



SUNIL VISHWAKARMA
@linkinsunil

JS

JavaScript

Ninja

Techniques

👉 nullish coalescing assignment operator

👉 nullish coalescing operator

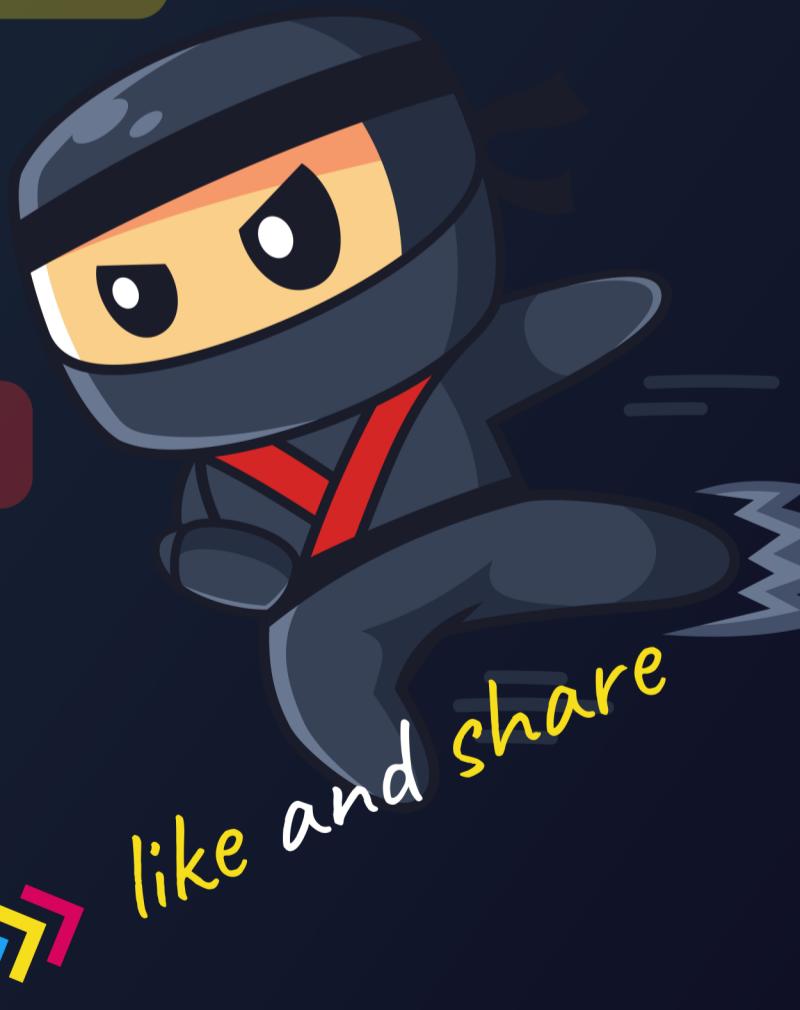
👉 spread operator

👉 optional chaining

👉 destructuring assignment

👉 dynamic property name

swipe →





nullish coalescing assignment operator

instead of this

```
● ● ●  
  
if (token === null || token === undefined) {  
  token = user;  
}
```

do this

```
● ● ●
```

```
token ??= user
```

swipe →



nullish coalescing operator

instead of this

```
if (token === null || token === undefined) {  
    user = "unknown";  
} else {  
    user = token;  
}
```

do this

```
user = val ?? "unknown";
```

swipe →



spread operator

instead of this



```
const combined = Object.assign({}, obj1, obj2)
```

do this



```
const combined = { ...obj1, ...obj2};
```

swipe →



optional chaining

instead of this



```
if (user) {  
  if(user.contact) {  
    return user.contact.email;  
  }  
}
```

do this



```
return user?.contact?.email;
```

swipe →



destructuring assignment

instead of this



```
for(const key in arr) {  
  const val = arr[key];  
  console.log(key, val)  
}
```

do this



```
for(const [key, val] of arr.entries()) {  
  console.log(key, val)  
}
```

swipe →



dynamic property name

instead of this



```
const obj = {};  
obj[prop] = value;
```

do this



```
const obj = {[prop] = value};
```

swipe →

Thats a Wrap!

If you liked it, visit my profile and checkout for other
short and easy explanations

Context API vs Redux-Toolkit

Feature ▾	Context API ▾	Redux-Toolkit ▾
State Management	Not a full-fledged state management tool. Passes down values and update functions, but does not have built-in ability to store, get, update, and notify changes in values.	A full-fledged state management tool with built-in ability to store, get, update, and notify changes in values.
Usage	Best for passing static or infrequently updated values and moderately complex state that does not cause performance issues when passed using props.	Best for managing large-scale, complex state that requires asynchronous actions and side-effects.
Code Complexity	Minimal setup and low learning curve. However, can become complex when used with a large number of components and nested Contexts.	
Performance	Can cause unnecessary re-renders if the state passed down is not simple and can require the use of additional memoization techniques to optimize performance.	
Developer Tools	Does not come with pre-built developer tools but can be used with third-party tools like React DevTools.	
Community	Has a large and active community.	

