

Next.js 全解（上）

Node.js Web 系列课程

版权声明

本内容版权属杭州饥人谷教育科技有限公司（简称饥人谷）所有。

任何媒体、网站或个人未经本网协议授权不得转载、链接、转贴，或以其他方式复制、发布和发表。

已获得饥人谷授权的媒体、网站或个人在使用时须注明「资料来源：饥人谷」。

对于违反者，饥人谷将依法追究 responsibility。

联系方式

如果你想要购买本课程

请微信联系 [xiedaimala02](#) 或 [xiedaimala03](#)

如果你发现有人盗用本课程

请微信联系 [xiedaimala02](#) 或 [xiedaimala03](#)

Next.js 的背景

- 开发团队是 zeit

- ✓ zeit 团队水平如何？看看[这个](#)。后改名为 Vercel
- ✓ 简言之，一个高中开始编程的、会做平面设计的复旦大学计算机专业毕业生，在微软工作过一年后，加入了 zeit 团队
- ✓ 「几乎每一个同事都有非常强大的背景」
- ✓ Next.js 核心团队四个人平均年龄 20 岁
- ✓ 按 star 数，zeit 是 GitHub 组织的 Top 20
- ✓ 全员远程工作

Next.js 的定位

- Node.js 全栈框架

- ✓ CSS-in-JS
- ✓ 页面预渲染 + SSR（服务端渲染）
- ✓ 前后端同构（代码同时运行在两端）
- ✓ Node.js 10.13 以上
- ✓ React yes!
- ✓ TypeScript yes!

- 弱项

- ✓ 完全没有提供数据库相关功能，可自行搭配 Sequelize 或 TypeORM
- ✓ 完全没有提供测试相关功能，可自行搭配 Jest 或 Cypress
- ✓ 有一个叫做 [Blitz.js](#) 的框架在这些方向上努力

谁在用 Next.js

- 国内

- ✓ 腾讯网手机版 <https://xw.qq.com>
- ✓ [Teambition](#)

- 国外

- ✓ Netflix Jobs、Twitch 手机版、Hulu、TikTok 抖音海外版、漫威官网、Material UI 官网

- 总结

- ✓ 国外用得比较多，国内用得少
- ✓ 那么国内学这个好找工作么？

学 Next.js 好找工作么

• 我的思考

- ✓ 国内一般招聘偏后端的前端，看看纯 Node 后端市场
- ✓ Next.js 切合了 SSR、Serverless 等技术热点
- ✓ 支付宝的 Umi.js 是一个类 Next.js 框架，说明被认可
- ✓ Next.js 背后的团队是我非常欣赏的
- ✓ 我们已经教了 express 和 koa，你可以自行掌握 egg.js

• 综上

- ✓ 进可往技术热点靠
- ✓ 退可回到 egg.js

Next.js 全解

一节课搞清核心技术

使用 Next.js 做博客系统

- 官方教程

- ✓ 官网入门教程就是做博客系统（我略有删改）

- 需求

- ✓ 我们对博客系统的需求足够了解，不需再讨论

- 难易程度

- ✓ 非常简单，易于学习

创建项目

- 命令

- ✓ `npm init next-app nextjs-blog-1`
- ✓ 然后选择 Default starter app
- ✓ 使用你的编辑器打开 `nextjs-blog-1`
- ✓ 打开编辑器的终端，添加 `.gitignore`
- ✓ `yarn dev`
- ✓ 用浏览器打开控制台给出的网址

Welcome to Next.js!

