

15: 框架实现: ELEMENT 节



当我们为一个 DOM 设置对应属性的时候，其实分成了两种情况：

1. HTML Attributes
2. DOM Properties

那么想要搞明白，上一小节中所出现的问题的原因，我们就需要搞明白上面的两种情况指的是什么意思。

HTML Attributes

HTML Attributes, 所代表的的就是 **定义在 HTML 标签上的属性**。比如我们以上一小节渲染的 DOM 为例:

<> 代码块

```
1 <textarea class="test-class" type="text"></textarea>
```

这里 HTML Attributes 指的就是 `class="test-class"` 和 `type="text"`

DOM Properties

DOM Properties，所代表的就是在 DOM 对象上的属性，比如我们上一小节的 ta：

<> 代码块

```
1 const el = document.querySelector('textarea')
```

我们就可以通过 `obj.attr` 的形式获取对应的属性：

<> 代码块

```
1   el.type // 'textarea'
2   el.className // 'test-class'
3   el.value // 'textarea value'
```

对比

然后我们对比 `HTML Attributes` 和 `DOM Properties` 可以发现双方对于 **同样属性的描述是不同** 的。而这个也是 `HTML Attributes` 和 `DOM Properties` 之间的关键。

那么明确好了这个之后，我们再来看对应方法。根据上一小节的代码，我们可以知道，设置属性，我们一共使用了两个方法：

1. `Element.setAttribute()`: 该方法可以 设置指定元素上的某个属性值,
2. `dom.xx`: 相当于 直接修改指定对象的属性

但是这两个方式却有一个很尴尬的问题，那就是 **属性名不同**。

比如：

1. 针对于 class 获取:

```
1 HTML.Attributes.prototype.getAttribute('class')
```

索引目录

12: 深入属性挂犁

HTML Attribut

DOM Properties

对比

针对于 class

总结



2. 针对于 textarea 的 type 获取：

1. HTML Attributes : ta.getAttribute('type')
2. DOM Properties : ta.type **无法获取**

3. 针对于 textarea 的 value 获取：

1. HTML Attributes : ta.getAttribute('value') **无法获取**
2. DOM Properties : ta.value

所以为了解决这种问题，咱们就必须要能够 **针对不同属性，通过不同方式** 进行属性指定。所以：vue 才会通过一系列的判断进行处理

针对于 class

除了以上内容我们需要知道之外，还有另外一个我们需要了解的知识就是：既然 class 可以使用 setAttribute 设置，也可以通过 className 设置，那么这样就存在一个问题：我们应该通过哪种方式设定 class 呢？

我们知道对于 vue 的代码而言，在 packages/runtime-dom/src/modules/class.ts 中的 patchClass 中：

<> 代码块

```
1   if (value == null) {
2     el.removeAttribute('class')
3   } else if (isSVG) {
4     el.setAttribute('class', value)
5   } else {
6     el.className = value
7   }
```

根据以上代码可知：只要 dom 不是 svg，则通过 className 设置 class。那么为什么要这么做呢？

我们创建以下测试案例，对两者之间的性能进行一下对比：

<> 代码块

```
1   <body>
2     <div id="app"></div>
3     <div id="app2"></div>
4   </body>
5
6   <script>
7     const divEle1 = document.querySelector('#app')
8     const divEle2 = document.querySelector('#app2')
9
10    console.time('classname')
11    for (let i = 0; i < 10000; i++) {
12      divEle1.className = 'test-2'
13    }
14    console.timeEnd('classname')
15
16    console.time('attr')
17    for (let i = 0; i < 10000; i++) {
18      divEle2.setAttribute('class', 'test')
19    }
20    console.timeEnd('attr')
21  </script>
```

最后打印的结果为：

<> 代码块

```
1   classname: 1.7470703125 ms
2   attr: 3.389892578125 ms
```

由打印结果可知 className 的性能 大于 setAttribute 的性能

[意见反馈](#)

[收藏教程](#)

[标记书签](#)



总结

本小节，我们主要分析了：两大块问题。

1. `HTML Attributes` 和 `DOM Properties`：我们知道想要成功的进行各种属性的设置，那么需要 **针对不同属性，通过不同方式 完成**
2. `className` 和 `setAttribute('class', '')`：因为 `className` 的性能更高，所以我们应该尽量使用 `className` 进行指定。

11: 源码阅读: class 属性和其他属性的区分... < 上一节 下一节 > 13: 框架实现: 区分处理 ELEMENT 节点的...

 我要提出意见反馈

[企业服务](#) [网站地图](#) [网站首页](#) [关于我们](#) [联系我们](#) [讲师招募](#) [帮助中心](#) [意见反馈](#) [代码托管](#)



Copyright © 2022 imooc.com All Rights Reserved | 京ICP备 12003892号-11 京公网安备11010802030151号



 意见反馈

 收藏教程

 标记书签