函数封装，就一句话：

相同的，重复，封起来；

不同的，要改的，传进去

# DOM操作

我们的JS分为三个部分组成

ECMAscript DOM BOM

核心（ECMAscript）欧洲计算机制造协会

描述了JS的语法，以及基本对象，相当于制定了JS的规则，例如

var a 和var A，是不一样的；

文档对象模型（DOM） 重点

处理网页内容的方法和接口

浏览器对象模型（BOM）

和浏览器的交互方法以及接口

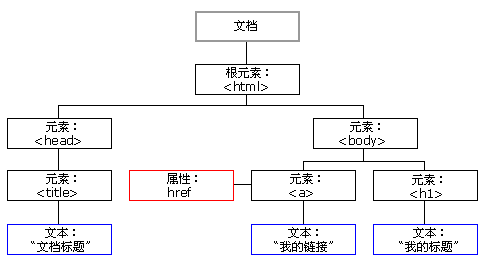
例如，用浏览器获取用户地址。。。。。

* 1. DOM的定义

DOM为文档提供了结构化表示，并定义了如何通过脚本访问文档结构

目的其实就是为了能够让JS操作HTML元素而制定的规范

DOM树



1.2 节点  
 标签

由结构树可以看出，整个文档，他就是一个文档节点；

每个HTML标签都是一个元素节点

标签中的文字则是文字节点

标签的属性是属性节点

页面中的一切都是节点

* 1. 访问节点

之前学过两种

getElementById()//用过ID访问节点

getElementsByTagName()//通过标签访问节点

getElementsByClassName()//通过类名获取节点，也是伪数组

但是，有小小的兼容性问题；

主流浏览器支持

IE 6 7 8不支持

怎么办？自己封装自己的类

* 1. 封装自己的class类

原理：我们去除所有的盒子，用遍历的方法，通过每个盒子的className来判断，如果相等就留下

* 1. 判断真假

我们用条件语句来判断5大数据类型的真假

数据类型 结论

数字类型： 所有数字都是真，包括负数，0是假；

字符串类型： 所有字符串都是真，” ”串为假;

对象类型： 所有对象都是真，null是假；

未定义： undefined为假，没有真（只要未定义，它就是假的，只要定义了，他就是真的，并且也就不是undefined）；

* 1. 访问关系

父节点 兄弟节点

parentNode nextSibling（下一个兄弟，一定要是写在一起的，不然就是文档流）

nextElementSibling（可以理解为兼容，也就是，只获取元素兄弟，不是元素的东西，就给他过滤掉了，例如字符串、数值，都不是元素，就给他过滤掉了）

previousSibling(上一个兄弟)

previousElementSibling

子节点获取

firstChild 第一个子节点

firstElementChild

lastChild 最后一个子节点

lastElementChild

childNodes 获取所有子节点，并且返回的是一个伪数组，他是标准属性

children 重要返回所有的子元素节点，这个更好用，常用；

节点也分三种，通常，我们只需要用到元素节点

但是，我们也要清除，哪三种节点

节点.nodeType;//获取节点类型；

nodeType==1 元素节点 也是我们常用的

nodeTpye==2 属性节点

nodeType==3 文本节点

* 1. 节点的操作

新建节点 插入节点 删除节点 克隆节点等等

* 1. 创建节点

var li=documemt.createElement(‘li’);//在文档上创建一个节点，节点是一个li元素标签；

var body=document.getElementsByTagName('body');//先，给谁添加节点，就要获取谁，现在是给body添加，就要获取body

var div=document.createElement('div');//想要添加标签，就要有才能添加，不能无中生有，所以先创建一个标签节点

body[0].appendChild(div)//给body添加子节点，子节点是div

for(var i=1;i<=5;i++){

var div=document.createElement('div');

body[0].appendChild(div)

}

//循环创建添加多个节点

appendChild()是添加在父节点的最后面

如果我们想在前面添加节点，就要用：

insertBefore（插入的节点，参照节点） 在前面添加子节点

insertBefore（两个参数都必须要设置，如果不想设置，可以设置为null）

insertBefore(想要插入的节点，参照节点)

参照节点：你设置的谁，就在谁的前面插入，如果不设置，就添加在子元素的最后

怎么添加兄弟节点：

就是给当前元素的父亲添加子节点，就相当于给当前元素添加兄弟节点；

移出节点：

removeChild()

克隆节点，就是复制节点

节点.cloneNode();

括号里面可以写参数，写true，就是深层复制，除了复制节点，还会复制子节点；

如果写false，那么只复制当前节点

如果不写参数，默认false