慕课网首页 免费课 体系课 慕课教程 专栏 手记 企业服务 实战课

我的课程



Q

08: 什么是运行时

?

··

从所有教程的词条中查询…

首页 > 慕课教程 > Vue3源码分析与构建方案 > 08: 什么是运行时?

全部开发者教程

Ξ

06: 为什么说框架的设计过程 其实是一个不断取舍的过程?

07: .vue 中的 html 是真实的 html 吗?

08: 什么是运行时?

09: 什么是编译时?

10: 运行时 + 编译时

11: 什么是副作用

12: Vue 3 框架设计概述

13: 扩展: 所谓良好的 `TypeScript` 支持,是如何提

供的?

14: 总结

第三章: Vue 3源码结构 -搭建框架雏形

01: 前言

02: 探索源码设计: Vue3 源

Sunday • 更新于 2022-10-19

◆ 上一节 07: .vue 中的 h... 09: 什么是编译... 下一节 ▶

索引目录

08: 什么是运行时?

在 Vue 3 的源代码中存在一个 runtime-core 的文件夹,该文件夹内存放的就是 运行时 的核心代码逻

runtime-core 中对外暴露了一个函数,叫做 渲染函数 render

我们可以通过 render 代替 template 来完成 DOM 的渲染:

有些同学可能看不懂当前代码是什么意思,没有关系,这不重要,后面我们会详细去讲。

```
<> 代码块
     <head>
       <meta charset="UTF-8">
      <title>Document</title>
       <script src="https://unpkg.com/vue@3.2.36/dist/vue.global.js"></script>
 5
     </head>
 6
     <body>
 8
       <div id="app"></div>
 9
     </body>
10
11
     <script>
12
      const { render, h } = Vue
     // 生成 VNode
1.3
      const vnode = h('div', {
14
        class: 'test'
15
16
     }, 'hello render')
17
18
      // 承载的容器
19
      const container = document.querySelector('#app')
20
21
       // 渲染函数
22
       render(vnode, container)
     </script>
```

我们知道, 在 Vue 的项目中, 我们可以通过 tempalte 渲染 DOM 节点, 如下:

```
<> 代码块
1
     <template>
2
         <div class="test">hello render</div>
     </template>
```

但是对于 render 的例子而言,我们并没有使用 tempalte, 而是通过了一个名字叫做 render 的函数, 返回了一个不知道是什么的东西,为什么也可以渲染出 DOM 呢?

带着这样的问题, 我们来看:

我们知道在上面的代码中,存在一个核心函数:渲染函数 render ,那么这个 render 在这里到底做了什 么事情呢?

我们通过一段代码实例来去看下:







假设有一天你们领导跟你说:

我希望根据如下数据:

```
1 {
2    type: 'div',
3    props: {
4         class: test
5    },
6    children: 'hello render'
7  }
```

渲染出这样一个 div:

那么针对这样的一个需求你会如何进行实现呢?大家可以在这里先思考一下,尝试进行一下实现,然后我们再继续往下看...

那么接下来我们根据这个需求来实现以下代码:

```
<> 代码块
1 <script>
const VNode = {
      type: 'div',
      props: {
        class: 'test'
5
      },
6
       children: 'hello render'
8
9
      // 创建 render 渲染函数
    function render(vnode) {
10
      // 根据 type 生成 element
11
12
      const ele = document.createElement(vnode.type)
      // 把 props 中的 class 赋值给 ele 的 className
13
14
      ele.className = vnode.props.class
      // 把 children 赋值给 ele 的 innerText
16
     ele.innerText = vnode.children
     // 把 ele 作为子节点插入 body 中
17
18
      document.body.appendChild(ele)
19
    }
21
     render(VNode)
22 </script>
```

在这样的一个代码中,我们成功的通过一个 render 函数渲染出了对应的 DOM ,和前面的 render s 不例 类似,它们都是渲染了一个 vnode ,你觉得这样的代码真是 妙极了!

但是你的领导用了一段时间你的 render 之后,却说:天天这样写也太麻烦了,每次都得写一个复杂的 vnode,能不能让我直接写 **HTML 标签结构的方式** 你来进行渲染呢?你想了想之后,说:如果是这样的话,那就不是以上 **运行时** 的代码可以解决的了!

没错! 我们刚刚所编写的这样的一个"框架", 就是 运行时 的代码框架。

那么最后,我们做一个总结: 运行时可以利用 render 把 vnode 渲染成真实 dom 节点。

▶ 我要提出意见反馈



□ 标记书签

∷





 \odot

6 6 ♣

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

⋮

?

.

<u></u>