导航

博客园
首页
新随笔
联系
订阅 XML
管理

< 2024年5月 >

日	一	二	三	四	五	六
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

公告

昵称： 前端码牛
园龄： 7年4个月
粉丝： 5
关注： 6
+加关注

搜索

找找看

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

数据类型(1)
JavaScript的.reduce()方法(1)
javascript事件(1)
bootstrap表单验证(1)
bootstrap4(1)
bootstrap(1)

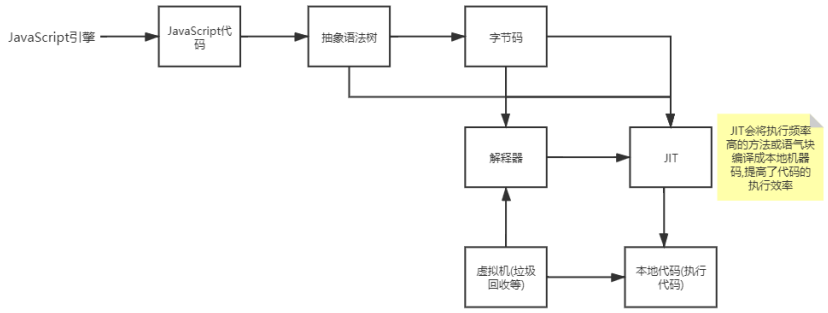
随笔分类

bootstrap(1)
javascript(32)
linux(4)
Vue(1)
版本控制(3)
计算机基础(2)
前端案例--AJAX(2)
前端开发(37)
软件开发基础知识点(17)
设计理论(1)

随笔档案

AST (abstract syntax code)抽象语法树

小前提，先来了解一下JavaScript引擎的工作原理吧！



如图所示，JavaScript引擎做的第一件事情就是把JavaScript代码编译成抽象语法树。

什么是AST抽象语法树

我们都知道,在传统的编译语言的流程中,程序的一段源代码在执行之前会经历三个步骤,统称为"编译":

分词/词法分析

这个过程会将由字符组成的字符串分解成有意义的代码块,这些代码块统称为词法单元(token).

举个例子: let a = 1, 这段程序通常会被分解成为下面这些词法单元: let 、a、 =、 1、 ； 空格是否被当成词法单元，取决于空格在这门语言中的意义。

解析/语法分析

这个过程是将词法单元流转换成一个由元素嵌套所组成的代表了程序语法结构的树,这个树被称为"抽象语法树"（abstract syntax code，AST）

代码生成

将AST转换成可执行代码的过程被称为代码生成.

抽象语法树（abstract syntax code，AST）是源代码的抽象语法结构的树状表示，树上的每个节点都表示源代码中的一种结构，之所以说是抽象的，抽象表示把js代码进行了结构化的转化，转化为一种数据结构。这种数据结构其实就是一个大的json对象，json我们都熟悉，他就像一颗枝繁叶茂的树。有树根，有树干，有树枝，有树叶，无论多小多大，都是一棵完整的树。

简单理解,就是把我们写的代码按照一定的规则转换成一种树形结构。

AST的用途

2022年10月(5)
2022年9月(7)
2022年7月(3)
2022年6月(3)
2022年5月(11)
2022年4月(15)
2022年3月(1)
2022年1月(1)
2021年11月(1)
2021年9月(3)
2021年8月(1)
2021年6月(1)
2021年5月(7)
2021年4月(1)
2019年12月(3)
更多

文章分类

ajax(1)
HTML+CSS(14)
Javascript知识点(16)
计算机英语(26)
软件使用（IDE使用）(5)
设计理论(23)
艺术设计(20)

阅读排行榜

1. 软件接口(API)(1861)
2. jQuery选择器(1823)
3. 计算机的基本概念、组成结构(894)
4. bootstrapV4表单验证(866)
5. AST (abstract syntax code)抽象语法树(831)

AST的作用不仅仅是用来在JavaScript引擎的编译上，我们在实际的开发过程中也是经常使用的，比如我们常用的babel插件将 ES6转化成ES5、使用 UglifyJS来压缩代码 、css预处理器、开发WebPack插件、Vue-cli前端自动化工具等等，这些底层原理都是基于AST来实现的，AST能力十分强大， 能够帮助开发者理解JavaScript这门语言的精髓。

