

Chapter 0: Giới Thiệu Về Khóa Học	2
#0.1. Demo Kết Quả Đạt Được Khi Kết Thúc Khóa Học	2
#0.2. Yêu Cầu Của Khóa Học	3
#0.3. Về Khóa Học này	4
#0.4. Về Tác Giả	6
Chapter 1: Setup Environment	7
#1. Why Coding (Frontend) with Typescript	7
#2. Setup Environment	9
#3.1 CRA (Create-React-App) is dead ?	10
#3.2 Setup React với Typescript	12
#4. Giải Thích Cấu Trúc dự án với Typescript (Vite)	14
Chapter 2: React với Typescript	16
#5. Ôn tập kiến thức React với Javascript	16
#6. Tài liệu sử dụng React với Typescript	17
#7. Component	18
#8. Ôn tập Typescript Type	20
#9. Hiểu về State với TypeScript Generic	23
#10.1 Bài tập Khai Báo State đơn giản với Typescript	25
#10.2 Hướng Dẫn Làm Bài Tập Khai Báo State	25
#11. Khai Báo State với Object	26
#12. Types or Interfaces ?	28
#13. Props	29
#14. Default props	30
#15. Bài tập sử dụng với props	31
#16. Xử lý Event với Form	32
#17. Hoàn Thiện Form Add User	33
#18. Sử dụng Class Components	33
#19. Tổng kết các kiến thức đã học	34
Chapter 3: Thực Hành Dự Án Portfolio	35
#20. Phân tích dự án HTML	35
#21. Chạy dự án với React	36
#22. Chia layout Portfolio	37
#23. Hoàn thiện homeCV component	38
#24. Setup SCSS	39
#25. Setup Path Alias	40
#26. Tạo base component Leftpart	41
#27. Tạo base component About	42
#28. Bài tập hoàn thiện component About	43
#29. Tạo base component Skills	44
#30. Bài tập hoàn thiện component Skills	45

#31. Tạo base component Project	45
#32.1 Bài tập hoàn thiện component Projects (Part 1)	46
#32.2 Bài tập hoàn thiện component Projects (Part 2)	46
#33. Component Contact me	47
#34. Float button scroll to top	47
#35. Show/Hide left part	47
#36. Remove Animation with JQuery (Part 1)	47
#37. Remove Animation with JQuery (Part 2)	47
#38. Active leftside bar	48
#39. Add Slugs	48
#40. Review Giao Diện Mobile	48
Chapter 4: Deploy Production - Triển Khai Dự Án Thực Tế	49
#41. Chế Độ Build Production	49
#42. Triển Khai Dự Án với Github Page	50
#43. Mua Tên Miền	51
#44. Mapping tên miền tới Github page	52
#45. Học Gì Tiếp Đây	53
Lời Kết	54

Chapter 0: Giới Thiệu Về Khóa Học

#0.1. Demo Kết Quả Đạt Được Khi Kết Thúc Khóa Học

Video demo: <https://youtu.be/jiXYcoJYXGY>

Một vài highlight của khóa học:

- **Hiểu rõ, thực hành và áp dụng được ngôn ngữ Typescript cho dự án React**
- Tạo được CV online, triển khai online với Github Page (hoàn toàn miễn phí), rất có ích cho việc tìm việc làm

Link demo: <https://ericdev.click/>

#0.2. Yêu Cầu Của Khóa Học

Các yêu cầu cần biết trước khi thực hành khóa học:

- **Cần có sự hiểu biết cơ bản về Typescript.** Nếu bạn chưa biết, có thể học nhanh tại đây:
https://www.youtube.com/playlist?list=PLncHg6Kn2JT5emvXmG6kgeGkrQjRqxs_b4
- Cần biết về React.JS (dùng với javascript), nắm được các khái niệm cốt lõi như cách sử dụng hook cơ bản (useEffect, useState), sử dụng state và props

Một vài yêu cầu kiên quyết:

- **Cài đặt môi trường Node.JS :** trong khóa học này, máy tính của mình dùng **version Node.JS 16.13.1**
- Để hạn chế lỗi, các bạn nên dùng version trên. Có thể dùng NVM để sử dụng nhiều version của Node.JS trên cùng máy tính
Về NPM, xem tại:
https://www.youtube.com/watch?v=ccjKHLyo4IM&list=PLncHg6Kn2JT6E38Z3kit9Hnif1xC_9Vql&index=40
- **Cài đặt Git:** các thao tác với Git và sử dụng Github, mình không hướng dẫn. Bạn nào chưa biết dùng Git, xem tại đây:
<https://www.youtube.com/watch?v=-BtolPy15fq&list=PLncHg6Kn2JT6nWS9MRjSnt6Z-9Rj0pAlo>
- **Cài đặt Visual Studio Code:** đây là công cụ dùng để code

#0.3. Về Khóa Học này

1. Tại sao mình làm khóa học này?

Mình làm khóa học này để giúp các bạn hiểu và áp dụng Typescript với thư viện React.

Sau khi đã học React & Typescript cơ bản, đây chính là khóa học các bạn cần để biết cách áp dụng chúng với nhau, được sử dụng rất nhiều trong các dự án thực tế.

