

JavaScript 的未來與現在

JavaScript 的未來與現在 10



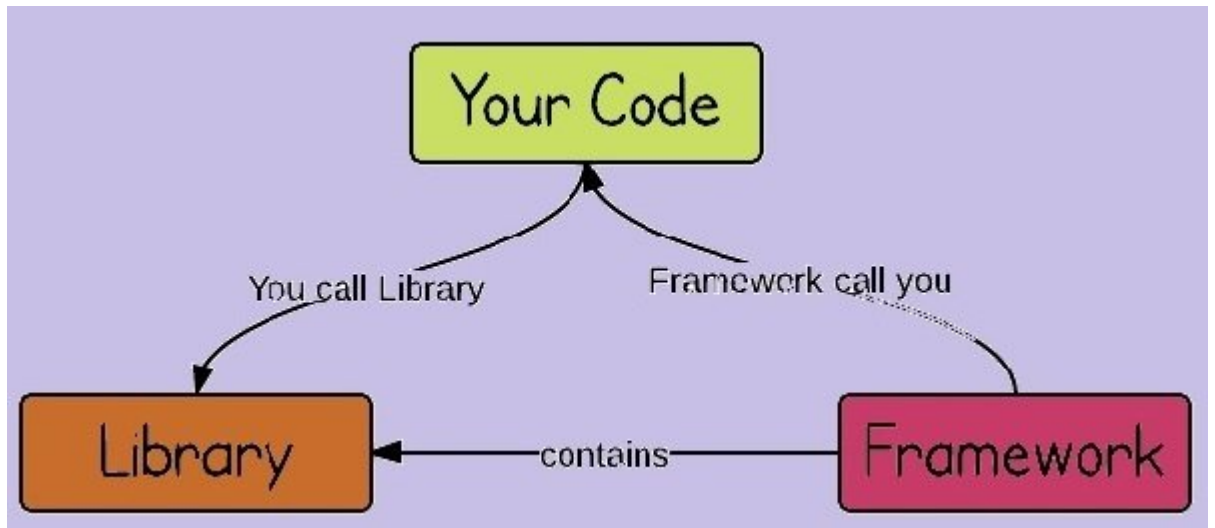
JavaScript 的未來與現在，這是一個關於 JavaScript 的未來與現在的討論。在 2017 年，JavaScript 已經成為最流行的編程語言之一。它的未來充滿了希望，因為它正在不斷地演進和改進。現在，JavaScript 已經成為開發者們最喜愛的語言之一。它的未來充滿了希望，因為它正在不斷地演進和改進。

198 年，JavaScript 被發明出來。它是一種基於對象的編程語言，具有動態類型和弱類型。它的未來充滿了希望，因為它正在不斷地演進和改進。現在，JavaScript 已經成為開發者們最喜愛的語言之一。它的未來充滿了希望，因為它正在不斷地演進和改進。

JavaScript 的未來與現在，這是一個關於 JavaScript 的未來與現在的討論。在 2017 年，JavaScript 已經成為最流行的編程語言之一。它的未來充滿了希望，因為它正在不斷地演進和改進。現在，JavaScript 已經成為開發者們最喜愛的語言之一。它的未來充滿了希望，因為它正在不斷地演進和改進。

• 00000 0000 000 00000 00 00000 0000 00 0000000000

□□□□ □□□ □□□□□□□ □ □□□□□□□□ □□□ □□□□□ □□

[illegible]

11 12 13 14 15 16 17 18 19 20



□□□□ □□□ □□□□□ □□□□□□□□□□ □□□□ □□

2017 IEEE

10/10/2020

. □□□□ □□□□ □□□□□□□□ □□ □□ □□□ □□□□ □□□□

jQuery

0 0000000000000000 000 00 0000 000 000000 0000 00 00000 0000000000 JQUERY 00000000 0000 0000 00
 000 00 2006 000 00 000 000000 000000 0000000000 000 .000000 00000000 0000 000 000000000000 0000000000

!!!!!!!

