慕课网首页 免费课 慕课教程 专栏 实战课 体系课 手记 企业服务 我的课程

Q

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:扩展知识: AST 到 JavaScript AST 的转化...

#### 全部开发者教程 ∷≡

> INMIXINI \_\_\_..... 语法树

05: 框架实现: 构建 parse 方 法, 生成 context 实例

06: 框架实现: 构建有限自动 状态机解析模板, 扫描 token 生成 AST 结构

07: 框架实现: 生成 `AST`, 构建测试

08: 扩展知识: AST 到 JavaScript AST 的转化策略和

09:源码阅读:编译器第二 步:转化 AST,得到

JavaScript AST 对象

10: 框架实现: 转化 JavaScript AST,构建深度优 先的 AST 转化逻辑

11: 框架实现: 构建 transformXXX 方法,转化对

# Sunday • 更新于 2022-10-19

◆ 上一节 07: 框架实现: ...

09: 源码阅读: ... 下一节 ▶

08: 扩展知识: A 转化策略

注意事项

索引目录

## 08: 扩展知识: AST 到 JavaScript AST 的转化策略和注意事项

在生成了 AST 之后, 我们知道接下来就需要把 AST 转化为 JavaScript AST 了, 但是在转化的过程 中,有一些对应的策略和注意事项,我们需要在本小节中进行描述。

### 转化策略

我们知道从 AST 转化为 JavaScript AST 本质上是一个对象结构的变化,变化的本质是为了后面更方便 的解析对象, 生成 render 函数。

在转化的过程中, 我们需要遵循如下策略:

- 1. 深度优先
- 2. 转化函数分离
- 3. 上下文对象

#### 深度优先

我们知道 AST 而言, 它是包含层级的, 比如:

- 1. 最外层是 ROOT
- 2. children 是根节点
- 3. ...

这样的结构下,就会存在一个自上而下的层级,那么针对这样的一个层级而言,我们需要遵循 深度优 **先, 自下而上** 的一个转化方案。

因为父节点的状态往往需要根据子节点的情况才能够进行确定,比如:

<>代码块 <div>hello world</div> <div>hello {{ msg }}</div>

## 转化函数分离

在处理 AST 的时候,我们知道,针对于不同的 token , 那么会使用不同的 parseXXX 方法进行处理 , 那 么同样的,在 transform 的过程中,我们也会通过不同的 transformXXX 方法进行转化。

但是为了防止 transform 模块过于臃肿,所以我们会通过 options 的方式对 transformXXX 方法进行注 入 (类似于 render 的 option)。

所有注入的方法,会生成一个 nodeTransforms 数组, 通过 options 传入。

#### 上下文对象

上下文对象即 context , 对于上下文对象而言我们其实并不陌生。比如在 parse 时、 setup 函数中、 vuex 的 action 上,都出现过 context 对象。

对于 context 而言,我们可以把它理解为一个 全局变量 或者 单例的全局变量,它是一个多模块都可以 访问的唯一对象。

在 transform 的策略中, 因为存在 转化函数分离 这样的一个特性, 所以我们我们必须要构建出这样的 一个 context 对象 田来保存 当前的 node 节点 等数据

▶ 意见反馈

♡ 收藏教程

□ 标记书签

?

··

#### 注意事项

说完了转化策略之后, 我们来看下注意事项。

对于 transform 转化方法而言, vue 本身的实现非常复杂, 比如: 指令、  $\{\{\}\}$  ... 都会在这里处理。

但是对于我们当前而言,我们不考虑这些复杂的情况,仅查看最简单的静态数据渲染,以此来简化整体逻辑。

✓ 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

\* \* 6 A

3

 $\odot$ 

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

口 标记书签