目录

**[一、 html以及css](#_Toc1904_WPSOffice_Level1)** **[3](#_Toc1904_WPSOffice_Level1)**

[1.1、 div+css 的布局较 table 布局有什么优点？](#_Toc449_WPSOffice_Level2) [3](#_Toc449_WPSOffice_Level2)

[1.2、 为什么利用多个域名来存储网站资源会更有效？](#_Toc12180_WPSOffice_Level2) [3](#_Toc12180_WPSOffice_Level2)

[1.3、 CSS 的盒子模型？](#_Toc26166_WPSOffice_Level2) [3](#_Toc26166_WPSOffice_Level2)

[1.4、 对 WEB 标准以及 W3C 的理解与认识](#_Toc11197_WPSOffice_Level2) [3](#_Toc11197_WPSOffice_Level2)

[1.5、 什么是外边距重叠？重叠的结果是什么？](#_Toc30044_WPSOffice_Level2) [4](#_Toc30044_WPSOffice_Level2)

**[二、js部分](#_Toc11197_WPSOffice_Level1)** **[4](#_Toc11197_WPSOffice_Level1)**

[2.1讲讲输入完网址按下回车，到看到网页这个过程中发生了什么](#_Toc17461_WPSOffice_Level2) [4](#_Toc17461_WPSOffice_Level2)

[2.2、谈谈你对前端性能优化的理解](#_Toc697_WPSOffice_Level2) [4](#_Toc697_WPSOffice_Level2)

[2.3、请你谈谈Cookie的弊端](#_Toc31601_WPSOffice_Level2) [5](#_Toc31601_WPSOffice_Level2)

[2.4、浏览器本地存储](#_Toc5165_WPSOffice_Level2) [5](#_Toc5165_WPSOffice_Level2)

[2.5、web storage和cookie的区别](#_Toc8768_WPSOffice_Level2) [6](#_Toc8768_WPSOffice_Level2)

[2.6、线程与进程的区别](#_Toc3173_WPSOffice_Level2) [6](#_Toc3173_WPSOffice_Level2)

[2.7、请说出三种减少页面加载时间的方法](#_Toc31048_WPSOffice_Level2) [7](#_Toc31048_WPSOffice_Level2)

[2.8、请解释一下 JavaScript 的同源策略](#_Toc14169_WPSOffice_Level2) [7](#_Toc14169_WPSOffice_Level2)

[2.9、请介绍下cache-control](#_Toc6887_WPSOffice_Level2) [7](#_Toc6887_WPSOffice_Level2)

[2.10、一次js请求一般情况下有哪些地方会有缓存处理？](#_Toc21289_WPSOffice_Level2) [8](#_Toc21289_WPSOffice_Level2)

[2.11、简述一下src与href的区别](#_Toc3344_WPSOffice_Level2) [8](#_Toc3344_WPSOffice_Level2)

[2.12、html5有哪些新特性、移除了那些元素？](#_Toc21767_WPSOffice_Level2) [8](#_Toc21767_WPSOffice_Level2)

[2.13、Promise实现ajax](#_Toc778_WPSOffice_Level2) [9](#_Toc778_WPSOffice_Level2)

[2.14、钩子函数](#_Toc29303_WPSOffice_Level2) [9](#_Toc29303_WPSOffice_Level2)

[2.15、es6的继承与es5的继承](#_Toc22082_WPSOffice_Level2) [10](#_Toc22082_WPSOffice_Level2)

#### html以及css

##### div+css 的布局较 table 布局有什么优点？

改版的时候更方便 只要改 css 文件。

页面加载速度更快、结构化清晰、页面显示简洁。

表现与结构相分离。

易于优化（seo）搜索引擎更友好，排名更容易靠前

* 1. **为什么利用多个域名来存储网站资源会更有效？**

a、CDN 缓存更方便

b、突破浏览器并发限制

c、节约 cookie 带宽

d、节约主域名的连接数，优化页面响应速度

e、防止不必要的安全问题

* 1. **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).

