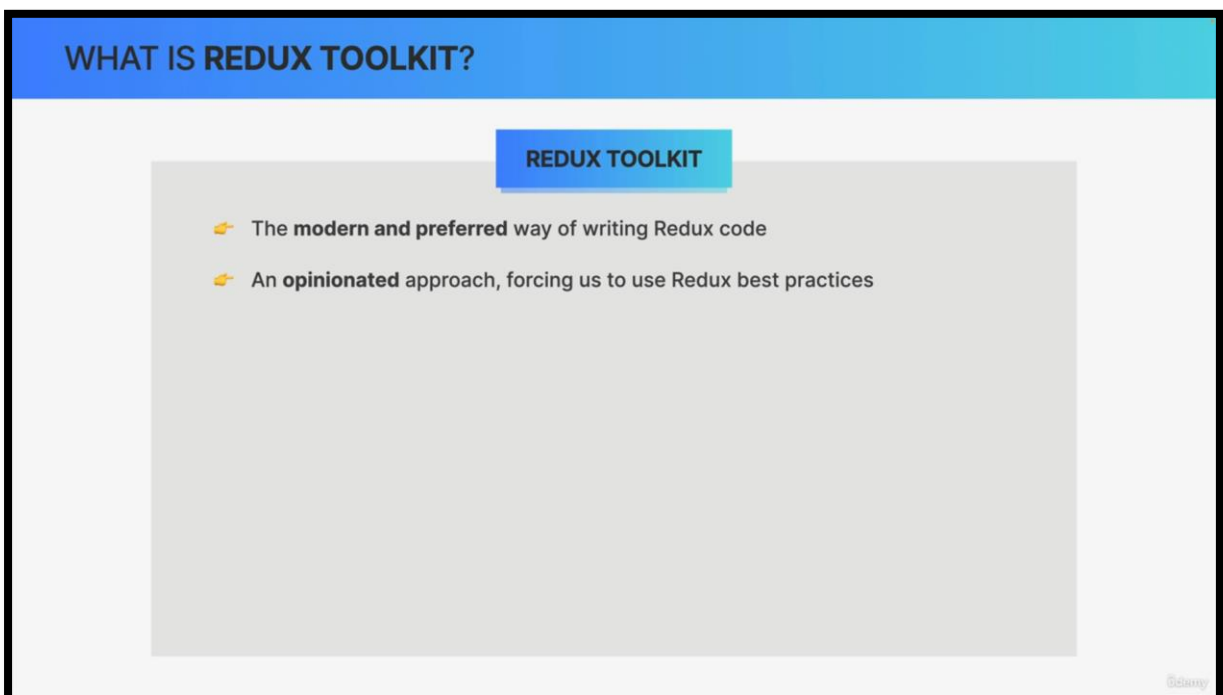


Redux Toolkit is the modern way of writing Redux code, and it is advised by the Redux team to now prefer Redux Toolkit over classic Redux.



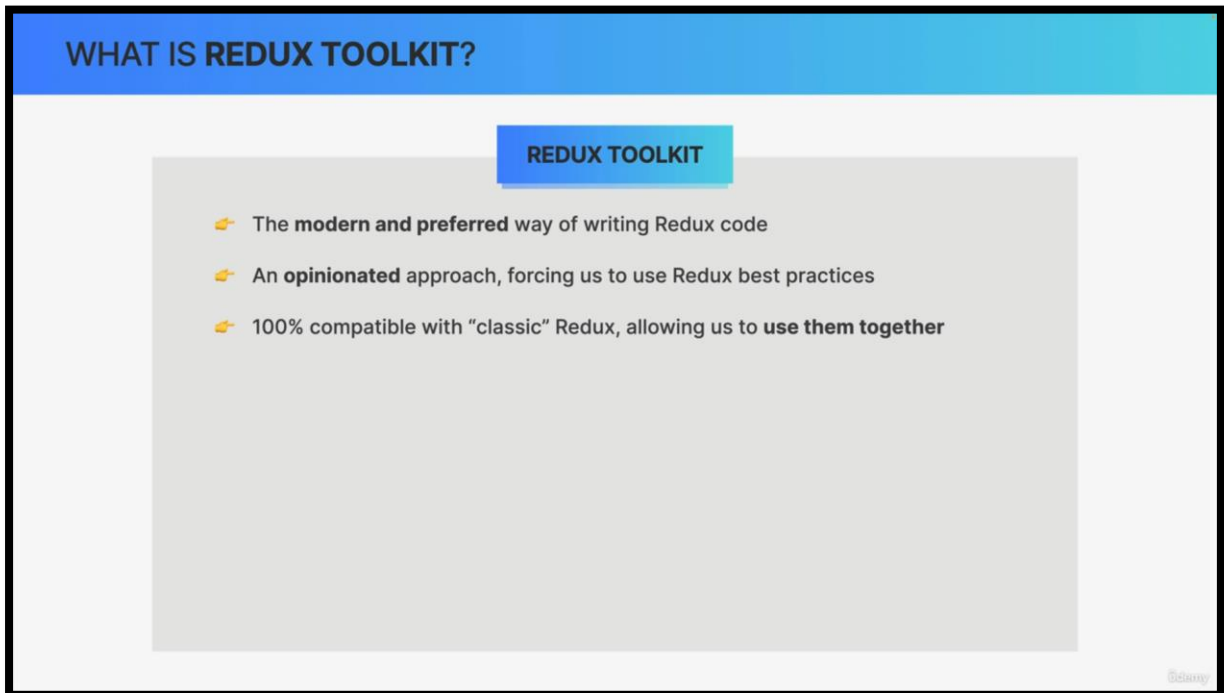
That's because Redux Toolkit is an opinionated approach that forces everyone to use best practices that the community has learned over the years.

