SUNIL VISHWAKARMA
@linkinsunil

Ways to Create Objects in JavaScript

👉Object constructor

👉Object's create method

👉Object literal syntax

👉Function constructor

👉Function constructor with prototype

👉ES6 Class syntax

👉Singleton pattern

save it, like and share

swipe ⟶


SUNIL VISHWAKARMA
@linkinsunil

Object Constructor

The simplest way to create an empty object is using the Object constructor. Currently this approach is not recommended.

```
var object = new Object();
```

swipe ⟶


SUNIL VISHWAKARMA
@linkinsunil

Object's Create Method

The create method of Object creates a new object by passing the prototype object as a parameter

```
var object = Object.create(null);
```

swipe ⟶

SUNIL VISHWAKARMA
@linkinsunil

Object Literal Syntax

The object literal syntax (or object initializer), is a comma-separated set of name-value pairs wrapped in curly braces.

```
var object = {
      name: "Sunil",
      age: 25
};

//Note:
Object literal property values can be
of any data type, including array,
function, and nested object.
```

Note: This is the easiest way to create an object

swipe ⟶

SUNIL VISHWAKARMA
@linkinsunil

Function Constructor

Create any function and apply the new operator to create object instances.

```
function Person(name) {
  this.name = name;
  this.age = 25;
}
var object = new Person("Sunil");
```

swipe ⟶

SUNIL VISHWAKARMA
@linkinsunil

Function Constructor With Prototype

This is similar to function constructor but it uses prototype for their properties and methods.

```
function Person() {}
Person.prototype.name = "Sunil";
var object = new Person();
```

This is equivalent to an instance created with an object create method with a function prototype and then call that function with an instance and parameters as arguments.

```
function func() {};
new func(x, y, z);
```

OR

swipe ⟶

SUNIL VISHWAKARMA
@linkinsunil

...continued

```
// Create a new instance using function prototype.
var newInstance = Object.create(func.prototype)

// Call the function
var result = func.call(newInstance, x, y, z),

// If the result is a non-null object then use it
// otherwise just use the new instance.
console.log(result && typeof result === 'object'
  ? result : newInstance);
```

swipe ⟶

SUNIL VISHWAKARMA
@linkinsunil

ES6 Class Syntax

ES6 introduces class feature to create the objects

```
class Person {
  constructor(name) {
    this.name = name;
  }
}

var object = new Person("Sunil");
```

swipe ⟶

SUNIL VISHWAKARMA
@linkinsunil

Singleton pattern

A Singleton is an object which can only be instantiated one time. Repeated calls to its constructor return the same instance and this way one can ensure that they don't accidentally create multiple instances.

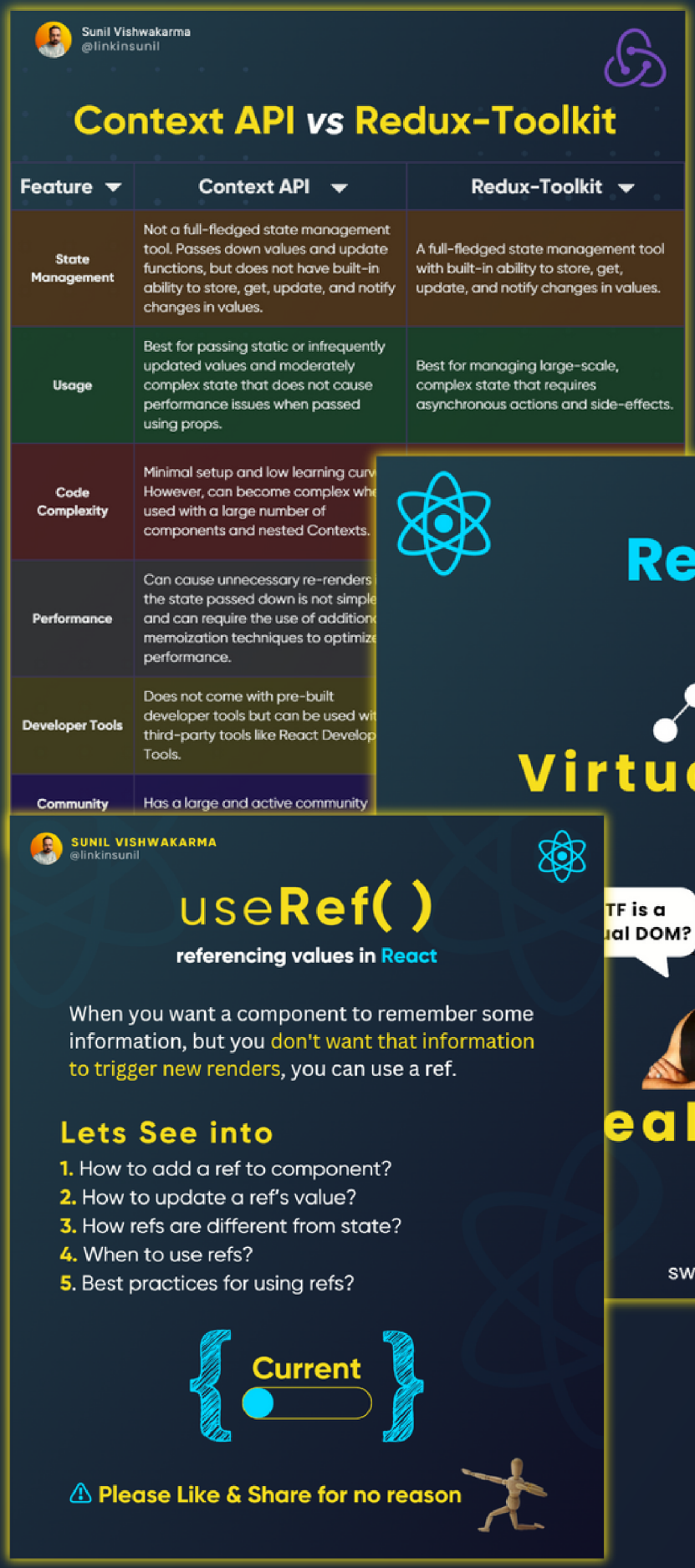
```
var object = new (function () {
  this.name = "Sunil";
})();
```