Get started by editing `pages/index.js`

CRM 学习

将贯穿整个学习过程

CRM

- 改内容

- ✓ 把 Welcome to 改成 Hello

- 加页面

- ✓ 创建 pages/posts/first-post.js
- ✓ 猜猜如何访问这个页面?
- ✓ 路径为 /posts/first-post

- 总结

- ✓ 约定大于配置
- ✓ 不需要配置，只需要约定，开发者自然意会
- ✓ Rails 的理念

Link 快速导航

- ✓ 官方教程[在此](#)

• 用法

- ✓ 把 `点击跳转` 改成
- ✓ `<Link href=xxx> <a>点击跳转 </Link>`

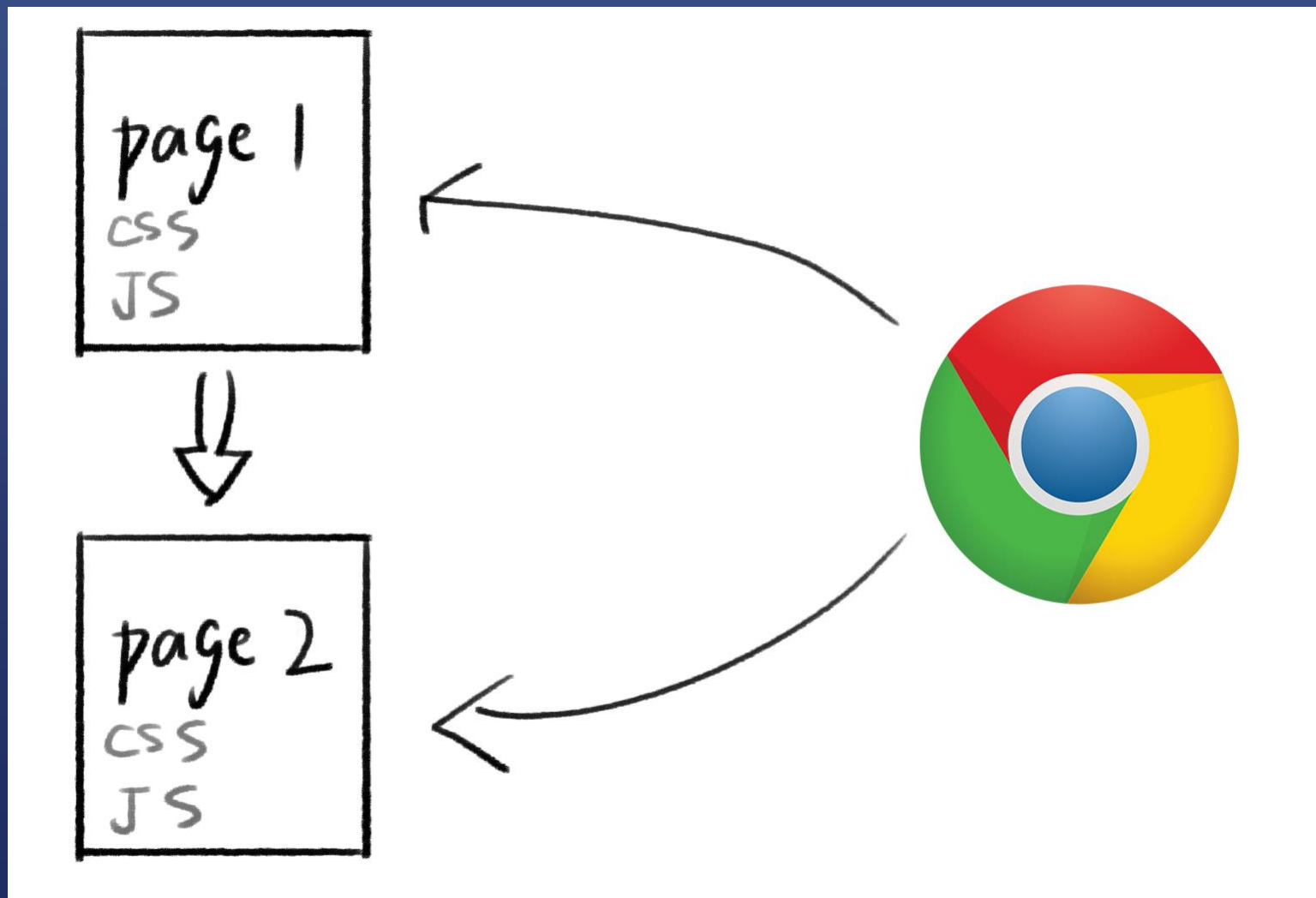
• 优点

- ✓ 页面不会刷新，用 AJAX 请求新页面内容
- ✓ 不会请求重复的 HTML、CSS、JS
- ✓ 自动在页面插入新内容、删除旧内容
- ✓ 因为省了很多请求和解析过程，所以速度极快