* 1. **对 WEB 标准以及 W3C 的理解与认识**

标签闭合、标签小写、不乱嵌套、提高搜索机器人搜索几率、使用外 链 css 和 js 脚本、结构行为表现的分离、文件下载与页面速度更快、内容能被更多的用户所访问、内容能被更广泛的设备所访问、更少的代码和组件，容易维 护、改版方便，不需要变动页面内容、提供打印版本而不需要复制内容、提高网站易用性

* 1. **什么是外边距重叠？重叠的结果是什么？**

在 CSS 当中，相邻的两个盒子（可能是兄弟关系也可能是祖先关系）的外边距可以结合成一个单独的外边距。这种合并外边距的方式被称为折叠，并且因而所结合成的外边距称为折叠外边距。

折叠结果遵循下列计算规则：

1. 两个相邻的外边距都是正数时，折叠结果是它们两者之间较大的值。

2. 两个相邻的外边距都是负数时，折叠结果是两者绝对值的较大值。

3. 两个外边距一正一负时，折叠结果是两者的相加的和。

#### 二、js部分

##### 2.1讲讲输入完网址按下回车，到看到网页这个过程中发生了什么

1. 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

##### 2.2、谈谈你对前端性能优化的理解

a. 请求数量：合并脚本和样式表，CSS Sprites，拆分初始化负载，划分主域

b. 请求带宽：开启GZip，精简JavaScript，移除重复脚本，图像优化，将icon做成字体

c. 缓存利用：使用CDN，使用外部JavaScript和CSS，添加Expires头，减少DNS查找，配置ETag，使Ajax可缓存

d. 页面结构：将样式表放在顶部，将脚本放在底部，尽早刷新文档的输出

e. 代码校验：避免CSS表达式，避免重定向

##### **2.3、请你谈谈Cookie的弊端**

a. 每个特定的域名下最多生成的cookie个数有限制

b. IE和Opera 会清理近期最少使用的cookie，Firefox会随机清理cookie

c. cookie的最大值大约为4096字节，为了兼容性，一般不能超过4095字节

d. 安全性问题。如果cookie被人拦截了，那人就可以取得所有的session信息。

##### **2.4、浏览器本地存储**

在HTML5中提供了sessionStorage和localStorage。

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，是会话级别的存储。

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。

**2.5、web storage和cookie的区别**

a. Cookie的大小是受限的

b. 每次你请求一个新的页面的时候Cookie都会被发送过去，浪费了带宽

c. cookie还需要指定作用域，不可以跨域调用

d. Web Storage拥有setItem,getItem等方法，cookie需要前端开发者自己封装setCookie，getCookie

e. 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

f. IE7、IE6中的UserData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

##### **2.6、线程与进程的区别**

a. 一个程序至少有一个进程，一个进程至少有一个线程

b. 线程的划分尺度小于进程，使得多线程程序的并发性高

c. 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

d. 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

e. 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

##### 2.7、请说出三种减少页面加载时间的方法

a. 尽量减少页面中重复的HTTP请求数量

b. 服务器开启gzip压缩

c. css样式的定义放置在文件头部、Javascript脚本放在文件末尾

d、压缩合并Javascript、CSS代码

##### **2.8、请解释一下 JavaScript 的同源策略**

同源策略指的是：协议，域名，端口相同，同源策略是一种安全协议。

指一段脚本只能读取来自同一来源的窗口和文档的属性。

##### **2.9、请介绍下cache-control**

每个资源都可以通过 Cache-Control HTTP 头来定义自己的缓存策略

Cache-Control 指令控制谁在什么条件下可以缓存响应以及可以缓存多久

Cache-Control 头在 HTTP/1.1 规范中定义，取代了之前用来定义响应缓存策略的头（例如 Expires）。

##### **2.10、一次js请求一般情况下有哪些地方会有缓存处理？**

a. 浏览器端存储

b. 浏览器端文件缓存

c. HTTP缓存304

d. 服务器端文件类型缓存

e. 表现层&DOM缓存

**2.11、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

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

**2.12、html5有哪些新特性、移除了那些元素？**

a. 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。

b. 拖拽释放(Drag and drop) API

c. 语义化更好的内容标签（header,nav,footer,aside,article,section）

d. 音频、视频API(audio,video)

e. 画布(Canvas) API

f. 地理(Geolocation) API

g. 本地离线存储 localStorage 长期存储数据，浏览器关闭后数据不丢失

h. sessionStorage 的数据在页面会话结束时会被清除

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

j. 新的技术webworker, websocket等

**移除的元素：**

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

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

**2.13、Promise实现ajax**

****

**2.14、钩子函数**

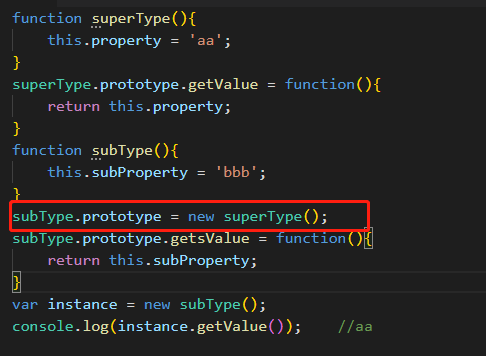
将需要执行的函数或者其他一系列动作注册到一个统一的入口，程序通过调用这个钩子来执行已经注册的函数

