## CSS

#### 伪类和伪元素分别有哪些？单引号写法和双引号有什么不同？它们有什么不同？

伪类：active focus hover link visited first-child lang

伪元素：first-letter first-line before after

单引号是伪类的写法，双引号是伪元素的写法，为了兼容旧版浏览器才有单引号伪元素的写法

伪类用于当已有的元素处于某个状态时，为其添加对应的样式，这个状态是根据用户行为而动态变化的。

伪元素用于创建一些不在文档树中的元素，并为其添加样式。它们允许我们为元素的某些部分设置样式。

#### Css3新增特性？新增伪类有哪些？

圆角、多列布局、阴影和反射、文字特效、文字渲染、线性渐变、transform、transition、aniamtion

nth-child、nth-last-child、last-child、only-child、nth-of-type、last-of-type、only-of-type、empty、target、not(elem)、enabled、disabled、checked

#### 设置margin和padding为百分比值的时候，是以哪个属性的百分比算？

以父级宽度的百分比计算

#### 说出可以上下左右居中的方法？

Flex布局

position: absolute; top: 0; left:0; bottom: 0; right: 0; margin: auto

position: absolute; top: 50%; left:50%;transform:translate(-50%,-50%)

#### css如何画一个三角形？如何画一个太极图？怎么实现0.5px边框？

三角形：宽高设置为0，三边边框颜色设置为透明，一个边填充颜色

太极图：用border-radius：50%画几个颜色大小不一样的园，再定位到不同位置

0.5px：用伪元素画1px边框，再用transform缩小

#### 简单介绍使用图片base64编码的优点缺点？

优点：减少http请求，

缺点：编码大小比源文件大1/3，无法直接缓存，ie8以前不支持

#### 有没有使用过CSS预处理器？描述下项目中如何使用它们

## JAVASCRIPT

#### Es6数组去重



#### Mouseenter和mouseover的区别

不论鼠标指针穿过被选元素或其子元素，都会触发 mouseover 事件，对应 mouseout。

只有在鼠标指针穿过被选元素时，才会触发 mouseenter 事件，对应 mouseleave。

#### 介绍一下你对浏览器内核的理解

内核主要分成两部分：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。

#### 请描述一下 cookies，sessionStorage 和 localStorage 的区别

共同点

都是保存在浏览器端，且同源的。

区别

cookie 数据始终在同源的 http 请求中携带（即使不需要），即 cookie 在浏览器和服务器间来回传递。

而 sessionStorage 和 localStorage 不会自动把数据发给服务器，仅在本地保存。

cookie 数据还有路径（path）的概念，可以限制 cookie 只属于某个路径下。

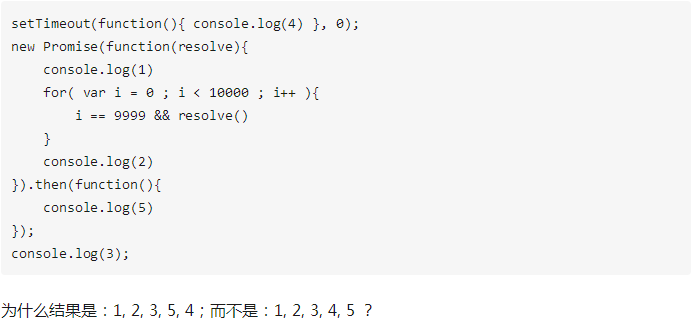
存储大小限制也不同，cookie 数据不能超过 4k，同时因为每次 http 请求都会携带 cookie，所以 cookie 只适合保存很小的数据，如会话标识。

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。

数据有效期不同，sessionStorage：仅在当前浏览器窗口关闭前有效，自然也就不可能持久保持；localStorage：始终有效，窗口或浏览器关闭也一直保存，因此用作持久数据；cookie 只在设置的 cookie 过期时间之前一直有效，即使窗口或浏览器关闭。

