



产品

解决方案

搜索社区



备案

控制台

登录

注册

开发者社区

首页

探索云世界

问产品

动手实践

官方博客



个人

积分

发布

## React系列(一)-邂逅React开发

2022-06-26 142

版权

**简介：**我个人一直喜欢使用邂逅这个词，我希望每个人在和一个新技术接触的时候，都是一场美丽的邂逅。而React本身绝对是优雅和美丽的，所以从这里开始，我们和React来一场美丽的邂逅吧！

# 一. 认识React

## 1.1. React是什么？

React是什么呢？相信每个做前端的人对它都或多或少有一些印象。

这里我们来看一下官方对它的解释：用于构建用户界面的 JavaScript 库。



React是什么？

我们知道对于前端来说，主要的任务就是构建用于界面，而构建用于界面离不开三个技术：

- HTML：构建页面的结构
- CSS：构建页面的样式
- JavaScript：页面动态内容和交互

那么使用最原生的HTML、CSS、JavaScript可以构建完整的用户界面吗？当然可以，但是会存在很多问题

- 比如操作DOM兼容性的问题；
- 比如许多兼容性带来的冗余问题。



l72sshxfx2ceg

+关注

90  
文章

0

0

0



目录

AI  
助理

- 在过去的一份调查中显示，全球前10,000个访问最高的网站中，有65%使用了jQuery，是当时最受欢迎的JavaScript库；
  - 但是越来越多的公司开始慢慢不再使用jQuery，包括程序员使用最多的GitHub；
- 现在前端领域最为流行的是三大框架：

- Vue
- React
- Angular



三个框架

而Angular在国内并不是特别受欢迎，尤其是Angular目前的版本对TypeScript还有要求的情况下。Vue和React是国内最为流行的两个框架，而他们都是帮助我们来构建用户界面的JavaScript库。

- 关于它们的对比，我会另外再写一篇文章

## 1.2. React的起源

React是2013年，Facebook开源的JavaScript框架，那么当时为什么Facebook要推出这样一款框架呢？这个源于一个需求，所产生的bug：



l72sshxfx2ceg

+关注

90  
文章

赏 0

☆ 0

👍 0

💬

目录



AI  
助理

bug是否可以修复呢？当然可以修复，但是Facebook的工程师并不满足于此；

他们开始思考为什么会产生这样的问题；

- 在传统的开发模式中，我们过多的去操作界面的细节；（前端、iOS、Android）
  - 比如说需要掌握和使用大量DOM的API，当然我们可以通过jQuery来简化和适配一些API的使用；
- 另外关于数据（状态），往往会分散到各个地方，不方便管理和维护；

他们就去思考，是否有一种新的模式来解决上面的问题：

- 1.以组件的方式去划分一个个功能模块
- 2.组件内以jsx来描述UI的样子，以state来存储组件内的状态
- 3.当应用的状态发生改变时，通过setState来修改状态，状态发生变化时，UI会自动发生更新

## 1.3. React的特点和优势

### 1.3.1. React的特点

声明式编程：

- 声明式编程是目前整个大前端开发的模式：Vue、React、Flutter、SwiftUI；
- 它允许我们只需要维护自己的状态，当状态改变时，React可以根据最新的状态去渲染我们的UI界面；