**2.15、es6的继承与es5的继承**

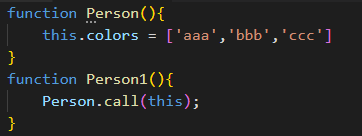
es5继承的方法

1. 对象的原型等于另一个类型的实例

Person.prototype = new Person1();则实现了Person继承了Person1，则Person的实例对象可以访问Person1原型对象上的属性。



1. 利用构造函数



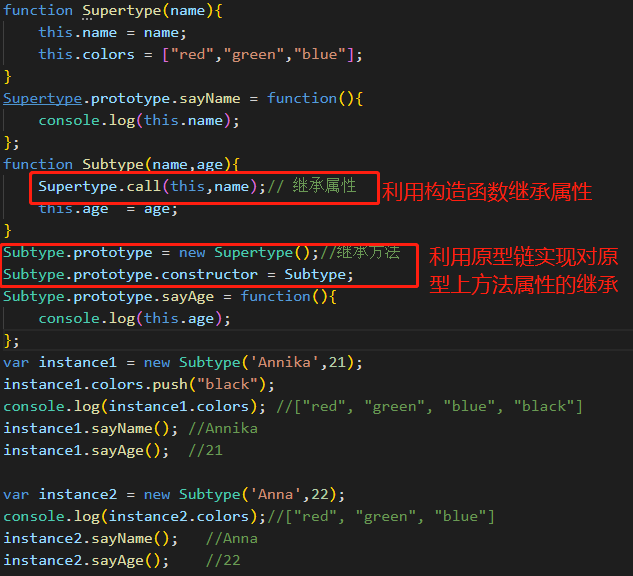
优点：可以传参，

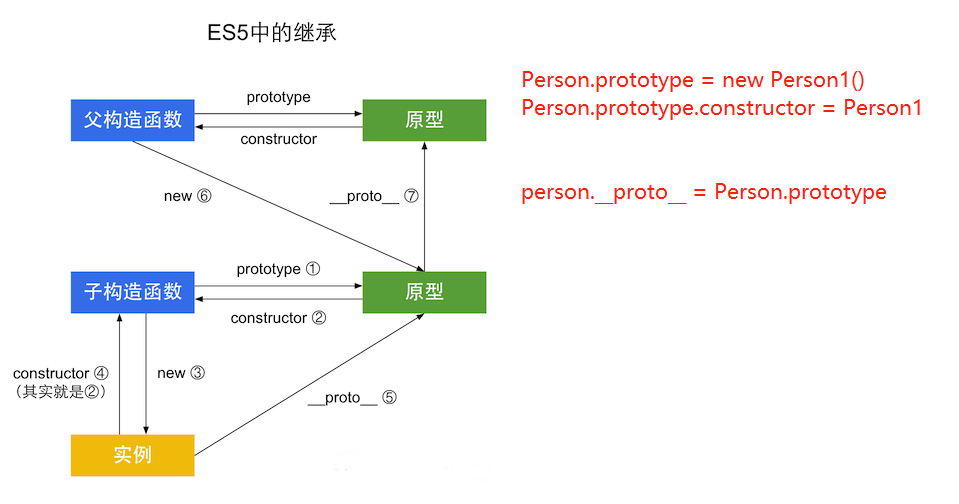
缺点：方法都在构造函数中定义，函数无法复用

在超类型中定义的方法，子类型不可见，结果所有类型都只能使用构造函数模式

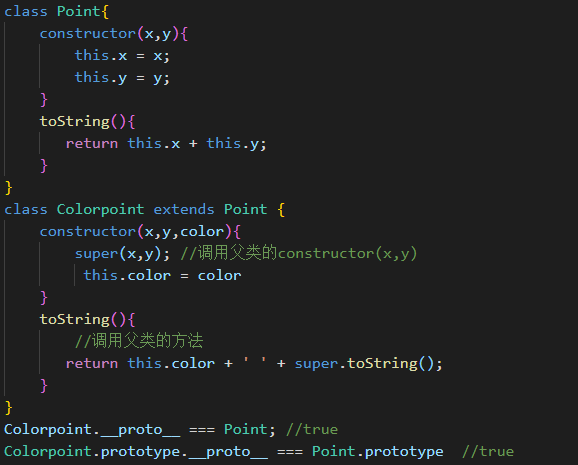
1. 原型链结合构造函数

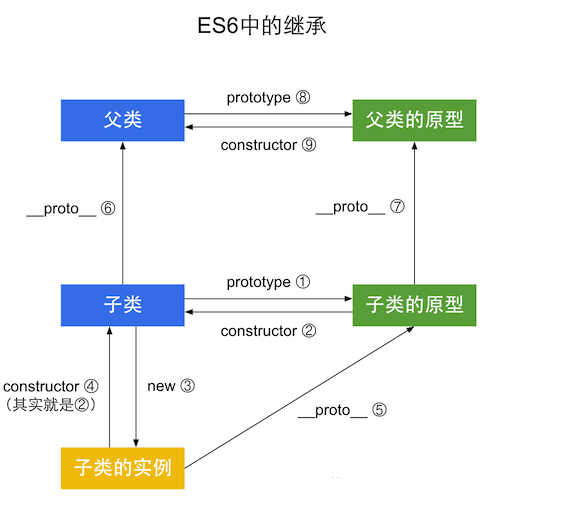