000000 00 **React** 0000000000 0000 000 00 000000 00000000 00 00 0000 00 000 000000 00000000 2013 0000 00
 00 0000 000 000000 **front-end** 00000000000000 000000000000 00 0000 00 **React** 000000 00 00 0000 00 00 .000
React.js 00 00 000000 0000 00 **React** .00000000 00000000 00 00 0000 000000 00 0000 000000 00 0000
 000000 00 00 00 0000000000 00 000000 000 000 00 0000000000 0000 .000 000 0000.00000000 00 00000000
 00000000 00 **SPA** 000000 0000 **React** .0000 000000 00 0000000000 00000000 00000000 000000 0 0000 00 0000
 000000 0000 00 0000 00 000 0000000000 00 0000 000 0000000000 00 000000 **SPA** .000000 00000000 00000000
 00000000 00 0000 00 0000000000 000 00 .0000 00000000 0000000000 000 00 0000 000 00 00000000 000000 0
 0000000000 00 0000000000 0000 .000000 0000000000 00000000 0000 0000 00 0 0000 000000 000000 000000 0000
 000 00 00000000 00 .00000000 000000 0000 00 00000000 0 0000 0000 00000000 00000000 00 000000 00000000
 00000000 **JSX** 00 **React** .000000 000000 00000000 0000 00 00000000 00000000 00000000 0000 00000000 00000000
 0000 00 **React** .00000000 000000 00 **HTML** 0 00000000 0000 00 0000 **XML** 00000000 000000 00 00000000
 00 000 00000 00000000000000 00 00000000 .000 0000000000 00000 0000 0000 00 00000 .0000 0000000000
 .000 00000000 0000 **JSX** 00 00 00000000 000000 00000 0000 00 0000 .0000 00000 0000 **JSX** 00 0000 000 00000000
 00000 0000000000 000 00 000000 000000 000000 000000 00 **front-end** 000000000000 00 000000 00 0000000000 0000
 .0000



Ember.js 是一個基於 MVC 模式的 JavaScript 框架，它可以帮助你快速构建 web 应用。

[Ember.js 2.0.0 更新日志](#) [Ember.js 2.0.0 安装指南](#) [Ember.js 2.0.0 迁移指南](#)

Ember.js 2.0.0 更新日志

Ember.js 2.0.0 版本带来了许多新的特性和改进。其中最引人注目的是 Glimmer.js 的引入。Glimmer.js 是一个新的模板引擎，它比 Ember.js 中的旧模板引擎更快、更轻量级。此外，Ember.js 2.0.0 还引入了新的路由系统、新的组件系统以及新的测试工具。这些改进使得 Ember.js 2.0.0 成为一个更加现代化、更加易于使用的框架。如果你正在使用 Ember.js，那么升级到 2.0.0 版本将是一个不错的选择。

Ember.js 2.0.0 版本带来了许多新的特性和改进。其中最引人注目的是 **Vue** 的引入。Vue 是一个新的模板引擎，它比 Ember.js 中的旧模板引擎更快、更轻量级。此外，Ember.js 2.0.0 还引入了新的路由系统、新的组件系统以及新的测试工具。这些改进使得 Ember.js 2.0.0 成为一个更加现代化、更加易于使用的框架。如果你正在使用 Ember.js，那么升级到 2.0.0 版本将是一个不错的选择。

Ember.js 2.0.0 更新日志

Ember.js 2.0.0 更新日志

Ember.js 2.0.0 版本带来了许多新的特性和改进。其中最引人注目的是 Bootstrap 的引入。Bootstrap 是一个流行的前端框架，它可以帮助你快速构建 web 应用。此外，Ember.js 2.0.0 还引入了新的路由系统、新的组件系统以及新的测试工具。这些改进使得 Ember.js 2.0.0 成为一个更加现代化、更加易于使用的框架。如果你正在使用 Ember.js，那么升级到 2.0.0 版本将是一个不错的选择。