2. SOS thì làm sao

Với các bạn học viên Udemy, để đảm bảo quyền lợi, các bạn chủ động inbox qua fanpage Hỏi Dân IT để nhận được sự hỗ trợ trong quá trình học tập nhé:

<https://www.facebook.com/askITwithERIC/>

4. Về chuyện leak khóa học và mua lậu

Mình biết rất nhiều bạn khi học khóa học này của mình, là mua lậu qua bên thứ 3. chuyện này là hoàn toàn bình thường, vì thương hiệu “Hoi Dân IT” đang ngày càng khẳng định được vị thế của mình.

Nhiều bạn hỏi mình, sao mình không ‘chặn việc mua lậu’. nói thật, nếu mình làm, là làm được đấy, cơ mà nó sẽ gây ra sự bất tiện cho học viên chân chính (con sâu làm rầu nồi canh). Với lại, ngay cả hệ điều hành windows, còn bị crack nữa là @@

Mình cũng có 1 bài post facebook về chuyện này:

<https://www.facebook.com/askitwitheric/posts/pfbid02gyasktd3semgxat6nevnvwh4c8epzu3i7kpzhr7s7gmmfcvucyz96eb8avnvgnwhl>

Với các bạn học viên chân chính, mình tin rằng, những cái các bạn nhận được từ mình khi đã chấp nhận đầu tư, nó sẽ hoàn toàn xứng đáng. vì đơn giản, với cá nhân mình, khách hàng là thượng đế.

VỚI CÁC BẠN MUA LẬU, MÌNH CHỈ MUỐN CHIA SẺ THỂ NÀY:

1. TRÊN ĐỜI NÀY, CHẲNG CÓ GÌ CHẤT LƯỢNG MÀ MIỄN PHÍ CẢ.

VIỆC BẠN MUA LẬU QUA BÊN THỨ 3, LÀ GIÚP BỌN CHÚNG LÀM GIÀU VÀ GÂY THIẾT HẠI CHO TÁC GIẢ.

NẾU NHÌN VỀ TƯƠNG LAI => Càng ngày càng ít tác giả làm khóa học => NGƯỜI BỊ HẠI CUỐI CÙNG VẪN LÀ HỌC VIÊN

2. HÃY HỌC THÓI QUEN TRÂN TRỌNG GIÁ TRỊ LAO ĐỘNG

NÓ LÀ THÓI QUEN, CŨNG NHƯ SẼ LÀ MỘT PHẦN TÍNH CÁCH CỦA BẠN.

ĐỪNG VÌ NGHÈO QUÁ MÀ LÀM MẤT ĐI TÍNH CÁCH CỦA BẢN THÂN.

NẾU KHÓ KHĂN, CỨ INBOX MÌNH, MÌNH HỖ TRỢ. VIỆC GÌ PHẢI LÀM VẬY =))

3. MÌNH ĐÃ TỪNG LÀ SINH VIÊN GIỐNG BẠN, MÌNH HIỂU TẠI SAO CÁC BẠN LÀM VẬY. HÃY BIẾT CHO ĐI. SỐNG ÍCH KỶ, THÌ THEO LUẬT NHÂN QUẢ ĐẤY, CHẲNG CÓ GÌ LÀ NGẪU NHIÊN CẢ

4. NẾU BẠN THẤY KHÓA HỌC HAY, HÃY BIẾT DONATE ĐỂ ỦNG HỘ TÁC GIẢ. LINK DONATE: <https://hoidanit.github.io/official/donate>

Hành động nhỏ nhưng mang ý nghĩa lớn. Hãy vì 1 cộng đồng IT Việt Nam phát triển. Nếu làm như các bạn, có lẽ chúng ta đã không có Iphone, không có Apple như ngày nay rồi @@

#0.4. Về Tác Giả

Mọi thông tin về Tác giả Hỏi Dân IT, các bạn có thể tìm kiếm tại đây:

Website chính thức: <https://hoidanit.com.vn/>

Youtube “Hỏi Dân IT” : <https://www.youtube.com/@hoidanit>

Tiktok “Hỏi Dân IT” : <https://www.tiktok.com/@hoidanit>

Fanpage “Hỏi Dân IT” : <https://www.facebook.com/askITwithERIC/>

Udemy Hỏi Dân IT: <https://www.udemy.com/user/eric-7039/>

Chapter 1: Setup Environment

#1. Why Coding (Frontend) with Typescript

React, Vue => (optional) Vanilla Javascript / Typescript

Angular => only Typescript

1. Tại sao chọn Javascript ?

- Vì javascript phổ biến, dễ học (phù hợp với beginners, chưa từng học ngôn ngữ OOP)

- Rút ngắn thời gian phát triển sản phẩm (code ít hơn typescript :v)

- Phù hợp với dự án nhỏ /team nhỏ => ko quan tâm tới việc mở rộng/tối ưu/bảo trì code
=> code cho chạy được

- => người code không cần IDE gợi ý => cần hiểu mình đang làm gì :v

2. Khi nào cần Typescript (why) ?

- Dự án lớn (có nhiều thành viên tham gia/nhiều chức năng)

- => khối lượng codebase lớn

- => quan tâm tới việc mở rộng/tối ưu/bảo trì code

- Gợi ý code (từ IDE) khi code

- Thông báo lỗi khi dịch code (compile), not runtime (sau khi đã build và chạy)

- => chất lượng sản phẩm cao hơn, ít bugs

3. Những khó khăn khi dùng với Typescript ?

- Biết javascript là chưa đủ => cần học cú pháp của typescript

- Code nhiều hơn javascript :v

Đổi lại:

- Gợi ý code (từ IDE) khi code

- Thông báo lỗi khi dịch code (compile), not runtime (sau khi đã build và chạy)

- => chất lượng sản phẩm cao hơn, ít bugs

4. Để học 1 framework/library với Typescript cần gì ?