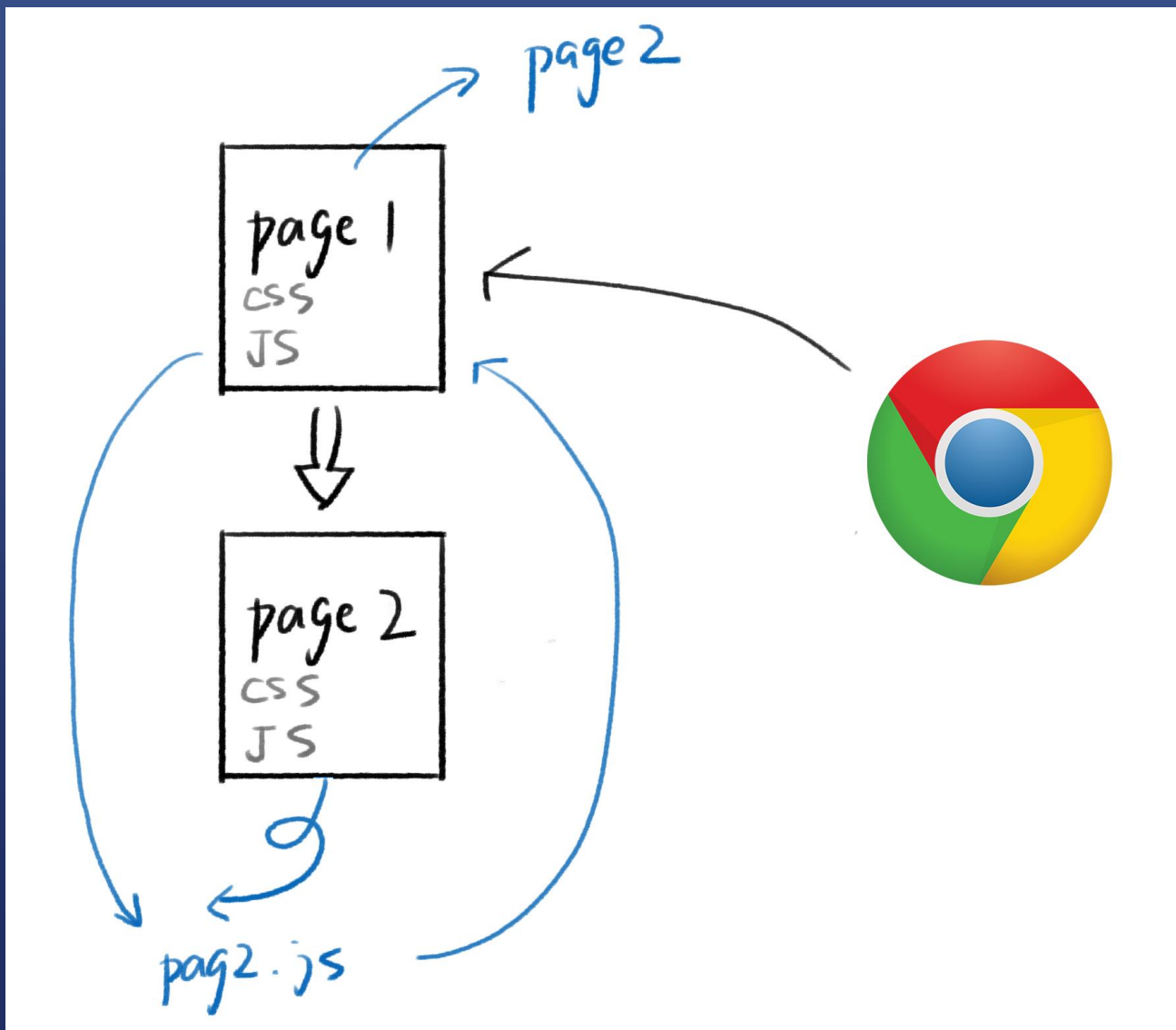
• 吐槽

- ✓ 借鉴了 Rails Turbolinks、pjax 等技术

传统导航



快速导航



同构代码

- 代码运行在两端

- ✓ 在组件里写一句 `console.log('执行了')`
- ✓ 你会发现 Node 控制台会输出这句话
- ✓ 你会发现 Chrome 控制台也会输出这句话

