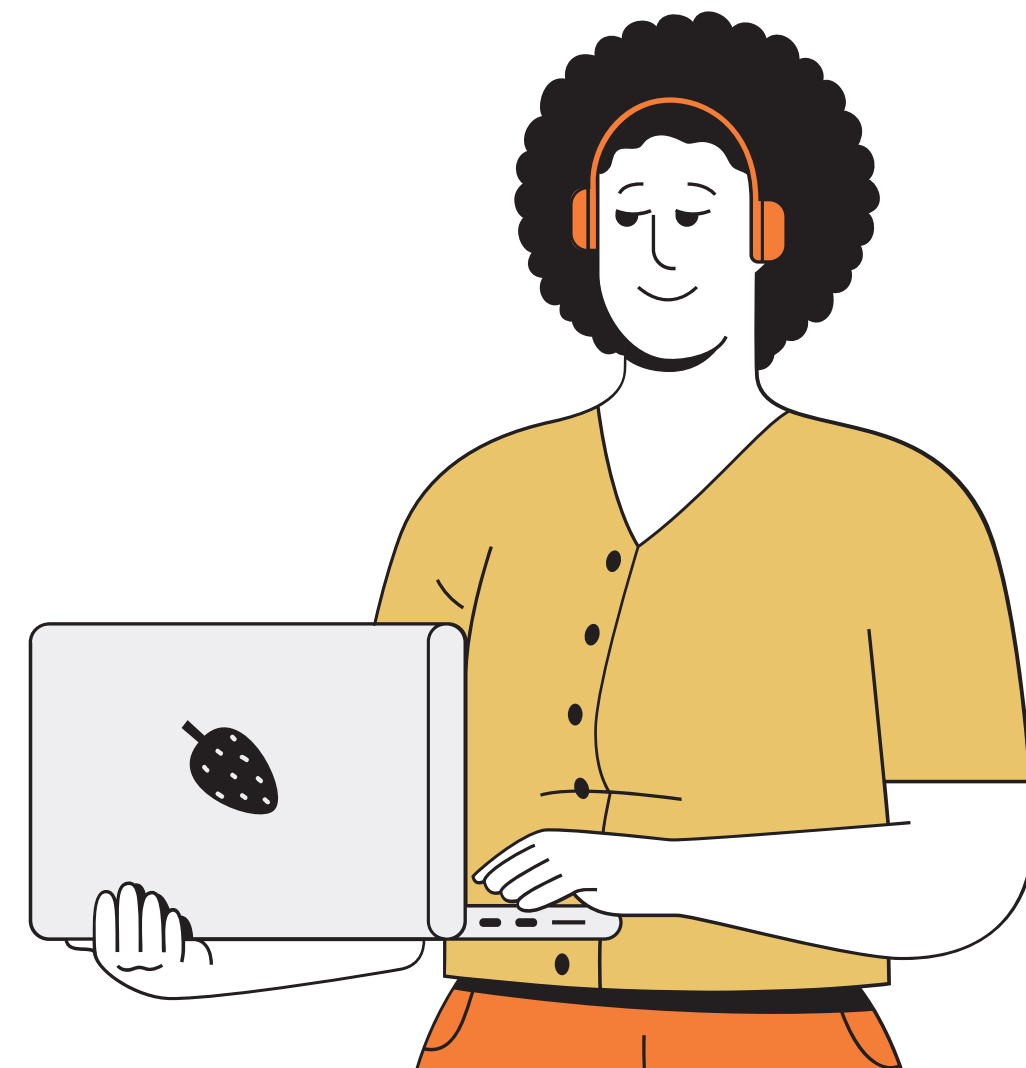
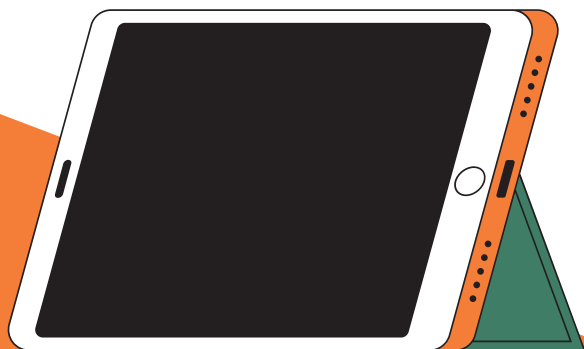


