

**JS**

# GENERATORS IN JS

# Generators

Generators are functions that can be paused and resumed

- Introduced in ECMAScript 6 (ES6)
- Generator function defined using the **function\*** syntax.
- You can't create generator function in arrow function

# Yield Keyword

What is yield Oxford dictionary?

[intransitive] to stop resisting something or someone; to agree to do something that you do not want to do

Other terms:

produce or generate (a result, gain, or financial return).

"this method yields the same results"

- The yield keyword is used inside a generator function to *pause its execution* and *produce a value to the caller*.
- When a generator encounters a yield statement, it returns *an object with the yielded value and a done property* indicating whether the generator has completed.

## Next method

- The next() method is used to resume the execution of a generator from where it was paused by the last yield statement.
- It can also be used to send a value to the generator.

## return() method

- The return method is used to force the generator to complete and return a specified value.
- It essentially acts as if a return statement was encountered in the generator function.

## throw() method

The throw method is used to throw an exception into the generator.

# Generators as Iterator

An iterator in JavaScript is an object that provides a way to access the elements of a **collection or sequence, one at a time**, in a specific order.

It follows the iterator protocol, which includes having a **next()** method.

The next() method returns an object with two properties:

**value**: The current value in the iteration.

**done**: A boolean indicating whether the end of the iteration has been reached (true if it has, false otherwise).

## Examples

- Arrays: for...of loop or iterable methods.
- Strings: Iterable character by character.
- Maps: Iterate over key-value pairs.
- Sets: Iterate over unique elements.
- Symbol.iterator
- **We can use Generator Functions as iterators**

# USE CASES

1. Can be used to Infinite sequences
2. Can be used as Iterator
3. Lazy Loading: Generators can help with lazy loading resources. You can load resources one at a time as needed, improving performance by avoiding unnecessary loading.



Thanks