React

Virtual DOM

useRef()
referencing values in React

When you want a component to remember some information, but you don't want that information to trigger new renders, you can use a ref.

Lets See into

1. How to add a ref to component?
2. How to update a ref's value?
3. How refs are different from state?
4. When to use refs?
5. Best practices for using refs?

{ Current }

⚠ Please Like & Share for no reason

VS

TF is a Virtual DOM?

real DOM

swipe →

JavaScript Evolution

ES6 ES2015

1. let and const
2. Arrow functions
3. Default parameters
4. Rest and spread operators
5. Template literals
6. Destructuring assignment
7. Classes and inheritance
8. Promises for asynchronous programming
9. Symbols for creating unique object keys
10. Iterators and generators

ES9 ES2018

1. Object.getOwnPropertyDescriptors()
2. Spread syntax for objects
3. Promise.prototype.finally()

ES10 ES2019

1. Array.prototype.flat()
2. Array.prototype.flatMap()
3. String.prototype.trimStart()
4. String.prototype.trimEnd()
5. Array.prototype.sort() (stable)

ES11 ES2020

1. BigInt
2. Nullish coalescing operator (??)
3. Optional chaining operator (?)
4. Promise.allSettled()

ES12 ES2021

1. String.prototype.replaceAll()
2. Logical assignment operators (|=, &=&, ??=)

ES13 ES2022

1. Array.prototype.lastIndexOf()
2. Object.hasOwn()
3. at() for strings and arrays
4. Top level await()

share

Redux Toolkit
Easiest Explanation Ever

React Redux Toolkit

swipe →

like and share

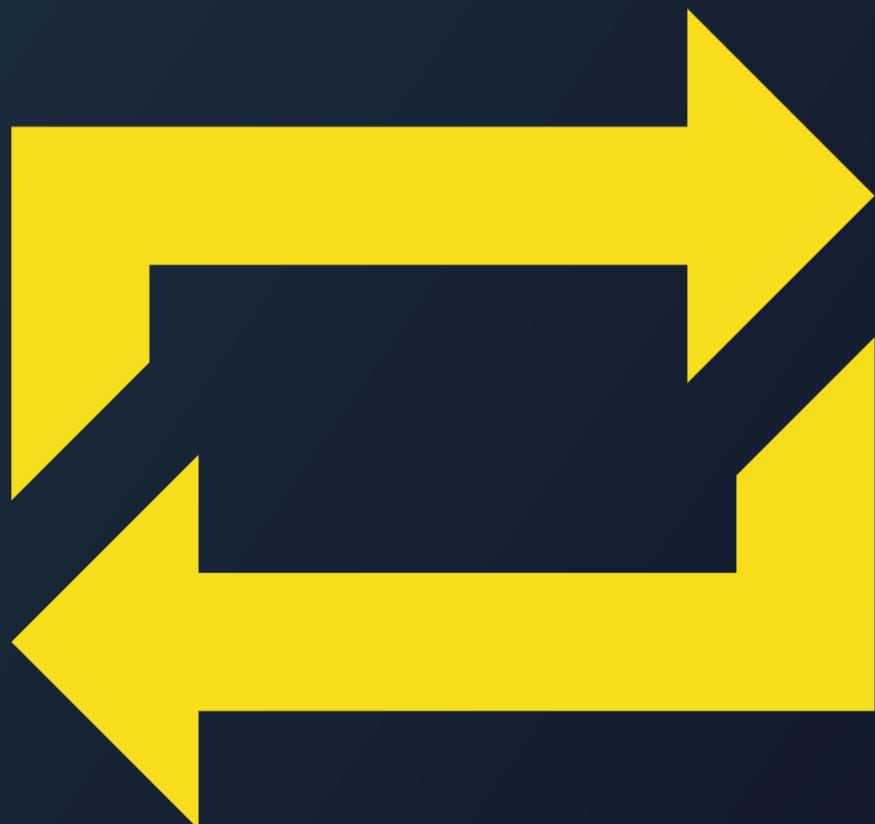
Instagram icon @linkinsunil LinkedIn icon @linkinsunil Twitter icon @officialskv



SUNIL VISHWAKARMA
@linkinsunil

P.S.

**Repost this if you
think your followers
will like it**





SUNIL VISHWAKARMA
@linkinsunil

Enjoyed this?

1. Follow me
2. Click the notification
3. Never miss a post