AST的结构

先来看一组简单的AST树状结构：

```
const team = '大转转FE'
```

输出如下AST树状结构：

```
{
  "type": "Program",
  "start": 0,
  "end": 18,
  "body": [
    {
      "type": "VariableDeclaration",
      "start": 0,
      "end": 18,
      "declarations": [
        {
          "type": "VariableDeclarator",
          "start": 6,
          "end": 18,
          "id": {
            "type": "Identifier",
            "start": 6,
            "end": 8,
            "name": "team"
          },
          "init": {
            "type": "Literal",
            "start": 11,
            "end": 18,
            "value": "大转转FE",
            "raw": "'大转转FE'"
          }
        }
      ],
      "kind": "const"
    }
  ],
  "sourceType": "module"
}
```

一个标准的AST结构可以理解为一个json对象，那我们就可以通过一些方法去解析和操作它

AST编译过程

AST编译流程图:

我们可以看到,AST工具会源代码经过四个阶段的转换:

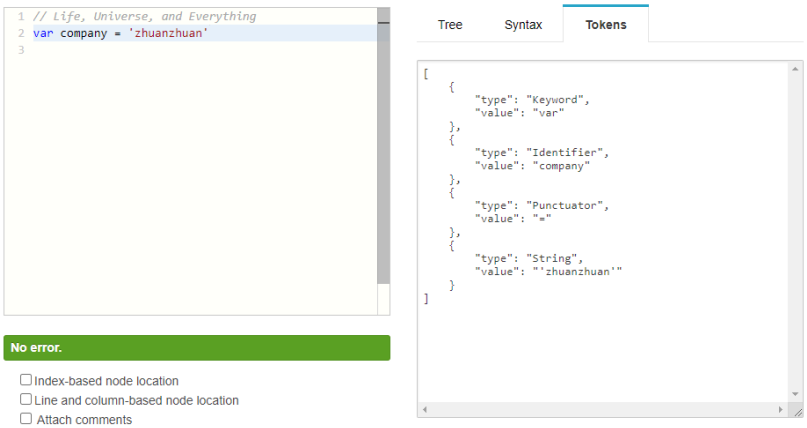
1.词法分析

```
var company = 'zhuanzhuan'
```

以上代码，在词法分析阶段，会先对整个代码进行扫描，生成tokens流，扫描过程如下:

1. 我们会通过条件判断语句判断这个字符是 字母， "/" , "数字" , 空格 , "(" , ")" , ";" 等等。
2. 如果是字母会继续往下看如果还是字母或者数字，会继续这一过程直到不是为止，这个时候发现找到的这个字符串是一个 "var"， 是一个Keyword，并且下一个字符是一个 "空格"， 就会生成{ "type" : "Keyword" , "value" : "var" }放入数组中。
3. 它继续向下找发现了一个字母 'company'(因为找到的上一个值是 "var" 这个时候如果它发现下一个字符不是字母可能直接就会报错返回)并且后面是空格，生成{ "type" : "Identifier" , "value" : "company" }放到数组中。
4. 发现了一个 "=", 生成了{ "type" : "Punctuator" , "value" : "=" }放到了数组中。
5. 发现了'zhuanzhuan',生成了{ "type" : "String" , "value" : "zhuanzhuan" }放到了数组中。

解析如下:



2.parser生成AST树

https://blog.csdn.net/P6P7qsW6ua47A2Sb/article/details/109172264

分类: javascript , 前端开发

好文要顶 关注我 收藏该文 微信分享

前端码牛
粉丝 - 5 关注 - 6

0 0

+加关注

升级成为会员

« 上一篇： VMware虚拟机中linux CentOS7上网联网

» 下一篇： javascript模块化

posted on 2022-07-27 21:35 前端码牛 阅读(831) 评论(0) 编辑 收藏 举报

会员力量，点亮园子希望 刷新页面 返回顶部

登录后才能查看或发表评论，立即 登录 或者 逛逛 博客园首页

【推荐】凡泰极客：跨越技术“鸿”沟，小程序一键生成鸿蒙App
【推荐】三生石上：ASP.NET Core中运行WebForms业务代码
【推荐】会员力量，点亮园子希望，期待您升级成为园子会员
【推荐】阿里云云市场联合博客园推出开发者商店，欢迎关注

编辑推荐：
· 架构与思维：4大主流分布式算法介绍
· 13年过去了，Spring 官方竟然真的支持 Bean 的异步初始化了！
· 日常 Bug 排查 - 偶发性读数据不一致
· 一次 nginx 文件打开数的问题排查处理
· 记一次 asp.net 8 服务器爆满的解决过程

阅读排行：
· 自己动手2小时学会配置游戏辅助
· C#.Net筑基-类型系统①基础
· 一款开源的.NET程序集反编译、编辑和调试神器
· .NET Aspire 正式发布：简化 .NET 云原生开发
· MLOps 学习之旅「GitHub 热点速览」