- 注意差异

- ✓ 不是所有的代码都会运行，有些需要用户触发
- ✓ 不是所有的 API 都能用，比如 `window` 在 Node 里报错

小结

- 创建项目

- ✓ `npm init next-app` 项目名

- 快速导航

- ✓ `<Link href=xxx>` 包住 `<a>`

- 同构代码

- ✓ 一份代码，两端运行（省了一份）

全局配置

- ✓ 官方教程[在此](#)
- 自定义 <head>
- 全局 CSS
- 局部 CSS

自定义 <hhead>

- 使用 <HHead>

- ✓ 改各页面 title
- ✓ 改各页面 meta:viewport

- 疑问

- ✓ 那怎么改全局 title 和 meta:viewport 呢

全局配置

- `pages/_app.js`

- ✓ `export default function App` 是每个页面的根组件
- ✓ 页面切换时 `App` 不会销毁，`App` 里面的组件会销毁
- ✓ 可用 `App` 保存全局状态

- 注意

- ✓ 创建 `_app.js` 之后需要重启 `yarn dev`

全局 CSS

- 放在 `_app.js` 里

- ✓ `import '../styles/global.css'`
- ✓ 因为切页面时 App 不会销毁
- ✓ 其他地方不能 `import global.css`
- ✓ 其他地方只能写局部 CSS

- 插曲

- ✓ 相对引用好烦，能改成 `import 'styles/global.css'` 吗？
- ✓ 看文档「[Absolute Imports](#)」章节

局部 CSS

- 官方支持

- ✓ 默认支持 [styled-jsx](#) 和 [CSS Modules](#)
- ✓ 一般来说，简单需求用前者，复杂需求用后者

- 我的个人偏好

- ✓ styled-jsx 不方便分离 CSS 和 JS
- ✓ CSS Modules 用起来太麻烦
- ✓ 我更喜欢 [styled-components](#)，后面有机会再改

使用 SCSS

- 安装依赖

- ✓ yarn add sass

- 直接用

- ✓ 把后缀 .css 改为 .scss 即可

静态资源

next 推荐放在 public/ 里，我不是很赞同

其他类型文件

- 图片

- ✓ 配置 file-loader
- ✓ 或者直接用 next-images

- SVG

- ✓ 一部分可以当作图片使用
- ✓ icons/svg 要当作 SVG Symbols 使用
- ✓ 目前没有插件，只能自己配置

- 更多

- ✓ 自己要 loader，然后配置 next.config.js
- ✓ 或者看有没有人封装成 next 插件

总结

- 创建项目

- ✓ npm init next-app 项目名

- 快速导航

- ✓ <Link href=xxx> 包住 <a>

- 同构代码

- ✓ 一份代码，两端运行

- 全局组件

- ✓ pages/_app.js

- 自定义 head

- ✓ 使用 <Head> 组件

- 全局 CSS

- ✓ 在 _app.js 里 import

- 局部 CSS

- ✓ <style jsx>
- ✓ <style jsx global> 不推荐
- ✓ xxx.module.css

- SCSS

- ✓ 安装 sass 依赖即可

WebStorm 技巧

- 安装 @types/xxx
 - ✓ 可以使 WebStorm 对 xxx 的提示更加智能
 - ✓ 因为会自动匹配 TypeScript 类型和 JS 源码
- VSCode 支持吗
 - ✓ 我个人不怎么用 VSCode
 - ✓ 好像也支持，你可以自行尝试

目前还是皮毛

最重要、最难懂的请看下节课