- Có hiểu biết về javascript/typescript

- Có hiểu biết về OOP. Nếu bạn đã biết các ngôn ngữ như C#, Java... đây là lợi thế

- Có khả năng đọc source code (optional)

- Search Google (always) :v => làm nhiều thì quen tay

Give it a try ?

- Bởi vì có thể code song song 1 dự án, dùng cả Javascript lẫn Typescript :v

- Có thể code javascript trong file typescript, còn ngược lại thì không.
(Không thể sử dụng cú pháp TS trong file JS)

Xu hướng chuyển dịch của thư viện/framework đang là Typescript ???

Dev dùng typescript => đỡ bugs

Máy (dùng javascript, version đã compile từ mã nguồn TS được viết)

=> như vậy javascript không die, mà chẳng qua, cách code javascript cần khác đi

=> nâng cao chất lượng sản phẩm, đảm bảo tính scale/maintain (các doanh nghiệp cần cái này)

#2. Setup Environment

1. Node.js

<https://nodejs.org/en/download/releases/>

Không nên sử dụng version mới nhất của Node.JS, vì rất nhiều thư viện chưa kịp hỗ trợ

=> ở trong khóa học này, mình sử dụng node v16.13.1 (nhỏ hơn <=16):

<https://nodejs.org/download/release/v16.13.1/>

=> nếu muốn cài đặt nhiều version của Node.JS trên cùng 1 máy tính, sử dụng nvm

<https://github.com/coreybutler/nvm-windows>

Check version của node.js

node -v

lưu ý: trong series này, khuyến khích sử dụng version node 16x để hạn chế tối đa lỗi

2. IDE

Sử dụng Visual Studio Code (VSCode): <https://code.visualstudio.com/download>

- Không cần thiết phải cài đặt extension, vì mặc định, VSCode hỗ trợ rất tốt cho typescript

Có thể tham khảo:

<https://react-typescript-cheatsheet.netlify.app/docs/basic/setup#vs-code-extensions>

#3.1 CRA (Create-React-App) is dead ?

Warning: chỉ xem video khi bạn "đã từng dùng CRA" trước đây

Link video: <https://youtu.be/1kSY7cCo700>

1. Dòng thời gian

2018: React Hook ra đời -> tạo ra xu hướng chuyển dịch từ **React Class => React Hook**

2022: React 18 ra đời => hỗ trợ mạnh mẽ SSR (server side rendering) với React

Theo thời gian, công nghệ thay đổi, và CRA cũng không ngoại lệ.

2. CRA (create-react-app)

Link trang chủ: <https://create-react-app.dev/>

Link github: <https://github.com/facebook/create-react-app>

=> để ý là, create-react-app vẫn thuộc facebook, nên nó sẽ không dead, mà chẳng qua, sẽ không phát triển nữa :v

Chỉ khi nào, 2 trang web ở trên, có thông báo là CRA is dead => chúng ta mới tin nhé

3. Lựa chọn công nghệ cho Frontend React.

Tương lai của CRA, được giải thích tại đây:

<https://github.com/reactjs/react.dev/pull/5487#issuecomment-1409720741>

CRA sẽ trở thành 1 công cụ, cho phép bạn lựa chọn "cách code React", ví dụ như:

1. Code React với Next.JS

2. Code React với Remix

...

n. Code React với Create-react-app

Nguyên nhân: theo dòng thời gian, website nó tiến hóa. Các framework hiện đại sẽ là con lai (lai giữa Frontend và Backend).

Bạn có thể code React ở phía server, như cách bạn vẫn làm lâu nay với CRA :v

=> modern frameworks are hybrid. they let you build SPAs *and more*

4. Giải pháp

- Về tương lai, do công nghệ thay đổi => dẫn tới các công ty thay đổi => frontend dev sẽ cần học nhiều hơn :v

- Nếu mình dạy về Next.JS và Remix hay những framework tương tự, sẽ là "một thách thức" cho beginners, vì các công cụ này đòi hỏi phải "có hiểu biết về React".

- Thay vì dùng CRA, mình sẽ sử dụng Vite, một công cụ "chỉ học frontend để tạo SPA - single page app".

