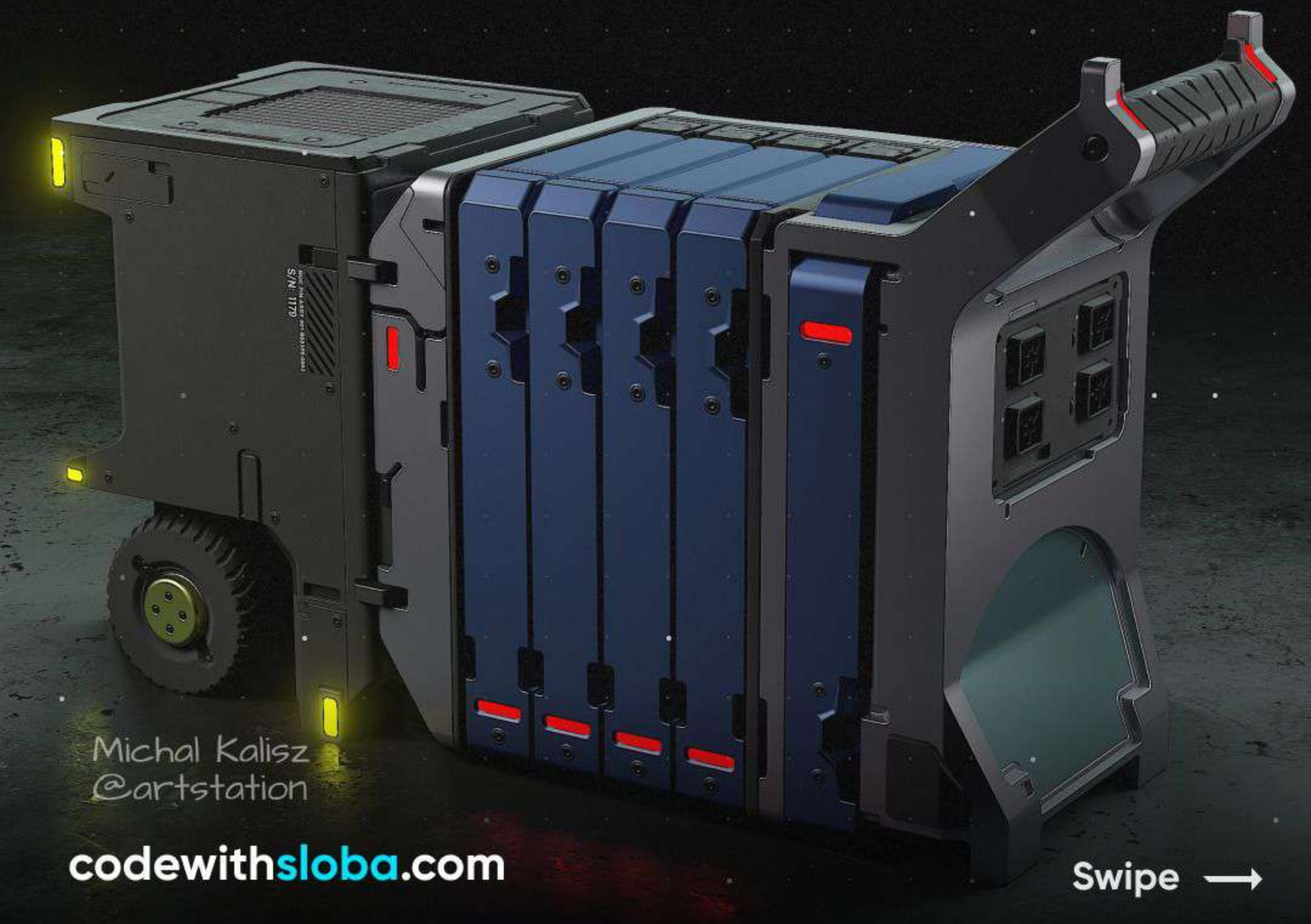
JS

like and share

Generator Functions

in JavaScript



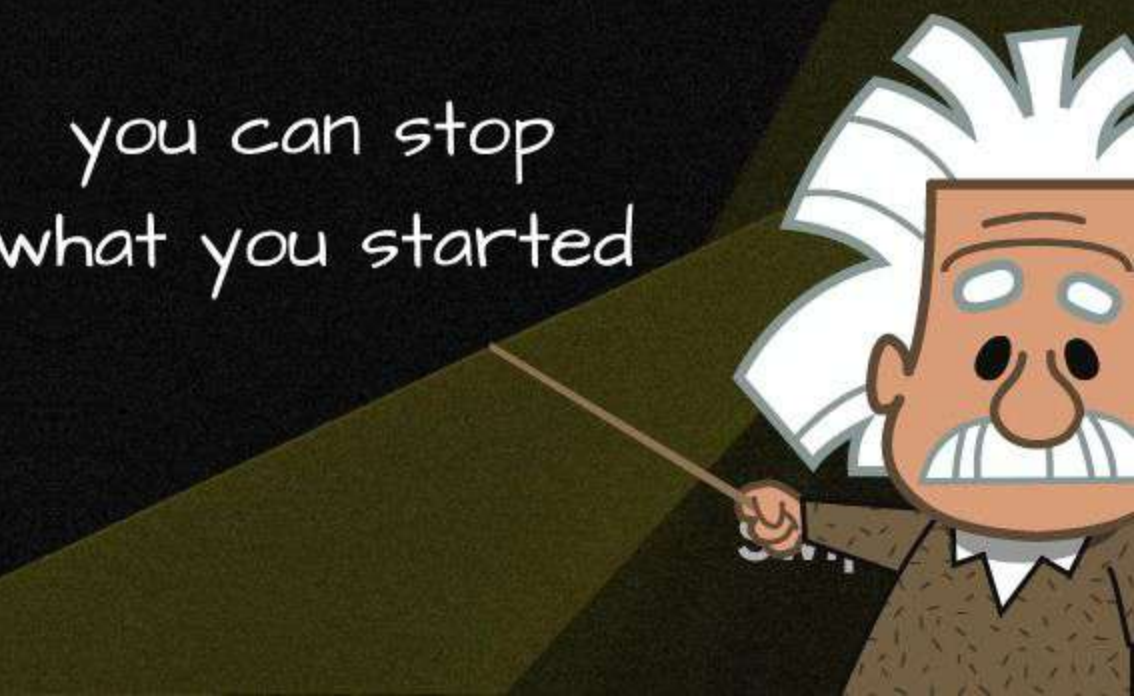
Michal Kalisz
@artstation

codewithsloba.com

Swipe →

A generator function is a special type of function that can be **paused** and **resumed**.

This makes it useful for creating iterators, which are objects that can be used to **iterate over a sequence of values**.



Generator functions are defined using the `function*()` syntax.

The * character tells the JavaScript interpreter that the function is a generator function.

```
function* fibonacci() {
  let a = 0;
  let b = 1;

  while (true) {
    yield a;
    a = b;
    b = a + b;
  }
}
```

yield keyword is used to pause the function and return a value.

Swipe →

The function can then be **resumed** by calling the **next()** method on the **iterator object**.

```
const fib = fibonacci();

console.log(fib.next()); // 0
console.log(fib.next()); // 1
console.log(fib.next()); // 2
console.log(fib.next()); // 4
```

```
{
  "value": 0,
  "done": false
}
```

```
{
  "value": 1,
  "done": false
}
```

```
{
  "value": 2,
  "done": false
}
```

```
{
  "value": 4,
  "done": false
}
```

Swipe →

The `next()` method returns an object with two properties: **value** and **done**.

The value property contains the **value that was yielded** by the function.

The done property is a **boolean value** that indicates whether the function has **finished generating** values.

Swipe →

codewithsloba.com

Get a weekly digest of my tips and tutorials by subscribing now.

Code with sloba

Educator • Senior JavaScript Developer • YouTuber