# Buổi 3 - Bổ sung kiến thức về React

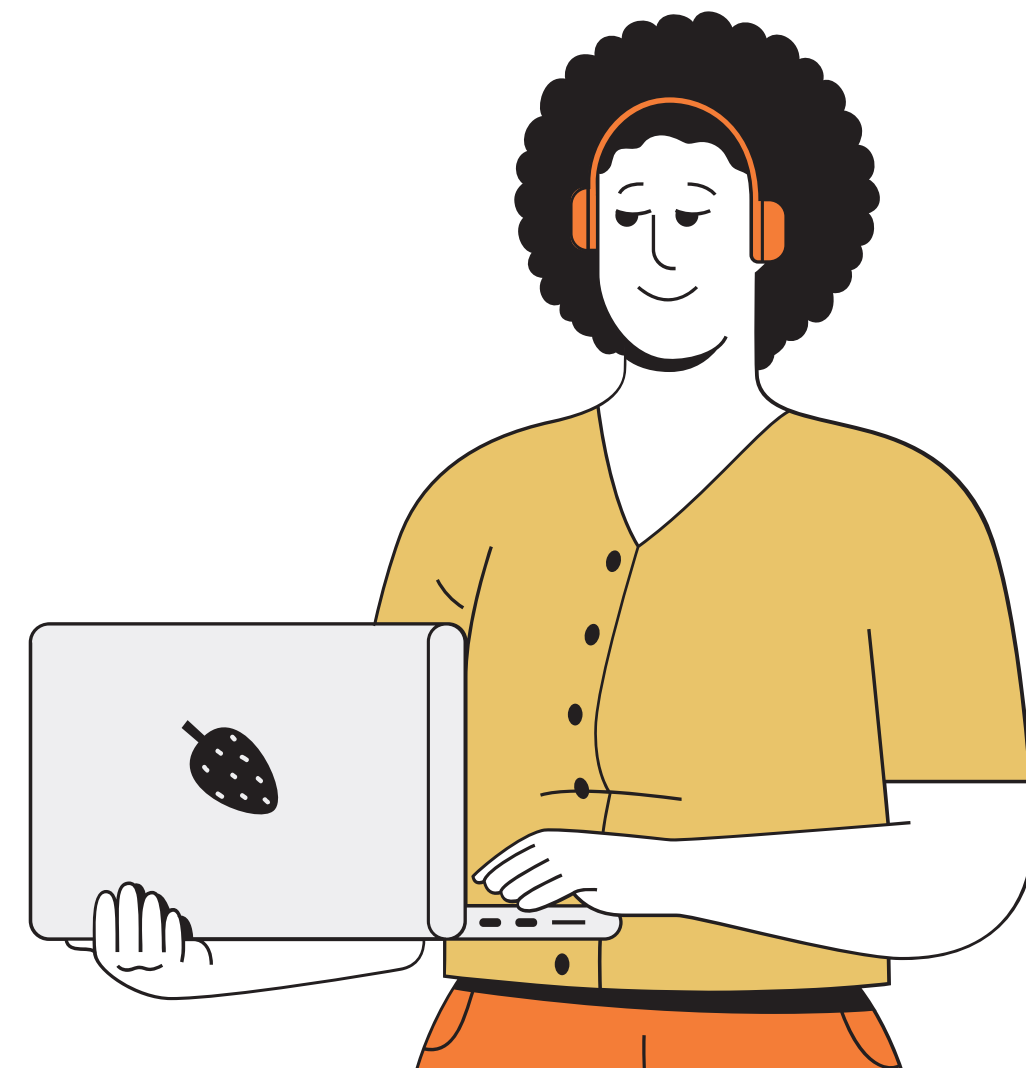


# Tổng quan

- Cơ bản về typescript.
- Các React hook thường dùng.
- ES6
- Styles
- Router
- Cơ bản của npm.
- Một số APIs mới của react.