Ember.js 2.0.0 更新日志

Ember.js 2.0.0 版本带来了许多新的特性和改进。其中最引人注目的是 AngularJS 1.x 的引入。AngularJS 1.x 是一个流行的前端框架，它可以帮助你快速构建 web 应用。此外，Ember.js 2.0.0 还引入了新的路由系统、新的组件系统以及新的测试工具。这些改进使得 Ember.js 2.0.0 成为一个更加现代化、更加易于使用的框架。如果你正在使用 Ember.js，那么升级到 2.0.0 版本将是一个不错的选择。

1000000000000000000



FrontEnd

而 Ember.js 則在 2011 年 10 月 1 日推出。Ember.js 是一個基於 Ember 的 JavaScript 框架，它提供了一個完整的 MVC 架構，並包含了一個 CLI 工具。Ember.js 的 CLI 工具可以幫助開發者快速搭建一個 Ember.js 應用，並提供了一些預定義的模板和組件。Ember.js 的 CLI 工具還可以幫助開發者管理應用中的依賴關係，並提供了一些其他的開發工具。Ember.js 的 CLI 工具是一個非常強大的工具，它為開發者提供了一個完整的開發環境，並使得開發者可以更加輕鬆地開發 Ember.js 應用。

Router

```
.router = Ember.Router.extend({
  // ...
});
```

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

Ruby on Rails 的测试框架是 RSpec，而 Ember.js 的测试框架是 Ember.Test。Ember.Test 是 Ember 框架的一部分，它提供了对 Ember.js 的测试支持。Ember.Test 的测试用例通常使用 RSpec 的语法来编写，但也可以使用其他测试框架如 Mocha 或 Jest。Ember.Test 的测试用例通常使用 RSpec 的语法来编写，但也可以使用其他测试框架如 Mocha 或 Jest。

0000 0000 000 000 0000 000000 000 00 000000 0000000 0000
 0000 00 0 000000 0000000 00 000 000 00 000 00 00 0000 00000
 000000 000000000000 0 00000000000

□□□□□ □□□□□□□ □□ :AURELIA

00000000 0000 0000 00000 .0000 00000000 00 00 000000000 0000 00 00000 front-end 0000 0000 0000000000 00 Aurelia
 000000000000 00000 00000 00 0000 0000000000 00000 000000000 00 Aurelia .00 0000000 00000000 2016 0000 00
 000000000 00 Ember.JS 0 React 0Angular 00 00000 Aurelia .00000000 000000 0000 00 00 0 00000000 000000000
 0000 00000000 00000 00000 00 00 0000000 00000000000 0000 00 0000 00000000000 0000 .0000 00000000 0 00000000 SPA
 .00000 00000 0000000000 00000

Aurelia .NET Core framework uses ES6 features like classes, modules, decorators, etc. to build modern web applications. It also supports TypeScript, which is a superset of JavaScript that adds static typing and other features. Aurelia uses the EcmaScript standard for its core functionality.

Aurelia

<https://www.shabakeh-mag.com/cover-story/11203/10-%DA%A9%D8%AA%D8%A7%D8%A8%D8%AE%D8%A7%D9%86%D9%87-%D9%88-%DA%86%D9%87%D8%A7%D8%B1%DA%86%D9%88%D8%A8-%D8%AC%D8%A7%D9%88%D8%A7-%D8%A7%D8%B3%DA%A9%D8%B1%DB%8C%D9%BE%D8%AA%DB%8C-%DA%A9%D9%87-%D8%A8%D8%A7%DB%8C%D8%AF-%D8%A7%D8%B2-%D9%88%D8%AC%D9%88%D8%AF-%D8%A2%D9%86%E2%80%8C%D9%87%D8%A7-%D8%A8%D8%A7%D8%AE%D8%A8%D8%B1-%D8%A8%D8%A7%D8%B4%DB%8C%D8%AF>