**UI = f( state )**

The layout on the screen      Your build methods      The application state

 coderwhy

声明式编程

组件化开发：

- 组件化开发页面目前前端的流行趋势，我们会讲复杂的界面拆分成一个个小的组件；
- 如何合理的进行组件的划分和设计也是后面我会讲到的一个重点；



l72sshxfx2ceg

+关注

90  
文章

赏 0

☆ 0

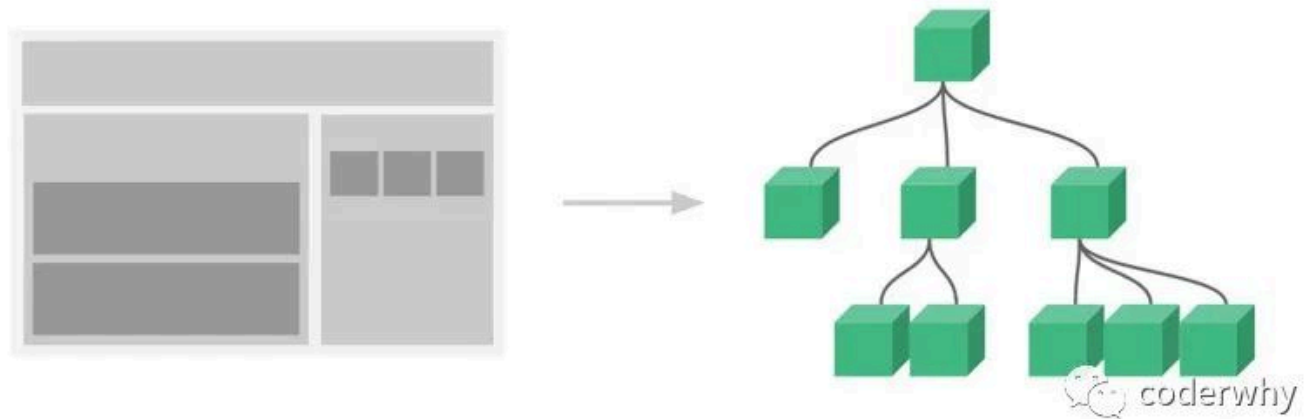
👍 0

💬

目录



AI  
助理



### 组件化开发

多平台适配：

- 2013年，React发布之初主要是开发Web页面；
- 2015年，Facebook推出了ReactNative，用于开发移动端跨平台；（虽然目前Flutter非常火爆，但是还是有很多公司在使用ReactNative）；
- 2017年，Facebook推出ReactVR，用于开发虚拟现实Web应用程序；（随着5G的普及，VR也会是一个火爆的应用场景）；



react多平台

### 1.3.2. React的优势



l72sshxfx2ceg

+关注

90  
文章

赏 0

☆ 0

👍 0

💬

目录



AI  
助理

- 包括Vue3很多新的特性，也是借鉴和学习了React
- 比如React Hooks是开创性的新功能（也是我们课程的重点）
- Vue Function Based API学习了React Hooks的思想

Flutter的很多灵感都来自React，来自官网的一段话：（SwiftUI呢）

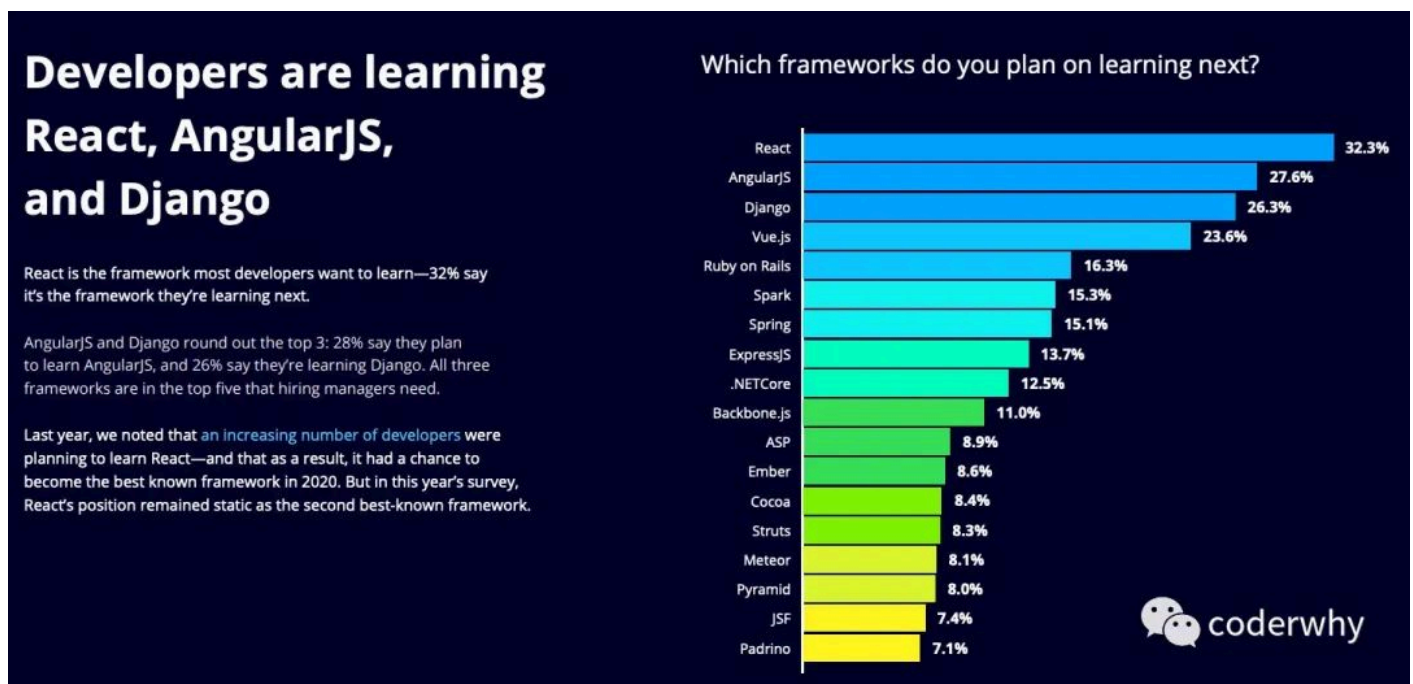
Flutter widgets are built using a modern framework that takes inspiration from [React](#).

