

Software Development

---

# React and Intermediate TypeScript

Module 13



# React and TypeScript



# React and TypeScript

01

At this point in the course, we've separately covered the React and TypeScript concepts.

02

In this module, we'll introduce additional TypeScript features and discuss how React and TypeScript fit together.



How do TypeScript  
**types** and **interfaces**  
compare?



# TypeScript Types and Interfaces

- 1 TypeScript types and interfaces can both be used for type definition.
- 2 With many uses cases, either types or interfaces can be used as the implementation for a particular situation.
- 3 In this module, we'll discuss some of the minor differences between types and interfaces, and introduce best practices accordingly.



What are TypeScript  
**generics?**



# TypeScript Generics

- 1 TypeScript generics provide a powerful mechanism for reusing code in a variety of contexts.
- 2 With generics, we can provide one common implementation to support a wide variety of supported variable types.
- 3 Generics are often helpful when using TypeScript in combination with React.

# TypeScript React Properties and State Management

- 1 We can effectively implement TypeScript features in the context of React properties and state management.
- 2 Throughout, we make use of TypeScript generics, interfaces, and types to do so.





What is the React  
**useContext** hook?



# The useContext Hook

- 1 The `useContext` hook provides a context that can be leveraged throughout our application.
- 2 Unlike with React properties, it isn't necessary to pass `useContext` hook properties to each component and subcomponent.
- 3 As with other React code, we can use TypeScript for `useContext` hook implementation.



# Instructor **Demonstration**

Mini-Project



**Questions?**





**The End**