

ECMAScript

sanfusu <sanfusu@foxmail.com>

2018 年 7 月 22 日

摘要

思前想后，如今非游戏类的桌面应用需要那么高的性能要求吗？一般而言，桌面类应用的目的在于展示信息，处理数据什么的交给服务器不好吗？现在的时代是云时代，单机处理时代早已过时，以至于 Google 出了一个 ChromeOS。如果按照这种思维，过往的 UI 框架都太过于繁复，为了能够实现富表现力的应用，往往需要花费大量的精力在 UI 表现上，从而忽视内容本身。做为 WEB 时代用来传输信息的 HTML 所衍生出来的一系列应用几乎带有解决这些问题的天然属性。一个桌面应用的组成部分可以划分为：内容、形式、交互，功能。照理想状态，这四个部分应该互相独立，可以互相套用。一个具体内容可以拥有数个表现形式，也可以提供多种交互方案。同一个功能却有可能做出两个完全不一样的应用出来。各组件之间有着互相关联的协议。以此来看 Linux 各发行版本中的 X-Windows 的 C/S 架构渲染效率虽不及 Windows。但 Linux 桌面所表现出来的富表现力远非 Windows 系统所能比的。

以上到此为止，吧啦吧啦的说了一堆实际上只是为了自己选择一条好用的桌面应用技术路线。当然本人菜鸟一枚，自然要考虑简单易用这些乱七八糟的东西，免得走上一条费力不讨好的路。下文是 ECMAScript 的相关内容。

†1 文法

非终止符号（也称为产生式）的定义有后面跟随一个或多个冒号的非终止符来引入定义。冒号的数量决定了产生式所属的语法分类。

词法和正则表达式语法使用 “::” 两个冒号区分；数值字符串语法的产生式则使用三个冒号 “:::” 区分；

和其他文法记号不一样，ECMAScript 中使用下标来表示特定含义。比如一般文法中使用中括号来表示可选，但 ECMAScript 中使用下标后缀 “*opt*” 来表示可选符号。

参数化文法记号 通过使用下标 “[*parameters*]” 来表示参数化的文法记号。其中的 parameter 既可以是单个参数名，也可以是有逗号隔开的参数名列表。

†2 数字类型

数字类型一共有 18437736874454810627 （也就是 $2^{64} - 2^{53} + 3$ ）个值，表示 IEEE 二进制浮点算术标准中 64-bit 双精度 IEEE 754-2008 格式。NaN 的表示和 IEEE 754 标准不太一样。三个特殊值：NaN、正无穷、负无穷。其余 $2^{64} - 2^{53}$ 个值被称为有穷数字。