So, basically, Redux Toolkit took all these best practices and placed them into this new way of writing Redux.

Now does this mean that everything that you just learned is obsolete?

Well, not at all.

First, without these fundamentals, it's very hard to understand Redux Toolkit because it is built on top of classic Redux.



Second, both ways of writing Redux are 100% compatible with each other.

So, you can actually use classic Redux in one part of the app and Redux Toolkit in another part if that's what you want to do.

What are the actual advantages of Redux Toolkit?

WHAT IS REDUX TOOLKIT?

REDUX TOOLKIT

- 👉 The **modern and preferred** way of writing Redux code
- 👉 An **opinionated** approach, forcing us to use Redux best practices
- 👉 100% compatible with "classic" Redux, allowing us to **use them together**
- 👉 Allows us to write **a lot less code** to achieve the same result (less "boilerplate")

Scrimba

The biggest advantage is certainly that Redux Toolkit allows us to write a lot less code to achieve the exact same result as before.

So, we say that we need a lot less boilerplate code, which is basically code that only sets things up but doesn't really do anything meaningful and so Redux Toolkit hides all that stuff, like setting up middleware and the developer tools.

WHAT IS REDUX TOOLKIT?

REDUX TOOLKIT

- 👉 The **modern and preferred** way of writing Redux code
- 👉 An **opinionated** approach, forcing us to use Redux best practices
- 👉 100% compatible with "classic" Redux, allowing us to **use them together**
- 👉 Allows us to write **a lot less code** to achieve the same result (less "boilerplate")
- 👉 Gives us 3 big things (but there are many more...):

Scrimba

So out of the box Redux Toolkit gives us many things but the three highlights are:

WHAT IS REDUX TOOLKIT?

REDUX TOOLKIT

- 👉 The **modern and preferred** way of writing Redux code
- 👉 An **opinionated** approach, forcing us to use Redux best practices
- 👉 100% compatible with "classic" Redux, allowing us to **use them together**
- 👉 Allows us to write **a lot less code** to achieve the same result (less "boilerplate")
- 👉 Gives us 3 big things (but there are many more...):
 - 1 We can write code that **"mutates"** state inside reducers (will be converted to **immutable** logic behind the scenes by "Immer" library)

Scrimba

First, it allows us to write code that actually mutates state inside reducers. Now behind the scenes a library called Immer will convert our code back to non-mutating,

So, like the one that we have been writing up until this point, but the new code that we're gonna write will look as if we were mutating the state object directly.

This is really the biggest advantage of using Redux Toolkit, because if you have a pretty complex state object with nested objects and arrays, this can actually be a game-changer in reducing complexity.

WHAT IS REDUX TOOLKIT?

REDUX TOOLKIT

- 👉 The **modern and preferred** way of writing Redux code
- 👉 An **opinionated** approach, forcing us to use Redux best practices
- 👉 100% compatible with "classic" Redux, allowing us to **use them together**
- 👉 Allows us to write **a lot less code** to achieve the same result (less "boilerplate")
- 👉 Gives us 3 big things (but there are many more...):
 - 1 We can write code that **"mutates"** state inside reducers (will be converted to **immutable** logic behind the scenes by "Immer" library)
 - 2 Action creators are **automatically** created

Scrimba

Redux Toolkit also automatically creates action creators from our reducers, and this can be quite helpful as, again, it helps us writing less code.

It can, however, also create some additional work in some situations as we will experience in practice soon.

WHAT IS REDUX TOOLKIT?

REDUX TOOLKIT

- 👉 The **modern and preferred** way of writing Redux code
- 👉 An **opinionated** approach, forcing us to use Redux best practices
- 👉 100% compatible with "classic" Redux, allowing us to **use them together**
- 👉 Allows us to write **a lot less code** to achieve the same result (less "boilerplate")
- 👉 Gives us 3 big things (but there are many more...):
 - 1 We can write code that **"mutates"** state inside reducers (will be converted to **immutable** logic behind the scenes by "Immer" library)
 - 2 Action creators are **automatically** created
 - 3 **Automatic** setup of thunk middleware and DevTools

Scrimba

Finally, Redux Toolkit will also automatically set up Thunk middleware and the developer tools so that we can focus on writing code that actually does something in the application.