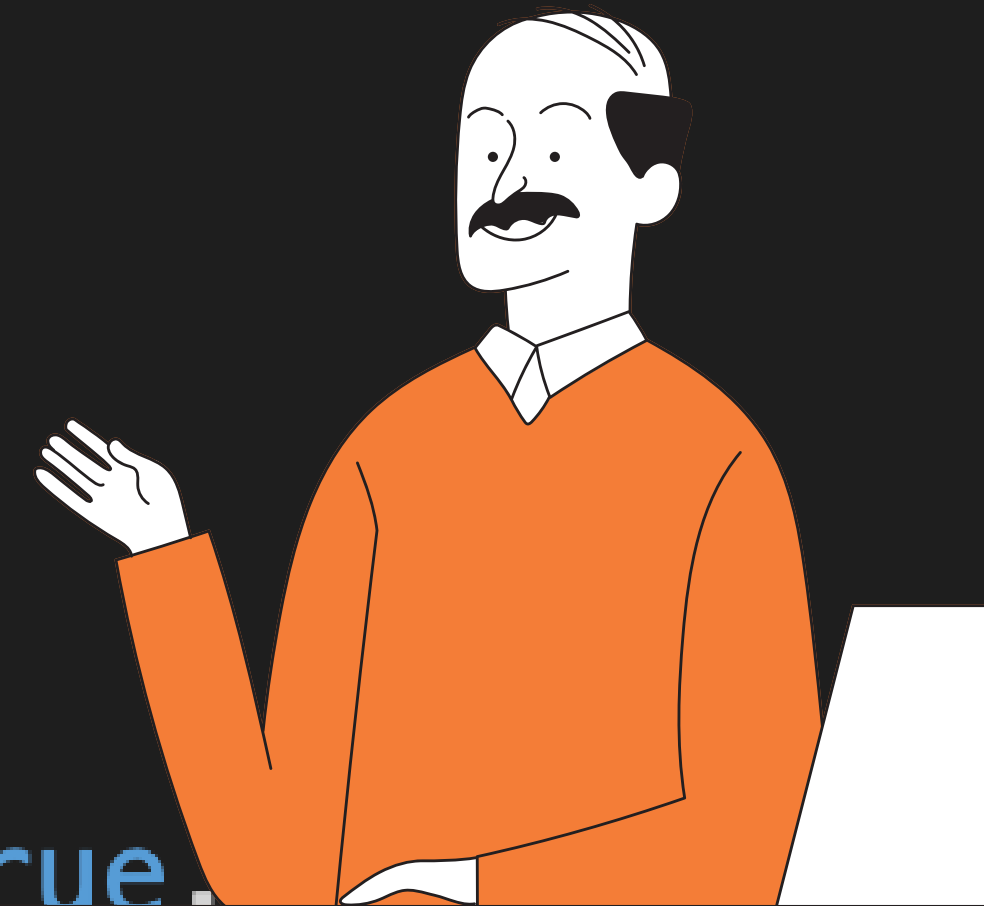


# Cơ bản về typescript



```
"compilerOptions": {  
  "target": "ESNext",  
  "useDefineForClassFields": true,  
  "lib": ["DOM", "DOM.Iterable", "ESNext"],  
  "allowJs": false,  
  "skipLibCheck": true,  
  "esModuleInterop": false,  
  "allowSyntheticDefaultImports": true,  
  "strict": true,  
  "forceConsistentCasingInFileNames": true,
```