来自Flutter官网

• 事实上Flutter中的Widget – Element – RenderObject，对应的就是JSX – 虚拟DOM – 真实DOM  
所以React可以说是前端的先驱者，它总是会引领整个前端的潮流。

## 1.4. React的现状

另外在HackerRank中，2020年有一份调用，你更想要学习的framework（框架）：



哪一个是你最想要学习的框架

国内外很多知名网站使用React开发：



I72sshxfx2ceg

+关注

90  
文章

赏 0

☆ 0

👍 0

💬

目录



AI  
助理





目前国内在大型公司使用React的较多：

### 职位诱惑：

年终奖、带薪年假、股票期权、阿里系

### 职位描述：

- 1、抽象通用功能组件，开发基础工具，提高团队效率；
- 2、参与产品设计讨论，从前端技术角度评估可行性，提供技术方案；
- 3、参与并实践 code review，提高自身与团队能力水平；
- 4、攻克技术难点并指导新人。

- 1、精通 html, css, javascript;
- 2、精通 react / vue 相关工具和技术栈，并掌握其核心原理；
- 3、熟悉 node 开发，有生产环境 node 项目应用经验；
- 4、有大中型 web app 项目的实践经验和基本架构能力。

### 加分项：

- 1、主动抽象工具，工作中有效提高工作效率；
- 2、有 jsLint, eslint 使用习惯，工作中保持良好的代码风格，有强烈的技术追求；
- 3、有任意语言的后端开发经验；
- 4、有项目管理经验；
- 5、对业务有较高关注度；
- 6、熟悉 mac 与 git。

coderwhy

## 高级前端工程师要求



I72sshxfx2ceg

+关注

90  
文章

赏 0

☆ 0

👍 0

🗨 0

目录



AI  
助理

## 2.1. 原生案例实现

为了演练React，我们可以提出一个小的需求：

- 在界面显示一个文本：Hello World
- 点击下方的一个按钮，点击后文本改变为Hello React
- 在界面显示一个文本：Hello World
- 点击下方的一个按钮，点击后文本改变为Hello React



### 案例效果

但是，我们使用React实现之前，先使用原生代码来实现，这样更加方便大家对比React和原生：

- 当然，你也可以使用jQuery和Vue来实现，对它们分别进行对比学习

原生实现代码如下：

```
1 <body>
2
3   <div class="app">
4     <h2 class="title">Hello World</h2>
5     <button class="change-btn">改变文本</button>
6   </div>
7   <script>
8     // 1.获取dom节点
9     const titleEl = document.getElementsByClassName("title")[0];
10    // 2.获取数据
11    let message = "Hello World";
12    // 3.将数据显示到titleEl中
13    titleEl.innerHTML = message;
14    // 4.改变按钮内容
```



l72sshxfx2ceg

+关注

90  
文章

赏 0

☆ 0

👍 0

💬

目录



AI  
助理

```
19    })  
20  </script>  
21 </body>
```

## 2.2. React开发依赖

开发React必须依赖三个库：

- react：包含react所必须的核心代码
- react-dom：react渲染在不同平台所需要的核心代码
- babel：将jsx转换成React代码的工具

