**学习内容**

**1.DOM**

**2.BOM**

**3.js对象：属性和方法。**

**属性：js对象是属性变量的容器**

**js对象是键值对的容器。**

**键值对通常写法为 name : value (键与值以冒号分割)。键值对在 JavaScript 对象通常称为 对象属性。**

**方法：对象的方法定义了一个函数，并作为对象的属性存储。**

**对象方法通过添加 () 调用 (作为一个函数)。**

**Java对象：变量（属性）和方法（行为）**

**（这周ssd1刚开始讲java和面向对象相关的知识，感觉有点点相似与不同）**

**面向对象：模拟现实中生活方式，具有通用性。**

**4.this：在js中随执行环境改变而改变**

在方法中，this 表示该方法所属的对象。

如果单独使用，this 表示全局对象。

在函数中，this 表示全局对象。

在函数中，在严格模式下，this 是未定义的(undefined)。

在事件中，this 表示接收事件的元素。

类似 call() 和 apply() 方法可以将 this 引用到任何对象。

5.**全局、局部变量**

全局变量：变量在函数内没有声明（没有使用 var 关键字），该变量为全局变量

## JavaScript 变量生命周期

JavaScript 变量生命周期在它声明时初始化。

局部变量在函数执行完毕后销毁。

全局变量在页面关闭后销毁。**//今天上C++也讲了变量的生命期**

**学习过程**

一：

了解了\*{margin: 0; padding: 0}的作用Reset（重置）浏览器默认样式，对于各浏览器样式统一的话有着简单粗暴的效果。这东西叫“通配符”用来匹配页面上所有元素。

\*{margin:0; padding:0;}代表了网页中所有元素，包括body ,ul, li列表标签 ,p,H标题标签,dd,dt 等……都有默认的margin 或padding值的，加上这句就可以删除浏览器这些默认值，方面后面的设置。

**//感觉存在弊大于利并且这次做得很粗暴，所以没用www**

二：

a {

text-decoration: none;

}

CSS文字修饰为无;一般用于设置清除超链接的默认下划线

三：加class=“on”

为了方便操作，如果加了"on"，即说明当前图片正在轮播 **//删除之后影响感觉不大呀……**

三：

偏移量

反复调图片宽高wsl，偏移量要和图片宽度一样奥。

四： a href="javascript:;" 是为了防止多次点击,并且为了防止跳出链接

**上次课总结**

**1.flex**

**轴**

**容器**

**2.动画**

**Animation**

**Keyframe**

**Transition**

**Transiate**

**3.webkit-appearance：使**像标准的用户界面元素

**4.dom/bom**

**5.面向对象**

**6.弱类型：** 弱类型定义语言：

数据类型可以被忽略的语言。它与强类型定义语言相反, 一个变量可以赋不同数据类型的值。   
强类型定义语言在速度上可能略逊色于弱类型定义语言，但是强类型定义语言带来的严谨性能够有效的避免许多错误。

**7.MVC：**

**Model（模型）** - 模型代表一个存取数据的对象或 JAVA POJO。它也可以带有逻辑，在数据变化时更新控制器。

**View（视图）** - 视图代表模型包含的数据的可视化。

**Controller（控制器）** - 控制器作用于模型和视图上。它控制数据流向模型对象，并在数据变化时更新视图。它使视图与模型分离开。

**MVVC：M： Model，它是与应用程序的业务逻辑相关的数据的封装载体，它是业务领域的对象，Model并不关心会被如何显示或操作，所以模型也不会包含任何界面显示相关的逻辑。也就是Angular中的Service**

**V: View，它专注于界面的显示和渲染，在angular中则是包含一堆声明式Directive的视图模板。**

**V(M)： ViewModel，它是View和Model的粘合体，负责View和Model的交互和协作，它负责给View提供显示的数据，以及提供了View中Command事件操作Model的途径；也就是Angular中的scope**

**Controller：这并不是MVVM模式的核心元素，但它负责ViewModel对象的初始化，它将组合一个或者多个service来获取业务领域Model放在ViewModel对象上，使得应用界面在启动加载的时候达到一种可用的状态。**

**8.**css3为了区分伪类和伪元素，伪元素采用双冒号写法。

**::after和::before**

用于在css渲染中向元素逻辑上的头部或尾部添加内容。

这些添加不会出现在DOM中，不会改变文档内容，不可复制，仅仅是在css渲染层加入。

所以不要用:before或:after展示有实际意义的内容，尽量使用它们显示修饰性内容