Danny-98, 3 we

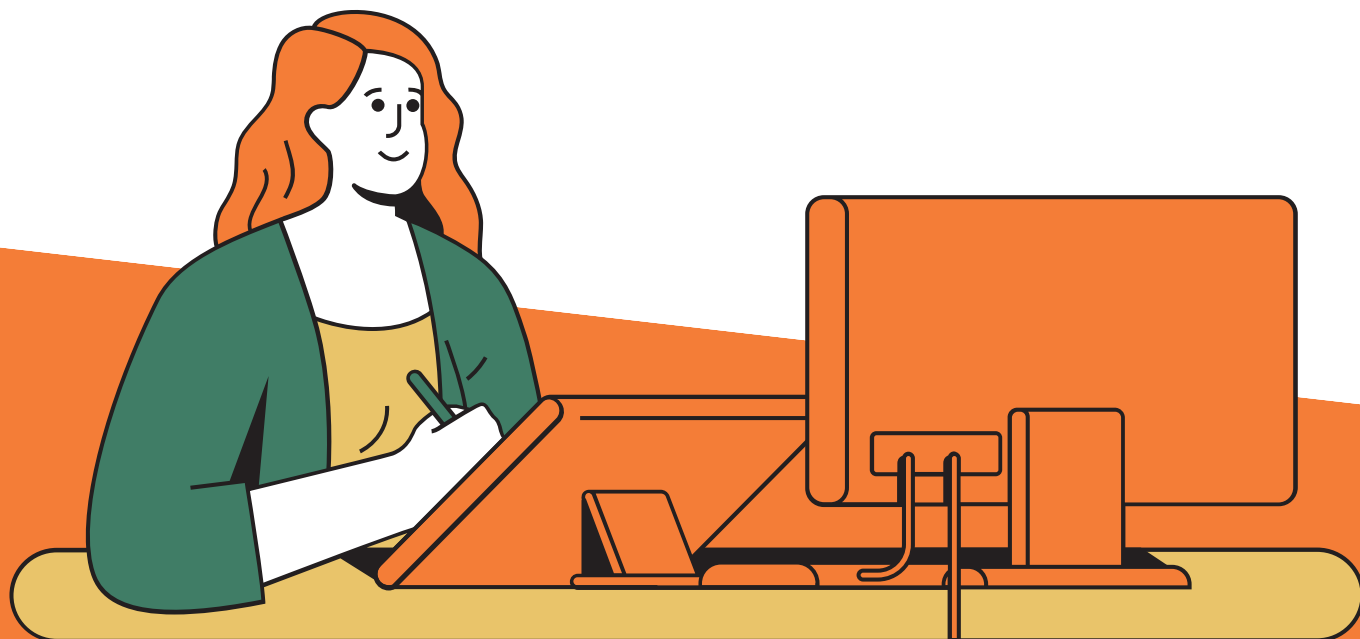


# Configuring the compiler

```
"baseUrl": "./src",  
"paths": {  
  "@modules/*": ["rest/modules/*"],  
  "@services/*": ["services/*"]  
}
```

# Alias import

Docs: <https://www.typescriptlang.org/tsconfig>

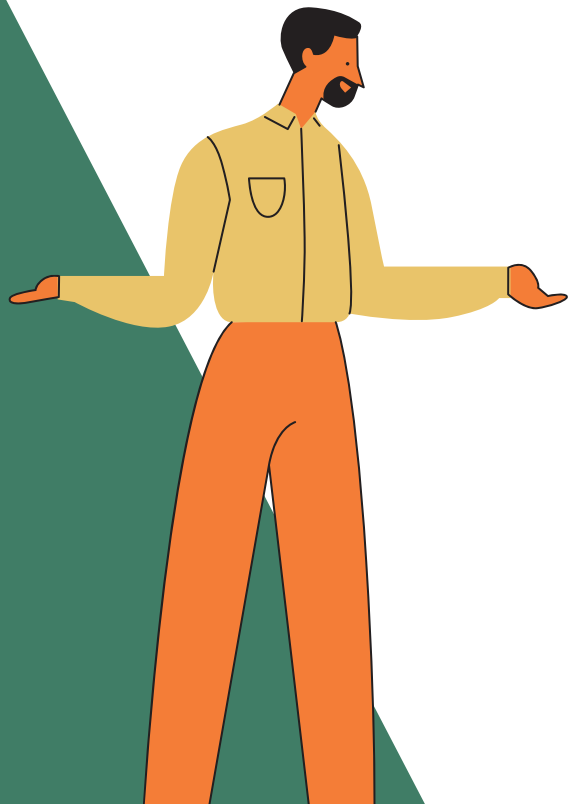


# Simple types

**boolean** - true hoặc false values

**number** - kiểu số thực và số nguyên

**string** - kiểu chuỗi "hello"



# Special types

- any - chấp nhận tất cả kiểu dữ liệu và cả lỗi liên quan -> trở về làm JS.
- unknown - tương tự any nhưng bị chặn khi sử dụng.
- never - không bao giờ xảy ra → một function(): never là không bao giờ retrun.
- void - function không trả về gì
- Union - có nhiều hơn một loại giá trị - > type ID = number | string.
- Casting - ép kiểu cho một biến → let test:unknown = "hello", test as string
- Keyof - trả về type chính xác của một object.