第一次接触React会被它繁琐的依赖搞蒙，对于Vue来说，我们只是依赖一个vue.js文件即可，但是react居然要依赖三个库。

其实呢，这三个库是各司其职的，目的就是让每一个库只单纯做自己的事情：

- 在React的0.14版本之前是没有react-dom这个概念的，所有功能都包含在react里。
- 为什么要进行拆分呢？原因就是react-native。
- react包中包含了react和react-native所共同拥有的核心代码。
- react-dom针对web和native所完成的事情不同：
  - web端：react-dom会讲jsx最终渲染成真实的DOM，显示在浏览器中
  - native端：react-dom会讲jsx最终渲染成原生的控件（比如Android中的Button，iOS中的UIButton）。

babel是什么呢？

- **Babel**，又名 **Babel.js**。
- 是目前前端使用非常广泛的编辑器、转移器。
- 比如当下很多浏览器并不支持ES6的语法，但是确实ES6的语法非常的简洁和方便，我们**开发时**希望使用它。
- 那么编写源码时我们就可以使用ES6来编写，之后通过Babel工具，将ES6转成大多数浏览器都支持的ES5的语法。

React和Babel的关系：

- 默认情况下开发React其实可以不使用babel。
- 但是前提是我们自己使用 `React.createElement` 来编写源代码，它编写的代码非常的繁琐和可读性差。
- 那么我们就可以直接编写jsx（JavaScript XML）的语法，并且让babel帮助我们转换成 `React.createElement`。
- 后续还会讲到；

所以，我们有源码，有代码，还有一个能够帮我们转换的工具，这就是我们需要的。



l72sshx2ceg

+关注

90  
文章

赏 0

☆ 0

👍 0

💬

目录



AI  
助理



- react-dom依赖: <https://unpkg.com/react-dom@16/umd/react-dom.development.js>
- babel依赖: <https://unpkg.com/babel-standalone@6/babel.min.js>

- 方式二: 下载后, 添加本地依赖
- 方式三: 通过npm管理 (后续脚手架再使用)

暂时我们直接通过CDN引入, 来演练下面的示例程序:

- 这里有一个crossorigin的属性, 这个属性的目的是为了拿到跨域脚本的错误信息

```
1 <script src="https://unpkg.com/react@16/umd/react.development.js" crossorigin></script>
2 <script src="https://unpkg.com/react-dom@16/umd/react-dom.development.js" crossorigin></script>
3 <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>
```

## 2.3. Hello World

下面我们通过一个Hello World的案例来看下如何使用React开发。

需求非常简单: 通过React, 在界面上显示一个Hello World

- 注意: 这里我们编写React的script代码中, 必须添加 `type="text/babel"`, 作用是可以让babel解析jsx的语法

```
1 <div id="app"></div>
2 <script src="https://unpkg.com/react@16/umd/react.development.js"></script>
3 <script src="https://unpkg.com/react-dom@16/umd/react-dom.development.js"></script>
4 <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>
5 <script type="text/babel">
6   // 通过ReactDOM对象来渲染内容
7   ReactDOM.render(<h2>Hello World</h2>, document.getElementById("app"));
8 </script>
```

代码解析:

- 依赖不需要多讲, 开发React代码必须添加三个依赖;
- ReactDOM.render函数:
  - 这里我们已经提前定义一个id为app的div
  - 这里我们传入了一个h2元素, 后面我们就会使用React组件
  - 参数一: 传递要渲染的内容, 这个内容可以是HTML元素, 也可以是React的组件
  - 参数二: 将渲染的内容, 挂载到哪一个HTML元素上

显示效果:



I72sshxfx2ceg

+关注

90  
文章

赏 0

☆ 0

👍 0

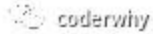
🗨 0

目录



AI  
助理

# Hello World



Hello World

但是目前我们渲染的内容是定义死的，能否将其抽取到一个变量中呢？

- 当然可以，我们可以通过{}语法来引入外部的变量或者表达式

```
1 // 将数据定义到变量中
2 let message = "Hello World";
3 // 通过ReactDOM对象来渲染内容
4 ReactDOM.render(<h2>{message}</h2>, document.getElementById("app"));
```

## 2.4. Hello React

按照我们最初的案例，我们已经实现了Hello World，但是我们希望点击一个按钮后，修改为Hello React

### 2.4.1. 错误的方式

下面的代码是我们正常的执行逻辑，但是会报错：

- 原因是默认情况下 `ReactDOM.render` 会覆盖挂载到的app原生中的所有内容；
- 所以在执行完 `ReactDOM.render` 之后，就不存在button原生了；

```
1 <body>
2
3   <div id="app">
4     <button class="change-btn">改变文本</button>
5   </div>
6   <script src="https://unpkg.com/react@16/umd/react.development.js"></script>
7   <script src="https://unpkg.com/react-dom@16/umd/react-dom.development.js"></script>
8   <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>
9   <script type="text/babel">
10     // 将数据定义到变量中
11     let message = "Hello World":
```