<https://vitejs.dev/guide/>

#3.2 Setup React với Typescript

Tài liệu:

<https://www.typescriptlang.org/docs/handbook/react.html>

1. Tích hợp Typescript vào dự án Javascript (code song song JS và TS)

- Phù hợp với dự án đã codebase từ Javascript
- Các tính năng mới phát triển = typescript, và ko gây ảnh hưởng tới code cũ

Các bước chính cần làm:

<https://stackoverflow.com/a/74560171>

- Cài đặt dependencies, ví dụ: typescript @types/node @types/react @types/react-dom @types/jest
- Tạo file config cho typescript: tsconfig.json (cấu hình file này để có thể dịch code Typescript sang javascript)
- Sử dụng typescript với tên file (ts hoặc tsx)

2. Tạo mới dự án React với typescript

- Có thể sử dụng CRA (create react app) or Vite để rút ngắn thời gian
<https://create-react-app.dev/docs/getting-started#creating-a-typescript-app>

- Setup với Vite:

Lưu ý: Môi trường cài đặt trong video, sử dụng node.js version 16x

Để sử dụng version "mới nhất" của Vite, sử dụng câu lệnh: `npm create vite@latest`

Tuy nhiên, để hạn chế lỗi tối đa (do thư viện update, chẳng may có breaking change), sử dụng câu lệnh sau:

`npm create vite@4.2.0`

<https://www.npmjs.com/package/vite>

trong trường hợp không thể cài được version 4.2.0, clone code tại link github sau:
<https://github.com/haryphamdev/react-ts-vite>

Cài đặt như trên, là để đảm bảo code của mình và bạn giống hệt nhau. còn về sau này, sau khi đã có kiến thức cơ bản, bạn thích cài version nào, là quyền của bạn, ok ?

Hỏi Dân IT với Eric

#4. Giải Thích Cấu Trúc dự án với Typescript (Vite)

Tương tự như dự án sử dụng CRA (create react app), hoặc dự án sử dụng javascript thuần, cấu trúc với typescript là giống hệt, chỉ bổ sung thêm config cho typescript

Lưu ý: không nên học thuộc tác dụng của các file làm gì, nên hiểu tác dụng của chúng. quên cũng chẳng sao.

Điều quan trọng là bạn hiểu cách hoạt động "của React". Cái gì quên thì google.

1. TypeScript files

<https://www.typescriptlang.org/docs/handbook/tsconfig-json.html>

File chính: tsconfig.json (có thể bao gồm tsconfig.node.json)

Các file config này phụ trách nhiệm vụ nói cho trình "compiler" biết cách dịch code. Cụ thể, chúng ta cần dịch code typescript thành javascript thì browser mới có thể chạy.

=> phụ trách nhiệm vụ "dịch code" từ typescript sang javascript

File phụ: vite-env.d.ts

d là viết tắt của declare => tác dụng của file này là giúp định nghĩa type => phục vụ quá trình type checking của ts

<https://stackoverflow.com/questions/21247278/about-d-ts-in-typescript>

<https://www.typescriptlang.org/docs/handbook/declaration-files/templates/module-d-ts.html>

<https://www.typescriptlang.org/docs/handbook/triple-slash-directives.html>

2. React files

Code React bao gồm CSS và Component (JSX). Đối với dự án React, có thể định nghĩa component bằng cách khai báo file:

- kết thúc với js (jsx)
- kết thúc với tsx

3. Các file khác

vite.config.ts => cấu hình Vite (ví dụ như port, plugin...) (nếu dùng js sẽ là vite.config.js)

package.json (và package-lock.json): lưu trữ cấu hình các thư viện đã cài đặt cho dự án

.gitignore : khai báo file KHÔNG ĐƯỢC ĐẨY LÊN github, gitlab... (remote hub)

index.html (khác với cách truyền thống, là file index.html nằm trong thư mục public) => đây là nơi dịch code sẽ được lưu (code React)

<https://vitejs.dev/guide/#index-html-and-project-root>

4. Cách thức project hoạt động

Để chạy project, chúng ta có sử dụng script bên trong file package.json:

```
{
  "scripts": {
    "dev": "vite", // start dev server => chạy tại localhost
    "build": "vite build", // build for production (chạy thực tế)
    "preview": "vite preview" // locally preview production build => chạy câu lệnh này để
    xem trước sản phẩm sẽ triển khai
  }
}
```

Như vậy, để chạy project, mặc định, chúng ta sử dụng câu lệnh: npm run dev

Khi chạy câu lệnh trên, quá trình sau sẽ diễn ra:

- Trình compiler sẽ dịch tất cả file typescript (.ts) thành javascript (.js)
- Build file index.html : dịch tất cả code React (.tsx/css...), sau đó nhúng vào div với id = "root"
- Khi test tại localhost => thực chất là sử dụng file index.html ở trên :v

Chapter 2: React với Typescript

#5. Ôn tập kiến thức React với Javascript

Yêu cầu: hiểu hook cơ bản: useState và useEffect

- Hiểu tác dụng của State và Props

Vấn đề phát sinh:

- Chưa check type của dữ liệu (ví dụ khi khai báo state)
- Không biết được component cần truyền data gì (ví dụ khi truyền props)

#6. Tài liệu sử dụng React với Typescript

Link: <https://react-typescript-cheatsheet.netlify.app/>

Thực chất là chúng ta khai báo component bình thường (giống với javascript), chỉ có điều định nghĩa thêm type cho typescript :v

Hỏi Dân IT với Eric

#7. Component

Yêu cầu: (với javascript)

- Đã hiểu tác dụng của component
- Đã biết cách khai báo và sử dụng component
- Sử dụng function component

Tài liệu:

https://react-typescript-cheatsheet.netlify.app/docs/basic/getting-started/function_components

1. Convert từ javascript sang typescript

- thay vì định nghĩa file .js (.jsx) => định nghĩa .ts
- tên component sẽ viết Hoa chữ đầu tiên

Cách 1:

```
const App = () => {  
  return ( <div> Hello world ts </div> )  
}
```

