**ES6**

[一、 谈谈css3Flex布局 1](#_Toc27444)

[二、 浅谈session,cookie,sessionStorage,localStorage 2](#_Toc1861)

[1.1 Cookie 2](#_Toc7857)

[三、 px,em,rem区别 2](#_Toc16195)

[四、 animation和transiton 2](#_Toc27649)

[五、 Css3写一个三角箭头 3](#_Toc12699)

[六、 js闭包 3](#_Toc22464)

[1.1 执行的过程 3](#_Toc12580)

[七、 前端跨域 3](#_Toc28528)

[八、 节流防抖 4](#_Toc2899)

[九、 This的指向 4](#_Toc20672)

[十、 http三次握手协议 4](#_Toc7399)

[十一、 http常见状态吗 4](#_Toc27674)

# 谈谈css3Flex布局

# [浅谈session,cookie,sessionStorage,localStorage](https://www.cnblogs.com/jing-tian/p/10991440.html)

## Cookie

Cookie是被存储在浏览器客户端，有大小和个数限制生命周期随浏览器的关闭而结束。当然可以设置过期的时间！

session用于保存每个用户的专用信息，变量的值保存在服务器端,通过SessionID来区分不同的客户

WebStorage:localStorage（本地存储）和sessionStorage（会话存储）。

localStorage:localStorage的生命周期是永久的，关闭页面或浏览器之后localStorage中的数据也不会消失。localStorage除非主动删除数据，否则数据永远不会消失。

 sessionStorage的生命周期是在仅在当前会话下有效。浏览器关闭就没了

# px,em,rem区别

<https://www.cnblogs.com/leejersey/p/3662612.html>

px像素（Pixel）。相对长度单位。像素px是相对于显示器屏幕分辨率而言的

em是相对长度单位。相对于当前对象内文本的字体尺寸。如当前对行内文本的字体尺寸未被人为设置，则相对于浏览器的默认字体尺寸,通常要在body上设置大小，会继承父元素的大小

# animation和transiton

比如我的信息，一开始加载以后下一次进来就不需要额外重新请求服务器了

<https://www.w3school.com.cn/cssref/pr_animation.asp>

<https://www.w3school.com.cn/cssref/pr_transition.asp>

transition-property

transition-duration

transition-timing-function

transition-delay

animation-name

animation-duration

animation-timing-function

animation-delay

animation-iteration-count

animation-direction

# Css3写一个三角箭头

# js闭包

## 1.1 执行的过程

index.vue-beforeRouteLeave

router-beforeEach

proList.vue-beforeEnter

proList.vue-beforeRouteEnter

router-afterEach

App.vue-beforeUpDate

proList.vue-beforeCreate

proList.vue-created

proList.vue-beforeMount

index.vue-beforeDestroy

index.vue-destroyed

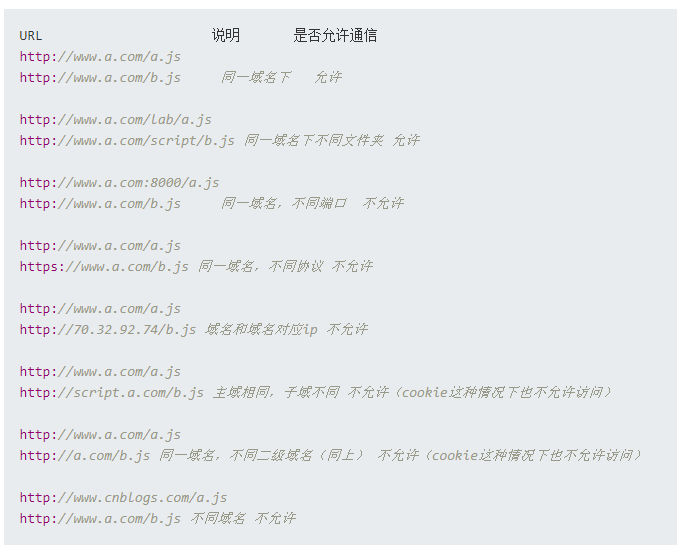
proList.vue-mounted

App.vue-updated

# 前端跨域

<https://segmentfault.com/a/1190000007326671>

这个时



# 节流防抖

function throttle(fn,time=250){

            let isRun=true;

            return function(){

                if(!isRun) return;

                isRun=false;

                setTimeout(()=>{

                    fn.apply(this,arguments);

                    isRun=true

                },time)

            }

        }

function debounce(fn, interval = 300) {

            let timeout = null;

            return function () {

                clearTimeout(timeout);

                timeout = setTimeout(() => {

                    fn.apply(this, arguments);

                }, interval);

            }

        }

# This的指向

构造函数 this 指向实例

普通函数 this 指向window

Apply call可以动态改变this指针

箭头函数

# http三次握手协议

# http常见状态吗