I72sshx2ceg

+关注

90  
文章

赏 0

☆ 0

👍 0

🗨 0

目录



AI  
助理

```
16     btnEl.addEventListener("click", (e) => {
17         console.log(e);
18     })
19 </script>
20 </body>
```

### 2.4.2. 正确的方式

虽然可以实现，但是整个代码的流畅过于繁琐

```
1 <body>
2
3 <div id="app">
4 </div>
5 <script src="https://unpkg.com/react@16/umd/react.development.js"></script>
6 <script src="https://unpkg.com/react-dom@16/umd/react-dom.development.js"></script>
7 <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>
8 <script type="text/babel">
9     // 将数据定义到变量中
10    let message = "Hello World";
11    // 通过ReactDOM对象来渲染内容
12    render();
13    // 定义一个执行的函数
14    function btnClick() {
15        message = "Hello React";
16        render();
17    }
18    function render() {
19        ReactDOM.render((
20            <div>
21                <h2>{message}</h2>
22                <button onClick={btnClick}>改变文本</button>
23            </div>
24        ), document.getElementById("app"));
25    }
26 </script>
27 </body>
```



I72sshxfx2ceg

+关注

90  
文章

赏 0

☆ 0

👍 0

💬

目录



AI  
助理

整个逻辑其实可以看做一个整体，那么我们就可以将其封装成一个组件：

- 我们说过 `ReactDOM.render` 第一参数是一个HTML原生或者一个组件；
- 所以我们可以先将之前的业务逻辑封装到一个组件中，然后传入到 `ReactDOM.render` 函数中的第一个参数；

在React中，如何封装一个组件呢？

- 这里我们暂时使用类的方式封装组件：
  - `render`当中返回的`jsx`内容，就是之后React会帮助我们渲染的内容
  - 1.定义一个类，继承自`React.Component`
  - 2.实现当前组件的`render`函数

具体的代码如下：

```
1 class App extends React.Component {  
2   render() {  
3     return (<h2>Hello World</h2>)  
4   }  
5 }  
6 ReactDOM.render(<App/>, document.getElementById("app"));
```

如果我们的Hello World是依赖变量的，并且会根据按钮的点击而改变呢？这里涉及到几个核心点

### 1.数据在哪里定义

- 在组件中的数据，我们可以分成两类：
  - 参与界面更新的数据：当数据变量时，需要更新组件渲染的内容
  - 不参与界面更新的数据：当数据变量时，不需要更新将组建渲染的内容
- 参与界面更新的数据我们也可以称之为是参与数据流，这个数据是定义在当前对象的`state`中
  - 我们可以通过在构造函数中 `this.state = {定义的数据}`
- 当我们的数据发生变化时，我们可以调用 `this.setState` 来更新数据，并且通知React进行`update`操作
  - 在进行`update`操作时，就会重新调用`render`函数，并且使用最新的数据，来渲染界面

### 2.事件绑定中的this

- 在类中直接定义一个函数，并且将这个函数绑定到html原生的`onClick`事件上，当前这个函数的`this`指向的是谁呢？
- 默认情况下是`undefined`
  - 很奇怪，居然是`undefined`；
  - 因为在正常的DOM操作中，监听点击，监听函数中的`this`其实是节点对象（比如说是`button`对象）；
  - 这次因为React并不是直接渲染成真实的DOM，我们所编写的`button`只是一个语法糖，它的本质



I72sshxfx2ceg

+关注

90  
文章

赏 0

☆ 0

👍 0

🗨 0

目录



AI  
助理

- 我们就需要在传入函数时，给这个函数直接绑定this
- 类似于下面的写法：`<button onClick={this.changeText.bind(this)}>改变文本`  
`</button>`

我们一起来看一下代码是如何实现的：

```
1 class App extends React.Component {
2   constructor() {
3     super();
4     this.state = {
5       message: "Hello World"
6     };
7   }
8   render() {
9     return (
10      <div>
11        <h2>{this.state.message}</h2>
12        <button onClick={this.changeText.bind(this)}>改变文本</button>
13      </div>
14    )
15  }
16  changeText() {
17    this.setState({
18      message: "Hello React"
19    })
20  }
21 }
22 ReactDOM.render(<App/>, document.getElementById("app"));
```