Cách 2:

```
const App = () => <div> Hello world ts </div>
```

Lưu ý về code: khuyến khích viết theo cách 1, như vậy cấu trúc rõ ràng (giống OOP), đồng thời, sau này cần thêm logic xử lý thì không cần sửa lại cấu trúc function

- Chuyển sang function component với typescript: the same ==))

Với cách viết 1 ở trên, typescript đủ thông minh để biết đây là component (khi return JSX element)

Lưu ý về cách định nghĩa function component:

Không nên dùng các cách sau:

Link giải thích lý do:

<https://github.com/facebook/create-react-app/pull/8177>

```
const App: React.FunctionComponent<{}> = () => (  
  <div> Hello world ts </div>  
)
```

hoặc

```
const App: React.FC = () => (  
  <div> Hello world ts </div>  
)
```

hoặc

```
const App: React.VoidFunctionComponent<{}> = () => (  
  <div> Hello world ts </div>  
)
```

#8. Ôn tập Typescript Type

Tài liệu:

Type Reference: <https://www.typescriptlang.org/docs/handbook/type-inference.html>

Union Type: <https://www.typescripttutorial.net/typescript-tutorial/typescript-union-type/>

Object Type:

<https://www.typescripttutorial.net/typescript-tutorial/typescript-object-type/>

Thông thường, khi báo biến (variable) hoặc hàm (function), chúng ta sẽ khai báo 'type' để cho typescript biết và check.

Ví dụ:

```
let name:string = "eric" ; //type = string  
name = 10; //error => vì gán string cho number
```

1. Type Inference (Tự động gán type)

Ví dụ:

```
let name = "eric" ; //type = string
```

Mặc dù không khai báo 'type', typescript 'sẽ cố gắng' gán type cho biến khai báo => amazing

2. Union Type

Ví dụ: biến name vừa là string, vừa là number

```
let name: string | number = "eric";  
name = 10; //ok
```

Union type giúp thông báo 1 biến 'có thể có' nhiều giá trị

EU: European Union :v

3. Object Type

Lưu ý: object với chữ 'o' viết thường

Cách 1:

```
let employee: object;  
employee = {  
  firstName: 'Eric',  
  lastName: 'Pham',  
  age: 25,  
  jobTitle: 'Web Developer'  
};  
console.log(employee);  
employee = 10; //error
```

Cách 2:

- khai báo type:

```
let employee: {  
  firstName: string;  
  lastName: string;  
  age: number;  
  jobTitle: string;  
};
```

- gán giá trị:

```
employee = {  
  firstName: 'Eric',  
  lastName: 'Pham',  
  age: 25,  
  jobTitle: 'Web Developer'  
};
```

Cách 3:

```
let employee: {  
  firstName: string;  
  lastName: string;  
  age: number;  
  jobTitle: string;  
} = {  
  firstName: 'Eric',
```

```
lastName: 'Pham',  
age: 25,  
jobTitle: 'Web Developer'  
};
```

Cách 4: Object tương tự như class (đối tượng) trong OOP

- Sử dụng Type:

```
type TEmployee = {  
  firstName: string;  
  lastName: string;  
  age: number;  
  jobTitle: string  
}
```

```
let employee: TEmployee = {  
  firstName: 'Eric',  
  lastName: 'Pham',  
  age: 25,  
  jobTitle: 'Web Developer'  
};
```

hoặc:

- Sử dụng interface:

```
interface IEmployee {  
  firstName: string;  
  lastName: string;  
  age: number;  
  jobTitle: string  
}
```

```
let employee: IEmployee = {  
  firstName: 'Eric',  
  lastName: 'Pham',  
  age: 25,  
  jobTitle: 'Web Developer'  
};
```

#9. Hiểu về State với TypeScript Generic

1. Type inference (tự động gán type)

```
const name = "eric" ; // console.log(typeof name) === string
```

```
const [name, setName] = useState("eric")
```

=> typescript tự động đoán và gán type dựa vào giá trị init cho state.
Tuy nhiên, cách viết này không tường minh, đặc biệt, khi sử dụng cho object.

2. Type annotation

- Tìm hiểu về typescript generic:

<https://www.typescripttutorial.net/typescript-tutorial/typescript-generics/>

https://www.w3schools.com/typescript/typescript_basic_generics.php

=> generic được sử dụng, khi "chưa biết trước kiểu type" (được dùng nhiều khi viết function tổng quát/reusable code)

Ví dụ:

```
//function useState<S>(initialState: S | (() => S)): [S, Dispatch<SetStateAction<S>>];
```

Hàm useState là 1 generic, có thể truyền vào initialState với kiểu type (S - chưa biết trước).

hàm này sẽ trả về 2 giá trị [S, Dispatch<SetStateAction<S>>]

=> đây là 1 array, phần tử đầu tiên là biến với type = S, phần tử thứ 2 là function giúp cập nhật giá trị.

