# 第三天

1. 创建元素的方式

第一种

document.wirte(要输入的内容) ：通过js往文档页面找那个写入内容

注： 1.写的sctipt可以在页面的位置，但是打开程序执行的时候，它自动把script标签放到body闭合标签前面 但是如果放在head标签里面，就不会被移动到body里面

2.这种方法不会覆盖之前的内容，但是上面的方法无法往某一个具体元素里面添加内容，比如，给p标签添加一个元素

3.如果在事件里面执行这个写入方法，它会把之前的页面上和已经添加进去的内容全部清除（页面上有多少就覆盖多少）

4.文档流，打开页面的的第一次，文档流会开始，页面的内容全部会加加进去整个页面，整个页面加载完毕之后，文档流会关闭。当单机事件的时候会在此开启一次文档流，但是会把之前的内容全部清空

优点 ： 操作简便

缺点： 在事件里面写该方法的时候它会把之前的内容全部清除，（因为他会开启一个新的文档流）2，不能往具体某个元素添加内容

一般不推荐使用第一种方法

第二种方法

元素.innerHTML= “” 可以设置读取或设置某个元素或者body里面内容

1.会覆盖之前的内容，使用.innerHTML+= “”之后就不会覆盖

2. js可以直接通过document下面直接读取body title head

缺点： 循环大量拼接字符串，性能比较差

优化： 当拼接多个字符串的时候 先在内存里面创建一个变量，存起来这一堆字符串，然后再加入到页面中

第三种方法

document.createElement(‘标签名’)； 先创建标签元素节点

某元素.appendChild(‘你添加的元素’) 再往里面放 放到节点的最后面

优点： 性能上较好，和优化后的innerHEML性能差不多

缺点：操作繁琐

总结：如果大量数字字符串拼接，使用innerHTML优化后的，推荐

如果要是生成内容数量较少，建议使用document.createElement();

二 . 获取和设置样式函数封装

1. js直接获取或者设置行间的样式 div.style.color =’ ’;

无法获取样式表中的样式

1. 标准浏览器及ie9以上，js提供了一个方法，获样式表中的样式

**getComoutedStyle(元素).属性**  获取该元素的某某属性

getComoutedStyle的主人可以看成是widow这个对象

注：ie 6,7,8，不支持上述方法，他们有自己的获取方法: **元素.currentStyle.属性**

方法ie9以上也可以支持但是火狐，谷歌，不支持 会报错，或者undefined；

兼容函数的封装

形参相当于局部变量 ，变量不能直接跟在点的后面

变量不能直接跟在.后面 只能[变量这么使用]；

函数执行完之后会立马返回，可以用变量接收这个值

1. dom 节点
2. 节点概念：

节点：

1. 作用： 整个文档都可看成
2. 整个页面的节点结构，可以看成一个树
3. 节点的类型（12种）包括 dom (xml 可扩展标记语言)和（html 超文本标记语言）
4. xml :一般用来做
5. childNades:只读属性子节点列表，可以获取直接子节点，不能获取孙级以及以下节点

标准浏览器下：可以获取空白文本节点和元素节点

非标准浏览器下：只包含元素节点

1. children:方法，没有兼容性问题，可以代替childNades
2. 1. A.firstChild A的里面的第一个子节点

问题：

标准浏览器下：包含空白文本和元素节点

非标准浏览器下： 只包含元素节点

2 在标准浏览器的方法：A.firsElementChild 可以直接获取第一个元素类型的子节点

解决兼容性问题：函数封装

1. 事件对象event

在ie/chrome事件

标准浏览器都支持：事件对象到底在哪，在事件触发的时候，作为事件处理函数的第一个参数传去进来的

形参如果没有值，相当于未定义

mousemove 这个事件触发的频率和距离像素没有关系，和在单位事件（很短的事件内）判断这一次的位置，和上一次的位置