1.h5新标签

<nav></nav>

<header></header>

<article></article>

<aside></aside>

<section></section>

<figure></figure>

<figcaption></figcaption>

<footer></footer>

css3新特性：

圆角边框，背景边框，文字阴影，盒子阴影，2D/3D转换，背景渐变，背景图片，字体定义，弹性盒子

新增属性选择器，新增结构伪类选择器，新增状态伪类选择器

ES5新API：

Array.forEach();Array.filter();Array.some();Array.every();Array.map()

2.事件对象兼容性写法

e=window.event || e;

事件对象Target兼容性写法

target=e.target || event.srcElement

阻止默认事件兼容性写法

if(e.preventDefault){

e.preventDefault()

}else{

e.returnValue=false;

}

阻止冒泡事件兼容性写法

if(e.stopPropagation){

e.stopPropagation()

}else{

e.cancelbubble=true

}

3.三层结构渲染（自己白话叙述）

浏览器在对html文件进行解析时，是按照自上而下的流方式进行加载的。

在遇到css引入标签link或者js引入标签随即立刻停止html的解析，执行对css的代码解析或者js代码的解析，直到解析完毕。再对html进行解析

css与js不同，页面样式的渲染需加载在最前端，这样在html解析的同时，样式一并应用到页面。

而js一般在最末端引入，等待html DOM解析完毕再对其行为样式进行渲染。

渲染过程中，若出现元素脱离文档流，此元素的上下结构布局会发生改变，在渲染时，浏览器会对其重新渲染以达到预期效果。此现象称为回流。

4.作用域链与原型链（白话自述）

通常函数执行都会存在作用域，就是其执行环境。正是作用域的存在，才出现的局部变量与全局变量。

而作用域链时在作用域的基础上而形成的 一系列局部作用域，其关系是从属关系，从而形成了作用域链，作用域链上的变量是以自下而上的方式进行访问的，

即先在局部作用域进行访问，若无，则向上寻找，直至访问到全局作用域。若再无，则为未定义。

原型链就是实例化对象的原型组成的链，若一个实例化对象的原型是另一个对象A，而此对象A也存在原型B，以此类推，直到找到原型Object。而Object

的原型是null。这样便形成了原型链，在访问属性或者方法的过程中，先在实例化对象的自身环境中寻找，再沿原型链往上找，直到找到。若无，则未定义。

5. //set方法

let arr1=[1,2,3]

let arr2=[2,3,4]

let newarr=[...arr1,...arr2]

console.log(newarr)

let lastarr=new Set(newarr)

console.log(lastarr)

//indexOf判断

let arr1=[1,2,3]

let arr2=[2,3,4,5]

let newarr=arr1.concat(arr2)

let lastarr=[]

console.log(newarr)

for( var i = 0 ; i < newarr.length ; i++){

if(lastarr.indexOf(newarr[i])==-1){

lastarr.push(newarr[i])

}

}

console.log(lastarr)

//先排序，再进行相邻判断

let arr1 = [1, 2, 3,"1"];

let arr2 = [2, 3,"1", 4, 5];

let newarr = arr1.concat(arr2);

newarr.sort();

let lastarr = [];

console.log(newarr);

for(var i = 0; i < newarr.length; i++) {

if(!Object.is(newarr[i],newarr[i+1])) {

lastarr.push(newarr[i])

}

}

console.log(lastarr)

6.json对象是一个格式，用于数据交互。

其属性名必须加引号，

其可以跨平台传输，速度快

属性不能是方法，且不能是undefined

js对象是实例化对象，存在原型

不可跨平台传输

键值对的方式出现

键的类型多样化，数值，字符串，对象，方法，boolean

7.bar bar undefined bar

8.url请求过程

第一步：网址的请求同时，远程服务器启动DNS查询，使得浏览器获得相应的IP地址

第二步：浏览器与服务器通过TCP进行三次握手建立连接，分别是同步报文，同步应答报文，应答报文。

第三步：建立连接，浏览器发送HTTP请求

第四部：服务器响应请求，并且返回数据

第五步：浏览器接受数据通过js解析

9.垃圾回收机制

标记清除：变量在执行环境中存在两种状态，当变量进入执行环境是，就标记这个变量为“进入环境”。当变量离开环境时，则将其标记为“离开环境”。

此标记时即将被回收的标记。

引用计数：引用计数的含义是跟踪记录每个值被引用的次数。当声明了一个变量并将一个引用类型赋值给该变量时，则这个值的引用次数就是1。相反，如果包含对

这个值引用的变量又取得了另外一个值，则这个值的引用次数就减1。当这个引用次数变成0时，则说明没有办法再访问这个值了，因而就可以将其所占的内存空间给收回来。

这样，垃圾收集器下次再运行时，它就会释放那些引用次数为0的值所占的内存。

10.短路运算符

&&与|| 运算符

如果逻辑与运算符左边的值布尔转换后为true，那么返回右边的值（不管右边的值是真还是假）。

但是当逻辑与的左边为 null/NaN/undefined ，结果就会得到null/NaN/undefined。

如果逻辑或运算符左边的值布尔转换后为false，那么返回右边的值（不管右边的值是真还是假）。