Cụ thể:

```
const [name, setName] = useState("eric");
```

ở đây, biến truyền vào có kiểu giá trị là "string"

```
useState<string>(initialState: string | (() => string)): [string,  
Dispatch<SetStateAction<string>>];
```


Do hàm `useEffect` trả về 1 array theo cú pháp trên => `const [name, setName] = ...` có nghĩa là:

biến `name` (phần tử đầu tiên) sẽ có kiểu giá trị là `string`

biến `setName` sẽ có khả năng cập nhật được biến `name`, do kiểu giá trị là `Dispatch<SetStateAction<string>>`

```
const [user, setName] = useState<string>("eric")
```

#10.1 Bài tập Khai Báo State đơn giản với Typescript

- convert từ javascript sang typescript, với các kiểu dữ liệu: string, number, boolean, array

#10.2 Hướng Dẫn Làm Bài Tập Khai Báo State

chưa hướng dẫn cách khai báo với object

Hỏi Dân IT với Eric

#11. Khai Báo State với Object

1. Vấn đề với object

Ví dụ:

- Khi dùng với javascript:

```
const [user, setUser] = useState({name: 'eric', age: 25})
```

```
setUser(10); //ok
```

```
setUser({name: 10, age: 'eric'}); //ok
```

- Khi dùng với typescript:

```
const [user, setUser] = useState({name: 'eric', age: 25})
```

```
setUser(10); //error
```

```
setUser({name: 10, age: 'eric'}); //error
```

Tuy nhiên, với cách khai báo trên, chúng ta 'không thể tái sử dụng được type User, đặc biệt là khi muốn export nó để sử dụng ở nơi khác'

-> giải pháp đề ra là định nghĩa type => code tường minh và rõ ràng hơn

2. Khai báo với Object

Cách 1: dùng trực tiếp với generic

```
const [user, setUser] = useState<{name: string, age: number}>({name: 'eric', age: 25})
```

Cách 2: định nghĩa type

```
type TUser = {
```

```
  name: string;
```

```
  age: number;
```

```
}
```

```
const [user, setUser] = useState<TUser>({name: 'eric', age: 25})
```

Cách 3: định nghĩa interface (khuyến khích sử dụng)

```
interface IUser {
```

```
  name: string;
```

```
  age: number;
```

```
}
```

```
const [user, setUser] = useState<IUser>({name: 'eric', age: 25})
```

3. Khai báo với null/undefined

```
const [user, setUser] = useState<IUser>({name: 'eric', age: 25})
```

Với cách khai báo trên, chúng ta cần khởi tạo giá trị mặc định cho biến 'user'

```
const [user, setUser] = useState<IUser>(null) //error
```

Để giải quyết vấn đề trên, chúng ta sử dụng union type:

```
const [user, setUser] = useState<IUser | null>(null)
```

=> như vậy, biến user sẽ có kiểu giá trị là IUser, hoặc là null => no error :v

#12. Types or Interfaces ?

https://react-typescript-cheatsheet.netlify.app/docs/basic/getting-started/basic_type_example#types-or-interfaces

Hỏi Dân IT với Eric

#13. Props

- Yêu cầu: đã có hiểu biết về Props (cách sử dụng/tác dụng của props)

Về cú pháp destructuring:

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Destructuring_assignment#object_destructuring

1. Defining props for a component

- Sử dụng interface

2. Optional props

=> edit interface with ?

#14. Default props

Tài liệu:

https://react-typescript-cheatsheet.netlify.app/docs/basic/getting-started/default_props

- Set default value for interface ?

interface => compile time

default value => run time

- set default value => set trực tiếp giá trị cho props

#15. Bài tập sử dụng với props

todo: truyền động default value với project => render form

Hỏi Dân IT với Eric

#16. Xử lý Event với Form

Tài liệu:

https://react-typescript-cheatsheet.netlify.app/docs/basic/getting-started/forms_and_events

Xử lý các event hay gặp như onClick, onChange...

Hỏi Dân IT với Eric

#17. Hoàn Thiện Form Add User

todo:...

#18. Sử dụng Class Components

Tài liệu:

https://react-typescript-cheatsheet.netlify.app/docs/basic/getting-started/class_components

#19. Tổng kết các kiến thức đã học

Các nội dung trọng tâm đã học khi sử dụng React với typescript:

- **Cách khai báo Component:** sử dụng function component
https://react-typescript-cheatsheet.netlify.app/docs/basic/getting-started/function_components
- **Cách khai báo State :** hiểu rõ bản chất của định nghĩa type với Generic
<https://react-typescript-cheatsheet.netlify.app/docs/basic/getting-started/hooks#usestate>
- **Cách khai báo Props:** sử dụng interface và default props /optional props
https://react-typescript-cheatsheet.netlify.app/docs/basic/getting-started/default_props
- **Sử dụng event với form :** xử lý event
https://react-typescript-cheatsheet.netlify.app/docs/basic/getting-started/forms_and_events

- Các kiến thức chưa đề cập ?

Chapter 3: Thực Hành Dự Án Portfolio

#20. Phân tích dự án HTML

Link source code:

<https://github.com/jemes888/Arlo---Personal-Portfolio-Resume-Template>

Link backup: <add link google drive>

Link demo: <https://jemes888.github.io/Arlo---Personal-Portfolio-Resume-Template/>

1. Test dự án chạy tại localhost

- Bố cục chia theo layout

- file css/js

#21. Chạy dự án với React

Lưu ý: checkout sang branch mới

- Tích hợp base component
- Tích hợp css/js
- Lưu ý về jquery

Hỏi Dân IT với Eric

#22. Chia layout Portfolio

- left side
- right side
- chia base screen **Portfolio**

Hỏi Dân IT với Eric

#23. Hoàn thiện homeCV component

Lưu ý: không sử dụng jquery vì tính 'không hiệu quả của nó'. Nếu bạn đã dùng framework frontend thì không nên dùng.