# React Hooks



Được ra mắt vào bản 16.8

Cho phép Function component sử dụng vòng đời

useState → tạo và theo dõi state trong function components

useEffect → sử dụng lifecycle

useContext → Context APIs giải pháp cho quản lý global state.

useRef → giữ nguyên values trong khi re-render, truy cập trực tiếp vào DOM

useReducer → tương tự useState nhưng có thể custom logic

useCallback → cache function

useMemo → cache Biến

Custom Hooks → khi có logic phức tạp cần dùng ở nhiều components, bắt đầu bằng `use`



# React ES6

Arrow functions → cú pháp ngắn hơn của function, không có `this`

Scope → giá trị let và const chỉ tồn tại trong phạm vi định nghĩa

Destructuring → lấy item của array hoặc object mà không cần dùng index hoặc key

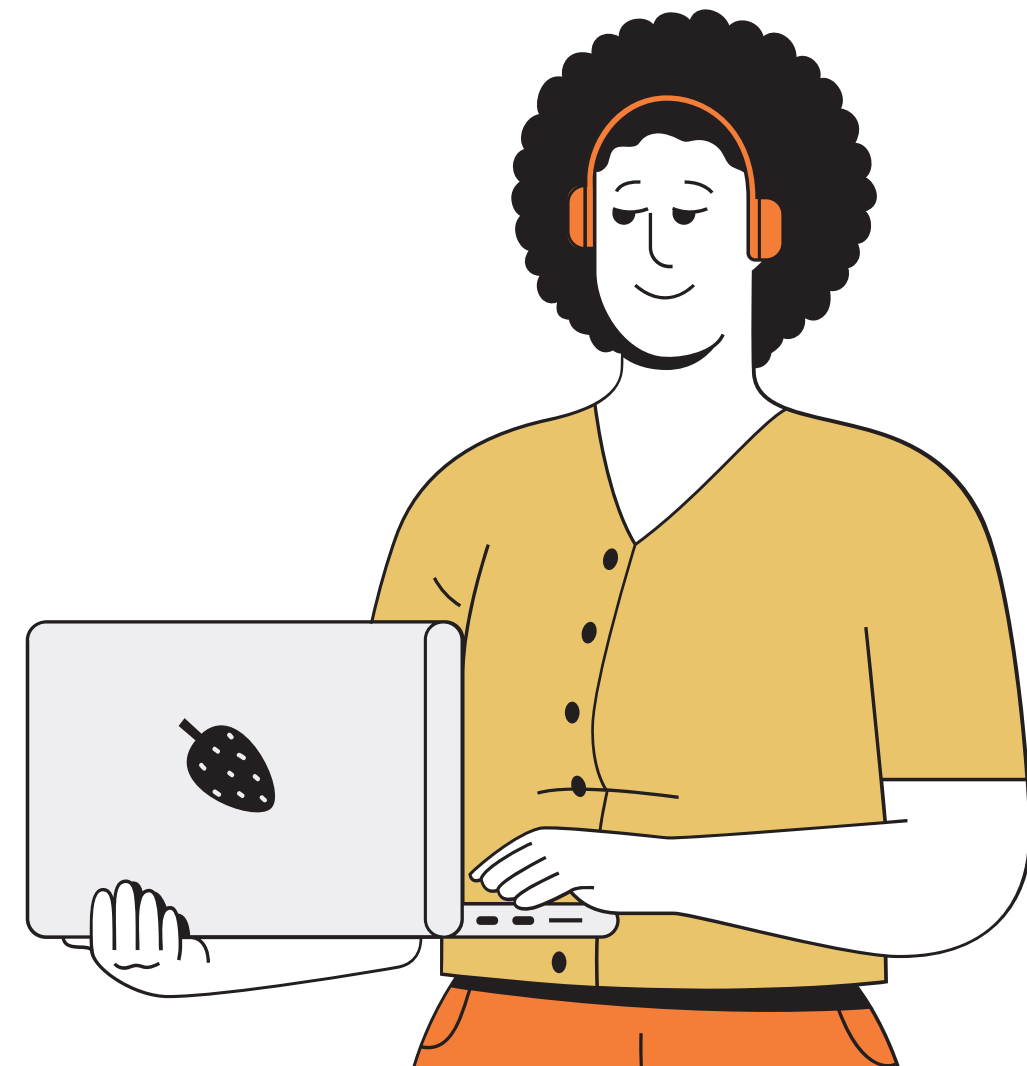
Spread Operator → copy nhanh các phần tử của array hoặc object

