

全部开发者教程 三

09: 框架实现：初步实现 watch 数据监听器

10: 问题分析：watch 下的依赖收集

11: 框架实现：完成 watch 数据监听器的依赖收集

12: 总结：watch 数据侦听器

13: 总结

第八章：runtime 运行时 - 运行时核心设计原则

01: 前言

02: HTML DOM 节点树与虚拟 DOM 树

03: 挂载与更新

04: h 函数与 render 函数

05: 运行时核心设计原则

06: 总结

第九章：runtime 运行时 - 构建 h 函数，生成 Vnode



Sunday • 更新于 2022-10-19

◀ 上一节 01: 前言 03: 挂载与更新 下一节 ▶

02: HTML DOM 节点树与虚拟 DOM 树

首先我们先来学习两个运行时的基础概念：

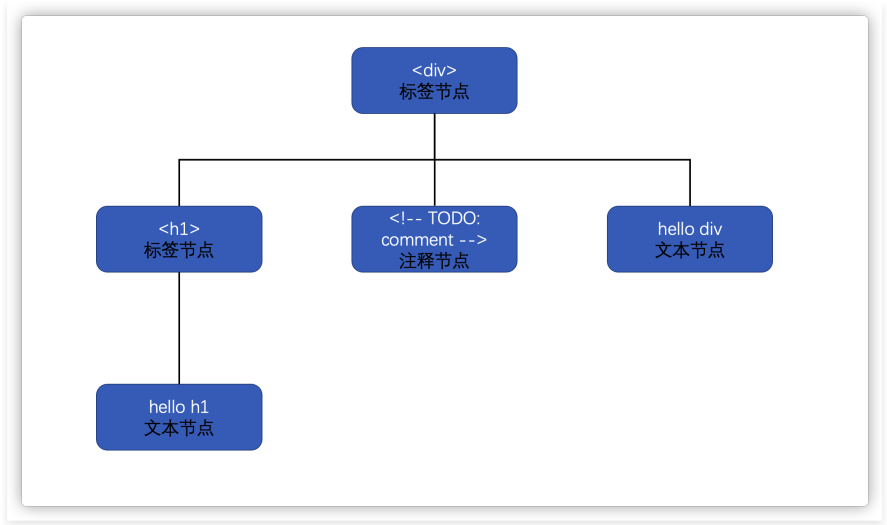
- 1. HTML DOM 节点树
- 2. 虚拟 DOM 树

我们来看下面这段 HTML：

<> 代码块

```
1 <div>
2   <h1>hello h1</h1>
3   <!-- TODO: comment -->
4   hello div
5 </div>
```

当浏览器看到这一段 html 时，它会生成一个对应的 DOM 树 来进行表示：



以上我们通过 **节点 (Node)** 来描述了以上所有的元素，在 HTML 中所有的元素都是一个节点，注释、文本都属于节点的一部分。

这样的通过节点构成的一个树形结构，我们就把它叫做 **HTML DOM 节点树**

那么明确好了什么叫做节点树之后，什么是 **虚拟 DOM 树**呢？

可能有很多同学听说过 **虚拟 DOM** 的概念，**虚拟 DOM 树** 和 **虚拟 DOM** 是息息相关的。

来自 vue 官方文档

虚拟 DOM (Virtual DOM, 简称 VDOM) 是一种编程概念，意为将目标所需的 UI 通过数据结构“虚拟”地表示出来，保存在内存中，然后将真实的 DOM 与之保持同步。这个概念是由 **React** 率先开拓，随后在许多不同的框架中都有不同的实现，当然也包括 **Vue**。

虚拟 DOM 是一种理念，比如，我期望通过一个 JavaScript 对象 来描述一个 div 节点它的子节点是一个文

索引目录

02: HTML DOM

📄

?

📱

💬

<> 代码块

```
1
2 // 通过 虚拟 dom 表示
3 const vnode = {
4   type: 'div',
5   children: 'text'
6 }
```

在上面这个对象中，我们通过 `type` 来表示当前为一个 `div` 节点，通过 `children` 来表示它的子节点，通过 `text` 表示子节点是一个文本节点，内容是 `text`。

这里所设计到的 `vnode`，是一个纯 JavaScript 对象，我们通常使用它来表示一个虚拟节点（或虚拟节点树）。它里面的属性名不是固定的，比如我可以使用 `type` 表示这是一个 `div`，也可以使用 `tag` 进行表示，都是可以的。

在 `vue` 的源码中，通过使用它来表示所需要创建元素的所有信息，比如：

<> 代码块

```
1 <div>
2   <h1>hello h1</h1>
3   <!-- TODO: comment -->
4   hello div
5 </div>
```

该例子如果使用 `vnode` 进行表示：

<> 代码块

```
1 const vnode = {
2   type: 'div',
3   children: [
4     {
5       type: 'h1',
6       children: 'hello h1'
7     },
8     {
9       type: Comment,
10      children: 'TODO: comment'
11     },
12     'hello div'
13   ]
14 }
```

在运行时 `runtime`，渲染器 `renderer` 会遍历整个虚拟 `DOM` 树，并据此构建真实的 `DOM` 树，这个过程我们可以把它叫做 **挂载** `mount`。

当这个 `VNode` 对象发生变化时，那么我们会对比 **旧的** `VNode` 和 **新的** `VNode` 之间的区别，找出它们之间的区别，并应用这其中的变化到真实的 `DOM` 上。这个过程被称为**更新** `patch`

那么这样的一个 挂载 和 更新的过程，具体是怎么做的呢？

下一小节会对此进行详细的介绍~~~

01: 前言 < 上一节 下一节 > 03: 挂载与更新

 我要提出意见反馈

