

课程介绍

2020年02月16日, 星期日 20:15

- 1.前端三大基础语言：HTML、CSS、JavaScript基础
- 2.全栈：前端+后端
- 3.ECMAScript(ES)是JavaScript的语法标准
- 4.前端JS构成：ECMAScript、DOM（文档对象模型）、BOM（alert、prompt、定时器）
- 5.ES5（2009年12月发布）当前网络上大部分用的是ES5
ES6（2015年06月发布）增加了许多新特性，并解决了很多ES5中的缺陷，逐渐流行开来
- 6.Chrome是对ES6支持力度最大的，反应速度最快的浏览器
- 7.VSCode插件：

Auto Rename Tag : 自动重命名配对的HTML / XML标签

HTML CSS support : HTML/CSS 自动补齐

CSS Peek：追踪样式

JS-CSS-HTML Formatter：代码格式化

AutoFileName：文件路径自动补全插件

Open-In-Browser：快捷打开浏览器

JS发展历程

2020年02月17日, 星期一 08:24

1.思考两个问题:

JavaScript 只能够在浏览器中执行?

答: 不是的, 也可以在node.js中运行, 此时称为服务端JS

JavaScript 在浏览器中是如何运行的?

答: 由JS引擎将JS语言解释为计算机可以识别的语言, 然后执行。

● JavaScript 只能够在浏览器中执行?

- 不是, JavaScript 除了在浏览器中运行, 还可以在其他的运行环境中运行, 如 node.js 环境。目前 JavaScript 的运行环境有浏览器和 node.js 环境两种。

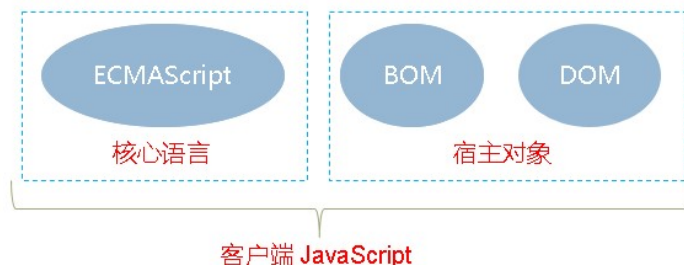
● JavaScript 在浏览器中是如何运行的?

- 浏览器下载 JavaScript 脚本文件后, 由浏览器 JavaScript 引擎解释执行。

2.ES是核心语言, BOM和DOM是宿主对象

● JavaScript 和 DOM 并不是不可分割的, 它们的语言标准相互独立。

- DOM 对 JavaScript 来说, 是**宿主对象**, 是语言中**可更换**的部分
- ECMAScript 对 JavaScript 来说, 是**核心语言**, 是**不可被替代**的功能



3.客户端JavaScript:

运行环境: 浏览器 (提供BOM、DOM)



服务端JavaScript:

新的宿主环境: node.js (提供fs文件等)



4.浏览器——客户端JS运行的宿主环境

要想执行JS语言要转换成计算机可以识别的语言, 这个工作由浏览器中的JS引擎完成

浏览器	JavaScript 实现方式(JS引擎)
Chrome	v8 
FireFox	SpiderMonkey
Safari	JavaScriptCore
Opera	Carakan
IE -> Edge	JScript (IE3.0-IE8.0) / Chakra (IE9+)

JS引擎对JS进行解释执行。

ES6到目前为止并不是所有浏览器都对其进行大力支持，目前支持力度最大的是谷歌浏览器，V8引擎对其进行解释执行。

5.ES是JS的语法标准。

ES版本：

- 每
年
更
新
一
个
版
本

 - **ES5** (2009年12月发布) 当前网络上大部分用的是 ES5
 - **ES6** (2015年06月发布) 增加了许多新特性，并解决了很多 ES5 中的缺陷，逐渐流行开来
 - **ES7** (2016年06月发布) 完善 ES6 规范
 - **ES8** (2017年06月发布) 增加新的功能，如并发、原子操作等
 - **ES9** (2018年06月发布) 增加了异步迭代，**RegExp** 等相关功能
 - **ES10** (2019年06月发布) 增加了一些新的方法

每年更新一个版本，并且在6月份发布

浏览器对这些并没有全部支持，因此有兼容性问题

ES6是ES的第六个版本，也叫ES2015。

JS语言特点

2020年02月17日, 星期一 08:47

JS语言特点:

- 直译式脚本语言
在宿主（浏览器、Node）中解释执行（与解释执行相对的是编译执行，如：C语言）
非编译语言，不是在执行前提前编译可执行文件或字节码
解释执行：解释一段执行一段
编译执行：由编译程序将全部目标代码一次性编译成目标程序，再由机器运行
解释执行速度更慢，而编译执行速度更快
- 弱类型、动态类型语言
TypeScript可以解决数据类型问题，JS语言的一个变形，或者说一个超级
写程序时不用给变量指定特定的数据类型（动态类型）
可以动态的更改变量的类型（弱类型）
- JS语言特点
ES5没有块作用域
基于原型链的继承方式（在Java等语言中的继承是基于类的）