# 1改变this指向的方法

**Call() apply() bind()**

**call、apply与bind区别：前两个可以自动执行，bind不会自动执行，需要手动调用**

**call、bind与apply区别：前两个都有无数个参数，apply只有两个参数，而且第二个参数为数组**

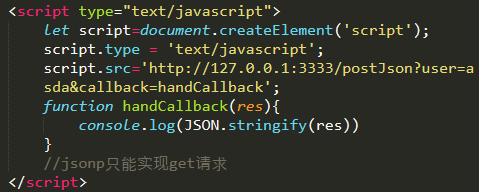
# 2.跨域

**跨域问题是由于javascript语言安全限制中的同源策略造成的. 同源策略是指一段脚本只能读取来自同一来源的窗口和文档的属性,这里的同一来源指的是主机名、协议和端口号的组合.（协议、域名、端口有任何一个不同，都会造成跨域）**

**解决方法：**

1. **Jsonp：浏览器允许js、css、img等静态资源从不同域名引入，基于此原理通过动态创建script标签来实现跨域通信，jsonp只能实现get请求**

**Scrip的src为请求地址加返回函数如：**



1. **Vue中proxy跨域解决原理：将域名发送给本地的服务器（启动vue项目的服务,loclahost:8080），再由本地的服务器去请求真正的服务器。**

**方法：在vue根目录下的vue.config.js中设置proxy跨域代理，设置内容api对象里的target（后台地址）、changOrigin（允许跨域）、parthRewrite（路径重写）**

# 3.js兼容IE

|  |  |  |
| --- | --- | --- |
|  | **普通网页** | **IE** |
| **事件冒泡** | **StopPropagation()** | **cancelBubble=true** |
| **阻止默认事件** | **PreventFefault()** | **returnValue=false** |

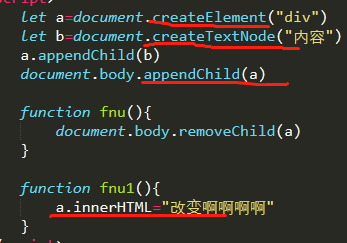
判断各种浏览器



# 4.css3新特性

**过度，transition、动画animation、形状转换器transform、阴影box-show、边框图片border-image、word-break、text-overflow超出省略、flex弹性布局**

# 5.原生方法操控元素



# 6. 浏览器从输入地址到页面渲染发生了什么

**1、浏览器（客户端）进行地址解析。**

**2、将解析出的域名进行dns解析。**

**3、通过ip寻址和arp，找到目标（服务器）地址。**

**4、进行tcp三次握手，建立tcp连接。**

**5、浏览器发送数据，等待服务器响应。**

**6、服务器处理请求，并对请求做出响应。**

**7、浏览器收到服务器响应，得到html代码。**

**8、渲染页面。**

# 7.css引入

**Link :1属于xhtml2.优先加载css文件到页面3. Link引入便于维护，会出现加载中断，实现了结构与样式的分离，没有兼容问题**

**@import:1.属于css2的2.先加载html结构在加载css文件**

# 8.es6新特性

## **1.Const let**:

**1.块级作用域**

**2.let变量可以重复赋值，const不可以**

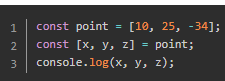
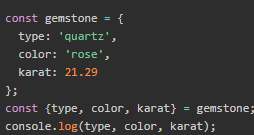
## **2.模板字面量**

**Es6之前拼接字符串的方法是”+”或者concat（）**

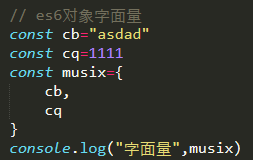
**用倒引号`${ }`(再加上占位符)**

## **3.结构赋值**

**对数组结构用【】，对对象结构用{}**

## **4.对象字面量**



## **5.for（const digit of digits）**

**可以忽略索引,随时停止或退出**

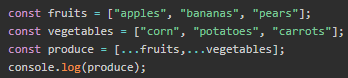
**for循环需要耿总技术器和退出条件**

**for(const digit in digits)需要使用index访问数组值**

## **6.展开运算符 ...**

**1.可以使字面量对象展开为多个元素**

**2.结合数组**



## **7.箭头函数**

**普通函数：1.可以是函数声明或者函数表达式2.this指向调用他的对象**

**尽头函数：1.始终是表达式**

**2.是匿名函数，不能作为构造函数，不能使用new**

**3.不绑定this，会捕获所在上下文的this，作为自己的this值，无法改变this的指向**

# 9数据深拷贝

**就是不管里面多少层，都遍历，克隆一个与旧不相关的，修改新的不影响旧的。**

**原数组里的数据包含引用类型**

**方法：1.递归 2.通过let a2=JSON.Parse( JSON.stringify(a1))**

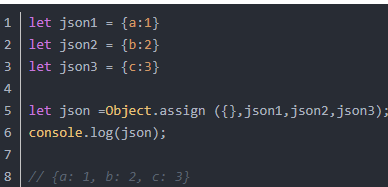
# 10数据浅拷贝

**只遍历一层，如果存在数组成员是对象，[{name:'jack'}],不会对对象里的值进行遍历拷贝**

**浅拷贝方法：**

**a1=[1,2,3]**

1. **直接赋值 ： let a2=a1**
2. **拓展运算符: let a2=[...a1]**
3. **assign(): let arr2 = Object.assign([],a1);**

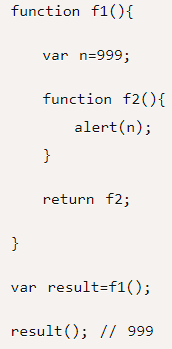


1. **let arr2 = [].concat(a1);**
2. **slice() let arr2 = arr1.slice(0);**

# 11.闭包

**①Js语言特点：**

**变量作用域：全局和局部变量，函数内部可以访问全局，但外部无法访问函数内部的局部变量**

**f1无法访问f2函数内的变量，但是f2可以访问f1的，把f2作为返回值，那么f1就可以在外部访问到，解决了全局无法访函数内部局部作用域**

1. **闭包作用1.读取函数内部的变量2.让这些变量始终保持在内存中**

**③闭包缺点：1内存消耗大2.导致内存泄漏（在函数退出前把不用的变量删除了）**

# 12.递归

**递归：函数自己调用自己**