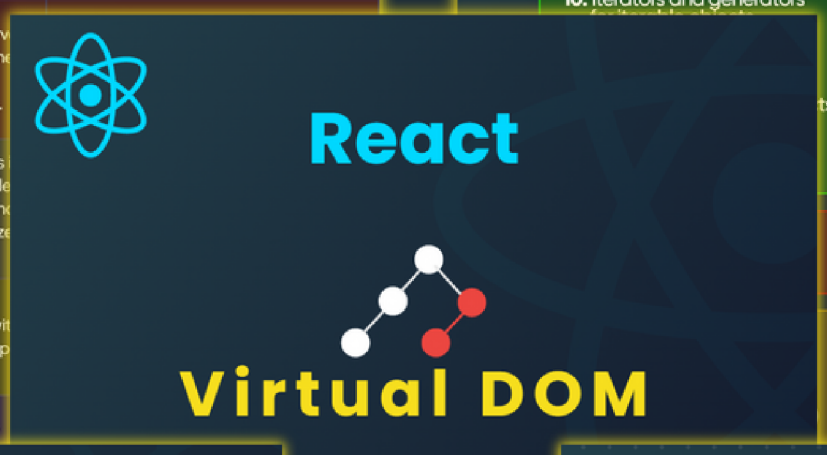
swipe ⟶

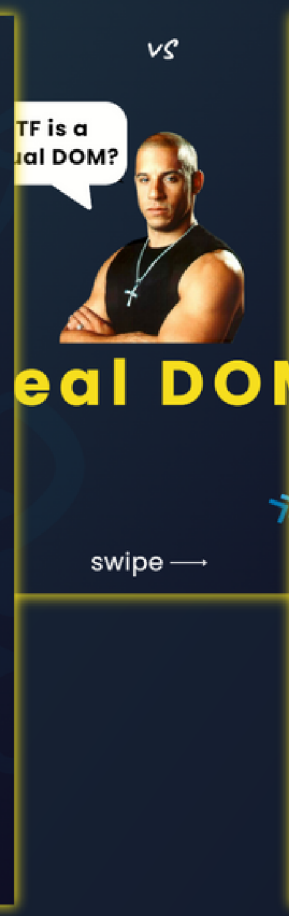
SUNIL VISHWAKARMA
@linkinsunil

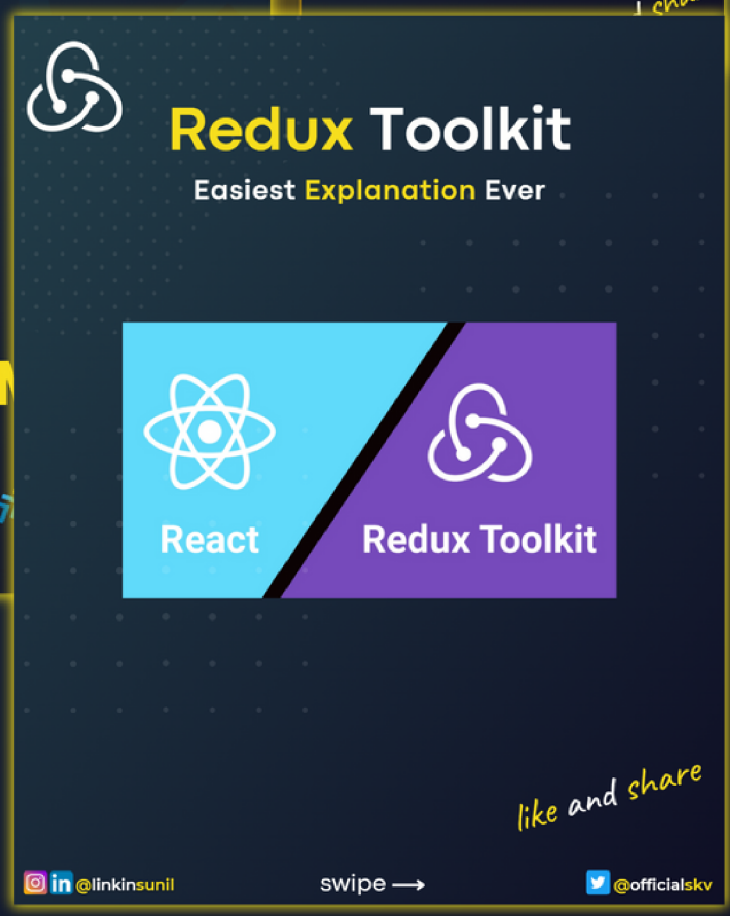
Thats a Wrap!

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









P.S.

Repost this if you think your followers will like it

Enjoyed this?

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