Còn nếu dùng html/css thuần, jquery là lựa chọn không hề tệ.

1.Text animation

Cài đặt:

npm install --save-exact react-type-animation@3.0.1

yarn add react-type-animation@3.0.1

2. Import images

//todo

#24. Setup SCSS

Download source code trong video:

https://drive.google.com/file/d/10Adofmd0s8_VlqBU9YoZS2s36jswU5o-/view?usp=share_link

1. Setup sass (scss)

Cài đặt thư viện:

npm install --save-exact sass@1.62.1

yard add sass@1.62.1

- Convert css to scss: <https://www.cssportal.com/css-to-scss/>
- Update import css

#25. Setup Path Alias

Tài liệu:

<https://dev.to/avxkim/setup-path-aliases-w-react-vite-ts-poa>

<https://stackoverflow.com/questions/75201705/how-to-set-multiple-aliases-in-vite-react/75201776#75201776>

Cài đặt thư viện:

npm i --save-exact @types/node@18.16.1 -D

yarn add --dev @types/node@18.16.1

#26. Tạo base component Leftpart

Import images với typescript:

<https://stackoverflow.com/questions/52759220/importing-images-in-typescript-react-cannot-find-module>

Hỏi Dân IT với Eric

#27. Tạo base component About

- remove send message button
- download files

Hỏi Dân IT với Eric

#28. Bài tập hoàn thiện component About

- Fill thông tin cá nhân
- Việt hóa các thông tin quan trọng

không nên ghi số điện thoại cá nhân => dễ bị spam

Giới thiệu về bản thân (sở thích, đam mê, mục tiêu nghề nghiệp...)

lưu ý về chuyên ngành (dân trái ngành)

- có thể không ghi => ghi điểm CPA
- nếu CPA quá thấp (thì không ghi CPA và chuyên ngành)

Ghi link Github, LinkedIn

#29. Tạo base component Skills

- việt hóa thông tin
- thêm section menu ứng với #skills

Cài đặt thư viện:

<https://www.npmjs.com/package/@ramonak/react-progress-bar>

npm install --save-exact @ramonak/react-progress-bar@5.0.3

yarn add @ramonak/react-progress-bar@5.0.3

Chia thành 3 loại kỹ năng: frontend, backend và kỹ năng khác

#30. Bài tập hoàn thiện component Skills

//todo

#31. Tạo base component Project

- Rename Portfolio => project
- Remove services
- Việt hóa thông tin

Cài đặt thư viện antd:

<https://ant.design/>

npm install --save-exact antd@5.4.6

yarn add antd@5.4.6

<https://ant.design/components/modal>

#32.1 Bài tập hoàn thiện component Projects (Part 1)

Cài đặt thư viện React icons:

<https://react-icons.github.io/react-icons/>

```
npm install --save-exact react-icons@4.8.0
```

```
yarn add react-icons@4.8.0
```

#32.2 Bài tập hoàn thiện component Projects (Part 2)

```
//todo
```

Hỏi Dân IT với Eric

#33. Component Contact me

-todo

...

#34. Float button scroll to top

<https://ant.design/components/float-button>

#35. Show/Hide left part

//todo

#36. Remove Animation with Jquery (Part 1)

Tài liệu:

<https://stackoverflow.com/questions/39195687/setting-a-backgroundimage-with-react-inline-styles>

<https://stackoverflow.com/a/62173627>

#37. Remove Animation with Jquery (Part 2)

Tài liệu:

<https://www.npmjs.com/package/parallax-js>

npm install --save-exact parallax-js@3.1.0 @types/parallax-js@3.1.0

<https://matthew.wagerfield.com/parallax/>

<https://github.com/wagerfield/parallax/issues/167+>

#38. Active leftside bar

//todo

#39. Add Slugs

<https://stackoverflow.com/questions/4282069/add-a-fragment-to-the-url-without-causing-a-redirect>

#40. Review Giao Diện Mobile

<https://www.npmjs.com/package/react-device-detect>

npm install --save-exact react-device-detect@2.2.3

yarn add react-device-detect@2.2.3

<https://stackoverflow.com/questions/42588171/how-to-perform-jquery-slidetoggle-equivalent-in-reactjs>

Chapter 4: Deploy Production - Triển Khai Dự Án Thực Tế

Lưu ý: về học chuyên sâu, hiểu rõ cách build và triển khai ứng dụng thực tế, mình đã có khóa học riêng về nó, cả frontend lẫn backend:

<https://hoidanit.com.vn/course/ultimate-guide-to-deploy-react-nodejs?id=640bee82f7099c369b3bc6a4>

Trong phạm vi khóa học này, do chỉ học mình frontend, nên sẽ có những hạn chế khi triển khai production.

Mục tiêu của chương này là chúng ta có được '1 đường link url' để có thể truy cập mọi lúc mọi nơi. Còn làm thực tế, các bạn cần học thêm kiến thức, cũng như là tham khảo khóa học ở trên.