作用域不同，sessionStorage 在不同的浏览器窗口中不共享，即使是同一个页面；cookie 和 localStorage 在所有同源窗口中都是共享的。

#### Promise 的队列与 setTimeout 的队列有何关联 ？



js 里面有宏任务（macrotask）和微任务（microtask）。

因为 setTimeout 是属于 macrotask 的，而整个 script 也是属于一个 macrotask，promise.then 回调是 microtask，执行过程大概如下：

由于整个 script 也属于一个 macrotask，由于会先执行 macrotask 中的第一个任务，再加上 promise 构造函数因为是同步的，所以会先打印出 1 和 2；

然后继续同步执行末尾的 console.log(3) 打印出 3；

此时 setTimeout 被推进到 macrotask 队列中， promise.then 回调被推进到 microtask 队列中；

由于在第一步中已经执行完了第一个 macrotask ，所以接下来会顺序执行所有的 microtask，也就是 promise.then 的回调函数，从而打印出 5；

microtask 队列中的任务已经执行完毕，继续执行剩下的 macrotask 队列中的任务，也就是 setTimeout，所以打印出 4。

#### 箭头函数和普通函数的区别

1,箭头函数是匿名函数，不能作为构造函数，不能使用new

2,箭头函数不绑定arguments，取而代之用rest参数...解决

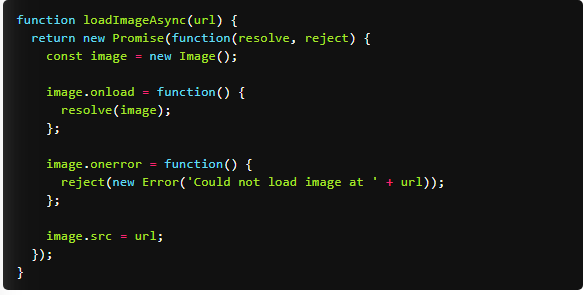
3,箭头函数不绑定this，会捕获其所在的上下文的this值，作为自己的this值

4,箭头函数通过 call() 或 apply() 方法调用一个函数时，只传入了一个参数，对 this 并没有影响。

5,箭头函数没有原型属性

6,箭头函数不能当做Generator函数,不能使用yield关键字

#### 如何用promise封装一个图片加载功能



#### 简述async/await

#### MVC,MVVC,MVVM模式的理解

##### Mvc

Model(模型):数据层，负责存储数据。  
View(视图):展现层，用户所看到的页面  
Controller(控制器):协调层，负责协调Model和View，根据用户在View上的动作在Model上作出对应的更改，同时将更改的信息返回到View上。  
三者之间的关系  
Controller可以直接访问Model，也可以直接控制View,但是Model和View不能相互通信，相当于COntroller就是介于这两者之间的协调者。

##### Mvvm

Model(模型):数据层，负责存储数据。  
View(控制器):就是ViewController层，他的任务就是从ViewModel层获取数据，然后显示。  
ViewModel(视图模型):就是View和Model层的粘合剂，封装业务逻辑处理，封装网络处理，封装数据缓存。就是把原来ViewController层的业务逻辑和页面逻辑等剥离出来放到ViewModel层

##### Mvvc

Model(模型):数据层，负责存储数据。  
View(视图):展现层，创建需求创建cell  
View(视图):定义数组，用来接收控制中的数据。处理回调（比如:刷新回调、点击cell回调、加载更多回调、动态视图高度回调等等）

Controller(控制器):加载网络数据懒加载

#### 简述Vue的响应式原理

当一个Vue实例创建时，vue会遍历data选项的属性，用 Object.defineProperty 将它们转为 getter/setter并且在内部追踪相关依赖，在属性被访问和修改时通知变化。

每个组件实例都有相应的 watcher 程序实例，它会在组件渲染的过程中把属性记录为依赖，之后当依赖项的 setter 被调用时，会通知 watcher 重新计算，从而致使它关联的组件得以更新

#### Vue生命周期

beforeCreate（创建前）、created（创建后）、beforeMount(载入前)、mounted（载入后）、beforeUpdate（更新前）、updated（更新后）、beforeDestroy（销毁前）、destroyed（销毁后）

#### computed和watch的区别和使用场景

#### 简述vuex

#### 简述父子通信，多层级子组件如何调用根级父组件方法？

#### $router和$route的区别