Module → cho phép import và export giúp chia code nhỏ hơn dễ bảo trì

Ternary Operator → toán tử 3 ngôi khi render component



# React Styles



```
const Header = () => {  
  return (  
    <>  
    <h1 style={{color: "red"}}>Hello Style!</h1>  
    <p>Add a little style!</p>  
    </>  
  );  
}
```



# Inline styles

import React from "react";      Danny-98, 3 weeks ago • first commit ...  
import ReactDOM from "react-dom/client";  
import "./index.css";  
import Router from "./router";  
  
ReactDOM.createRoot(document.getElementById("root") as HTMLElement).render(  
 // <React.StrictMode>  
 <Router />  
 // </React.StrictMode>  
);



# CSS Styles



# Module Styles

Tạo file có tên dạng `.module.css`  
Chỉ tồn tại trong module được  
import không lo trùng tên class  
hay id.



# React Router

sử dụng react-router-dom

# Cơ bản về npm



**Gỡ bỏ mọi thư viện đã cài đặt**

```
npm uninstall
```

**Package sẽ được thêm vào  
dependencies trong file  
package.json**

```
npm install  
packageName@version --save
```

**Package sẽ được thêm vào  
optionalDependencies trong file  
package.json**

```
npm install  
packageName@version --save-  
optional
```

**Cài đặt Global**

```
npm install -g  
packageName@version
```

**Package sẽ được thêm vào  
devDependencies trong file  
package.json**

```
npm install  
packageName@version --save-dev
```

**Package không được thêm vào  
package.json**

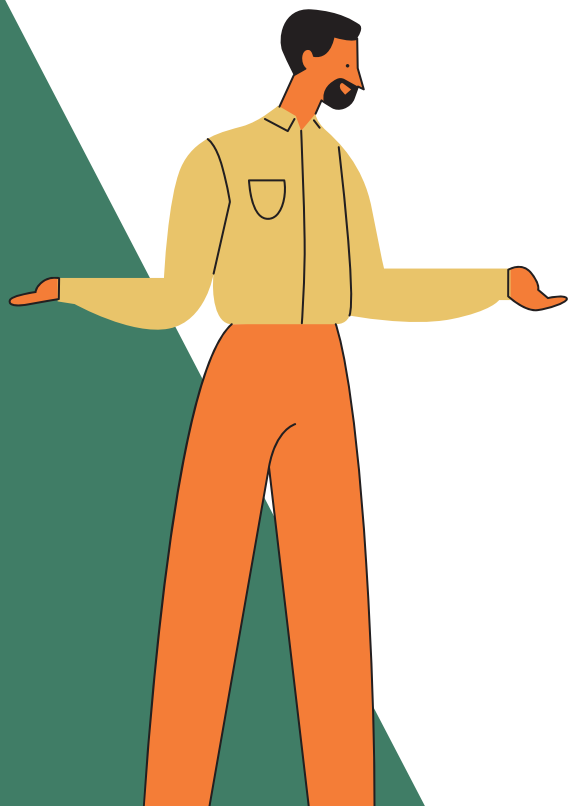
```
npm install  
packageName@version --no-save
```

# Một số APIs mới của React

startTransition : <https://github.com/reactwg/react-18/discussions/65>

<https://react.dev/reference/react/forwardRef>

<https://github.com/reactwg/react-18/discussions/37>







# Bài tập

Cấu hình import alias

Quản lý global state với Context  
API

