#### ES2019提案预览

##### 之面向对象

摘要

JavaScript语言特性——Static class features

@link <https://github.com/tc39/proposal-static-class-features/>

JavaScript语言特性——Class field declarations for JavaScript

@link <https://github.com/tc39/proposal-class-fields>

JavaScript语言特性——Private methods and getter/setters for JavaScript classes

@link <https://github.com/tc39/proposal-private-methods>

JavaScript语言特性——ECMAScript class property access expressions

@link <https://github.com/tc39/proposal-class-access-expressions>

正文

JavaScript语言面向对象

以前听人这样说，JavaScript除了名字中包含Java，其余和Java没有关系。那么二者有没有关系呢？有的。JavaScript最早设计这门语言的作用是作为衔接后端Java组件与前端Web组件的“胶水语言”。因为是衔接Java组件的脚本语言，所以命名为（Java+Script=）JavaScript。因为前后端传值时会由于诸多因素（如弱网络环境等）导致一些错误（如空值、空指针），所以设计为弱类型。因为时代背景与开发时间的限制（第一版JavaScript脚本引擎是3天??写出来的），所以采用原型链（prototype chain），未原生支持面向对象特性。综上所述，JavaScript语言中的面向对象特性本身就是一层PolyFill。

Static class features

以Hash tag（#）标记私有属性、私有方法，示例如下：

class ColorFinder {

static #red = "#ff0000";

static #green = "#00ff00";

static #blue = "#0000ff";

static colorName(name) {

switch (name) {

case "red": return ColorFinder.#red;

case "blue": return ColorFinder.#blue;

case "green": return ColorFinder.#green;

default: throw new RangeError("unknown color");

}

}

// Somehow use colorName

}

Class field declarations for JavaScript

与Static class features类似

Private methods and getter/setters for JavaScript classes

支持getter、setter方法，示例如下：

class Counter extends HTMLElement {

xValue = 0;

get x() { return this.xValue; }

set x(value) {

this.xValue = value;

window.requestAnimationFrame(this.render.bind(this));

}

}

ECMAScript class property access expressions

这个提案建议支持以下特性：

在静态方法中访问对象中的非静态属性

一个神奇的特性

□