文章标签： [JavaScript](#) [前端开发](#) [iOS开发](#) [Android开发](#) [vr&ar](#) [CDN](#) [5G](#) [程序员](#) [移动开发](#)

[API](#)

关键词： [React Native开发](#) [react.js开发](#) [React开发](#)

## 相关文章

### 从零开始学习React Native开发

React Native是一种基于React框架的移动端开发框架，使用它可以快速地构建出高性能、原生的移动应用。本文将从零开



I72sshx2ceg

+关注

90  
文章

赏 0

☆ 0

👍 0

💬

目录



AI  
助理

虚拟DOM是React等前端框架的关键技术，它以轻量级JavaScript对象树形式抽象表示实际DOM。当状态改变，React不...

龙大吉 43 阅读

## 开发语言漫谈-React

React同Vue一样，都是基于Javascript上的框架，不是独立语言

大道不孤，众行致远 23 阅读

## 【Flutter前端技术开发专栏】Flutter与React Native的对比与选择

【4月更文挑战第30天】对比 Flutter（Dart，强类型，Google支持，快速热重载，高性能渲染）与 R...

梦回故国楼台梦 92 阅读



## 杨校老师项目之基于SpringBoot+React框架开发的医院挂号管理系统

杨校老师项目之基于SpringBoot+React框架开发的医院挂号管理系统

杨校 16 阅读

## React适合开发什么类型的应用？

【5月更文挑战第27天】React适合开发什么类型的应用？

sunrr 25 阅读

## 使用React Native开发跨平台移动应用的技术详解

【5月更文挑战第22天】本文详述了使用React Native开发跨平台移动应用的技术，该框架由Facebook推出，基于JavaScr...

1941623231718325 42 阅读

## 基于 React + Nest 全栈开发的后台系统

这篇文章介绍了一个基于React+Nest全栈开发的后台系统XmwAdmin。项目包括前端和后端技术栈，...

白雾茫茫、 26 阅读



## React这些新特性在开发效率上有哪些改进

【4月更文挑战第18天】React 18 提升开发效率，引入新Root API `createRoot` 优化挂载，支持渐进式升级，减少迁移成...

小周sir 80 阅读



l72sshxfx2ceg

+关注

90  
文章



0



0



0



目录



AI  
助理



热门文章

最新文章

- 1

实战：使用 React 实现渐进式加载图片

🔥 4
- 2

手把手教你创建一个简单的React/Vue.js组件

🔥 1
- 3

元素渲染 (精读React官方文档—03)

🔥 1
- 4

React旧版本与React16中生命周期的区别

🔥 2
- 5

React学习笔记(四) 条件渲染与列表渲染

🔥 1

展开更多

相关课程

- React 入门与实战
- React 入门教程开发文档
- Vue.js完全自学手册图文教程
- Vue.js 入门与实战

更多

相关电子书

- 利用编译将 Vue 组件转成 React 组件
- React Native 全量化实践
- React在大型后台管理项目中的工程实践

更多



I72sshxfx2ceg

+关注

90  
文章

👍 0

★ 0

👍 0

💬

目录



AI  
助理

为什么选择 产品和定价 解决方案 文档与社区 权益中心 支持与服务 关注阿里云



l72sshxfx2ceg

+关注

90  
文章

赏 0

☆ 0

👍 0

💬

目录

阿里云公众号或



AI  
助理

- 全球基础设施

产品动态

技术领先

稳定可靠

安全合规

分析师报告
- 产品定价

价格计算器

云上成本管理
- 天池大赛


培训与认证
- 企业扶持计划


推荐返现计划
- 迁云服务

官网公告

健康看板

信任中心
- 注云资讯，随时随地

运维管控云服务
- 



联系我们：  
4008013260

法律声明

Cookies政策

廉正举报

安全举报

联系我们

加入我们

阿里巴巴集团 淘宝网 天猫 全球速卖通 阿里巴巴国际交易市场 1688 阿里妈妈 飞猪  
阿里云计算 AliOS 万网 高德 UC 友盟 优酷 钉钉 支付宝 达摩院 淘宝海外 阿里云盘  
饿了么

© 2009-2024 Aliyun.com 版权所有 增值电信业务经营许可证：浙B2-20080101 域名注册服务机构许可：浙D3-20210002 京D3-20220015



浙公网安备 33010602009975号

浙B2-20080101-4



I72sshxfx2ceg

+关注

90 文章

赏

0

☆

0

👍

0

💬

目录



AI 助理