直方向，一个接一个地放置。
2. Box垂直方向的距离由margin决定。属于同一个BFC的两个相邻Box的margin会发生重叠
3. 每个元素的margin box的左边， 与包含块border box的左边相接触(对于从左往右的格式化，否则相反)。即使存在浮动也是如此。
4. BFC的区域不会与float box重叠。
5. BFC就是页面上的一个隔离的独立容器，容器里面的子元素不会影响到外面的元素。反之也如此。
6. 计算BFC的高度时，浮动元素也参与计算

**（27）浮动和绝对定位不与任何元素产生 margin 折叠**

原因：浮动元素和绝对定位元素不与其他盒子产生外边距折叠是因为元素会脱离当前的文档流，违反了上面所述的两个margin是邻接的条件同时（所以不与相邻元素margin折叠），又因为浮动和绝对定位会使元素为它的内容创建新的BFC，因此该元素和子元素所处的BFC是不相同的（因此不会与子元素margin折叠）。

**（28）alert与console.log的区别**

|  |  |
| --- | --- |
| alert | console.log |
| 弹窗，阻塞作用，不点击确定无法继续 | 在控制台输出，无阻塞 |
| 只能输出string，如果不是，调用toString | 可以输出任何类型 |
| 不支持多个参数输出 | 支持多个参数 |

**（29）回到顶部的多种方法**

锚点、scrollTop=0、scrollTo(0,0)、scrollBy(0,-scrollTop)、scrollIntoView()

动画的可以用setTimeout、setInterval和requestAnimationFrame

**（30）requestAnimationFrame（ie9-不支持）**

回调函数在重绘之前调用

1）requestAnimationFrame会把每一帧中的所有DOM操作集中起来，在一次重绘或回流中就完成，并且重绘或回流的时间间隔紧紧跟随浏览器的刷新频率

2）在隐藏或不可见的元素中，requestAnimationFrame将不会进行重绘或回流，这当然就意味着更少的CPU、GPU和内存使用量

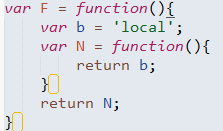
3）requestAnimationFrame是由浏览器专门为动画提供的API，在运行时浏览器会自动优化方法的调用，并且如果页面不是激活状态下的话，动画会自动暂停，有效节省了CPU开销

4）重绘频率会根据浏览器来确定，不用开发人员担心

**（31）谈一谈闭包**

概念：闭包是有权访问另一个函数作用域中变量的函数。

常见的创建方式：

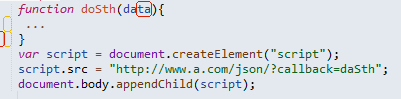
1. 嵌套的目的是造出一个局部变量
2. return N的目的是使外部能访问到N（闭包的形式有很多，但相同的是必须返回闭包，以保证外部可以访问到）

作用主要有两个：可以读取到外部函数的变量（在别处无法读取）；使这些变量一直保存在内存中；这两个用途可以用在很多地方，比如

1. 匿名自执行函数：对于只需要执行一次的函数，内部变量不需要维护，可以采用匿名自执行函数包裹，保证变量不污染全局环境（比如UI的初始化）
2. 实现封装：保证外部函数的变量在别处无法访问，只有通过闭包可以访问

【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