#41. Chế Độ Build Production

1. Chế độ build

Vite hỗ trợ 2 câu lệnh sau:

```
"build": "tsc && vite build",  
"preview": "vite preview"
```

=> câu lệnh **npm run build** dùng để build project (code dùng cho production)

Câu lệnh **npm run preview** dùng để chạy project (sau khi đã build) ở môi trường localhost

2. Thống kê chế độ build production

Cài đặt thư viện:

npm install --save-dev rollup-plugin-visualizer@5.9.0

yarn add --dev rollup-plugin-visualizer@5.9.0

Tài liệu: <https://www.npmjs.com/package/rollup-plugin-visualizer>

Dùng với create-react-app: <https://create-react-app.dev/docs/analyzing-the-bundle-size/>

#42. Triển Khai Dự Án với Github Page

Tài liệu: <https://vitejs.dev/guide/static-deploy.html>

Hỏi Dân IT với Eric

#43. Mua Tên Miền

Trong khóa học này, mình lựa chọn “mắt bảo” để mua tên miền, một nhà cung cấp trong nước. (vì nó dễ thanh toán & chấp nhận bank nội địa)

Nếu bạn có thẻ visa/master card, có thể mua với nhà cung cấp nước ngoài.

Link trang chủ: <https://www.matbao.net/>

Lưu ý:

Trong quá trình mua tên miền, nếu bạn gặp khó khăn gì, thì đừng ngại gọi support của mắt bảo nhé. Thông tin nó ghi hết trên trang chủ ấy.

Yên tâm là Mắt Bảo thuộc top đầu của Việt Nam về kinh doanh tên miền các bạn nhé :v

Support 24/7

1900 1830 (1000 đ/phút)

Tư vấn miền Nam

(028) 3622 9999

Tư vấn miền Bắc

(024) 35 123456

#44. Mapping tên miền tới Github page

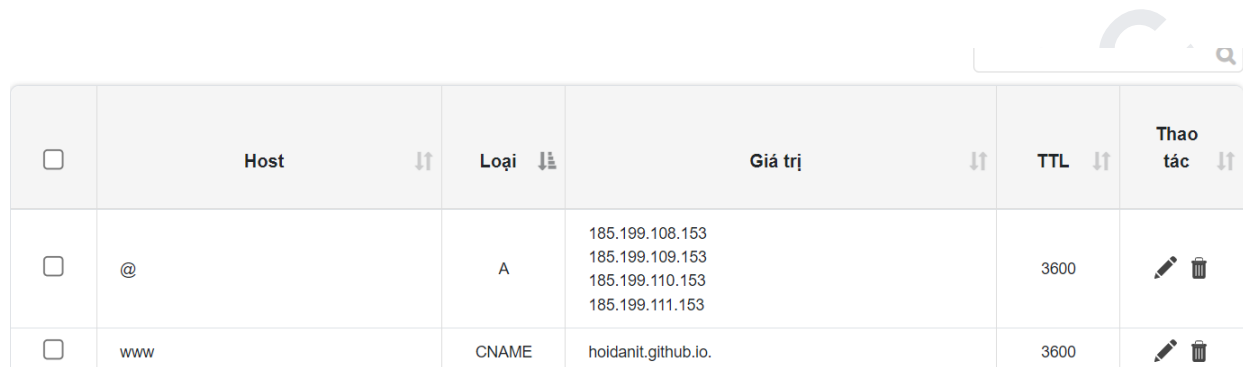
Cần cập nhật giải IP sau, tương tự như ảnh đính kèm





185.199.108.153

185.199.109.153

185.199.110.153

185.199.111.153



<input type="checkbox"/>	Host	Loại	Giá trị	TTL	Thao tác
<input type="checkbox"/>	@	A	185.199.108.153 185.199.109.153 185.199.110.153 185.199.111.153	3600	 
<input type="checkbox"/>	www	CNAME	holdanit.github.io.	3600	 

Trong quá trình xác minh tên miền, nếu bạn vừa mua tên miền xong, rồi thực hiện luôn video này,

Nếu có lỗi phát sinh vui lòng chờ đợi tối đa 24h để chắc chắn mọi thứ sẽ hoạt động chính xác bạn nhé :v

#45. Học Gì Tiếp Đây

Về lộ trình frontend của Hỏi Dân IT, các bạn xem tại đây:

<https://hoidanit.com.vn/huong-dan-su-dung-udemy>

Hỏi Dân IT với Eric

Lời Kết

Như vậy là chúng ta đã cùng nhau trải qua hơn 50+ videos về cách luyện mindset (cách tư duy) với React.

Tất cả các kiến thức mình chia sẻ, đều **được lấy từ kinh nghiệm đi làm của mình và...**

trang tài liệu typescript: <https://react-typescript-cheatsheet.netlify.app/>

Dĩ nhiên rằng, trong quá trình quá trình thực hiện khóa học này, mình sẽ không thể tránh khỏi những sai sót (vì nếu không có sai sót thì mình làm member của team React rồi :v).

Vì vậy, nếu thấy sai sót, các bạn cứ thoải mái đóng góp qua Fanpage Hỏi Dân IT nhé.

<https://www.facebook.com/askITwithERIC>

Nếu bạn thấy khóa học này hữu ích, đừng quên Review đánh giá trên Udemy để nhận được ưu đãi (giảm giá) cho các khóa tiếp theo nhé ^^

Hẹn gặp lại các bạn ở các khóa học tiếp theo

Hỏi Dân IT

Hỏi Dân IT với Eric