



SUNIL VISHWAKARMA  
@linkinsunil

# # JavaScript Ninja Techniques

👉 nullish coalescing assignment operator

👉 nullish coalescing operator

👉 spread operator

👉 optional chaining

👉 destructuring assignment

👉 dynamic property name



like and share

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 →



**SUNIL VISHWAKARMA**  
@linkinsunil


# 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
<b>State Management</b>	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.
<b>Usage</b>	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.
<b>Code Complexity</b>	Minimal setup and low learning curve. However, can become complex when used with a large number of components and nested Contexts.	
<b>Performance</b>	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.	
<b>Developer Tools</b>	Does not come with pre-built developer tools but can be used with third-party tools like React Developer Tools.	
<b>Community</b>	Has a large and active community	



## JavaScript Evolution

**ES6 ES2015**

- let and const
- Arrow functions
- Default parameters
- Rest and spread operators
- Template literals
- Destructuring assignment
- Classes and inheritance
- Promises for asynchronous programming
- Symbols for creating unique object keys
- Iterators and generators

**ES9 ES2018**

- Object.getPrototypeOf()
- Spread syntax for objects
- Promise.prototype.finally()

**ES10 ES2019**

- Array.prototype.flat()
- Array.prototype.flatMap()
- String.prototype.trimStart()
- String.prototype.trimEnd()
- Array.prototype.sort() (stable)

**ES11 ES2020**


- BigInt
- Nullish coalescing operator (??)
- Optional chaining operator (?.)
- Promise.allSettled()

**ES12 ES2021**


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

**ES13 ES2022**


- Array.prototype.lastIndexOf()
- Object.hasOwn()
- at() for strings and arrays
- Top level await()



## 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

- How to add a ref to component?
- How to update a ref's value?
- How refs are different from state?
- When to use refs?
- Best practices for using refs?




**Current**

⚠ Please Like & Share for no reason


VS

TF is a real DOM?





## Real DOM

swipe →



## Redux Toolkit

Easiest Explanation Ever

React Redux Toolkit

swipe →

like and share

@linkinsunil @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