使用原型链实现对原型属性和方法的继承，用借用构造函数模式实现对实例属性的继承





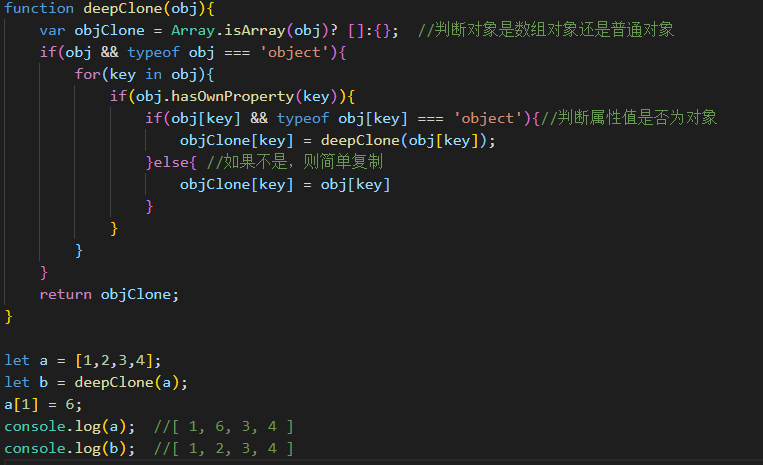
es6继承



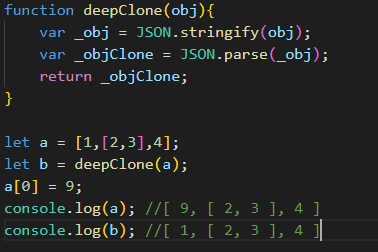


**2.16、深刻拷贝和浅拷贝**

（1）自定义方法



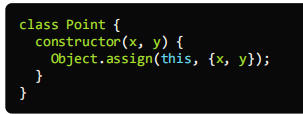
1. 利用Json对象的parse和stringfy



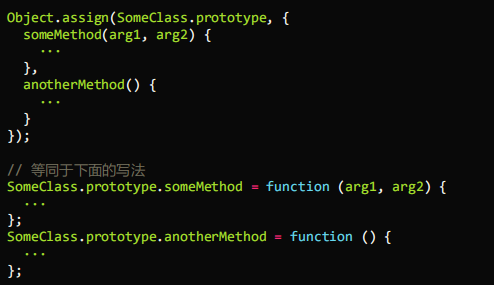
**2.17、Object.assign**

（1）为对象添加属性

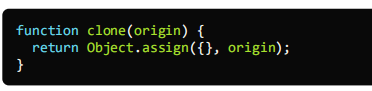
为类Point 添加属性x 和 y



（2）为对象添加方法



1. 克隆对象（浅拷贝）



保持继承链



1. 合并多个对象
2. 为属性指定默认值