

day11

1.属性集合

获得元素节点的属性集合 element.attributes

返回结果 类数组对象
属性集合里面保存了所有的属性节点

每一个属性节点都是一个单独的对象

```
console.log(typeof box.attributes.title);//object  
console.log(box.attributes['class']);//class="box"
```

1

获取节点的节点名、节点类型、节点值

```
console.log(box.attributes['title'].nodeName); // 属性节点的节点名 属性名  
console.log(box.attributes['title'].nodeType); // 属性节点的节点类型 2  
console.log(box.attributes['title'].nodeValue); // 属性节点的节点值 属性值
```

1

属性的操作

1.获得属性值

方法1

element.attributes[索引值].value
弊端：索引不好确定属性的位置

方法2

element.attributes[属性名].value
过程 先获得属性集合，再通过属性名获得属性节点，最后.value获得节点值

方法3

console.log(box.attributes['id'].value)
element.getAttributeNode('属性名').value
通过获得属性节点 获得属性值

方法4（推荐）

element.getAttribute('属性名');
console.log(box.getAttribute('title'));

2.设置属性

方法1（推荐）

element.setAttribute(attrName,attrValue);
第一个为属性名，第二个为属性值

方法2

box.setAttribute('class', 'red');
element.setAttributeNode(attrNode);
例子
var attr = document.createAttribute('class');
attr.nodeValue = 'red';
box.setAttributeNode(attr);

3.删除属性

element.removeAttribute(attrName);
例子：
box.removeAttribute('with');

4.判断属性是否存在

element.hasAttribute(attrName);
返回一个布尔值
例子 console.log(box.hasAttribute('class'));

类名的切换

```
// 类名切换  
var flag = true;  
box.onclick = function() {  
  // var className = ;  
  this.setAttribute('class', flag ? 'red' : 'green');  
  flag = !flag;  
}
```

1

2.类名操作

HTML5标准 类名API

classList

1.查看标签的类名

```
console.log(box.classList);
```

1

2.添加标签的类名

```
box.classList.add('red', 'green');
```

1

如果有多个类名，那么就用逗号隔开

3.删除标签的类名

```
box.classList.remove('box');
```

1

4.元素选取

1.通过标签名选取元素

```
node.getElementsByTagName('tagName');
```

返回一个类数组对象

```
var divElms = document.getElementsByTagName('div');  
console.log(divElms);
```

1

2.通过id获取元素

```
var box = document.getElementById("id名");
```

注意：括号内是单引号的字符串

3.通过name属性获取元素

```
document.getElementsByName('name');  
var checkBox = document.getElementsByName('check');  
console.log(checkBox);
```

1

4.通过类名选取元素

```
node.getElementsByClassName();  
var t1 = document.getElementsByClassName('t1');  
console.log(t1);
```

1

5.通过CSS选择器获取元素

1.通过css选择器选择所匹配的到第一个元素 node.querySelector(selector);

2.通过css选择器选择所有匹配的元素

返回被选元素的集合(类数组对象)

node.querySelectorAll(selector);

selector

可以是id，写成'#box'
可以是class，写成 '.class'
可以是属性选择符，如以某字符开始，a[href\$="baidu.com"]

6.其他获取

1.获取html

document.documentElement

2.获取body

document.body

3.获取head

document.head

7.创建元素

1.创建元素节点

document.createElement('tagName');

2.在指定父节点的末尾添加一个子节点

parentNode.appendChild(childNode);
document.body.appendChild(div);

1

3.在指定节点之前插入一个新节点

parentNode.insertBefore(newChild,existingChild);
第一个为新增加的节点，第二个为指定在某个节点之前
var title = document.querySelector('#title');
document.body.insertBefore(div, title);

1

9.删除节点

parentNode.removeChild(childNode);

删除子节点

返回被删除的元素

```
window.onload = function() {  
  var list = document.querySelector('.list');  
  var li = document.querySelector('.list>li:nth-child(3)');  
  var temp = list.removeChild(li);  
  console.log(temp);  
  console.log(list);  
}
```

1

10.替换节点

使用新节点替换旧节点

parentNode.replaceChild(newChild,oldChild);

```
// 将p元素替换成h1  
window.onload = function() {  
  var p = document.querySelector('.tit');  
  // 新建一个元素  
  var h1 = document.createElement('h1');  
  h1.innerHTML = p.innerHTML;  
  // 复制属性  
  var attrs = p.attributes;  
  for (var i = 0; i < attrs.length; i++) {  
    h1.setAttribute(attrs[i].nodeName, attrs[i].nodeValue);  
  }  
  // 替换元素  
  document.body.replaceChild(h1, p);  
}
```

1