

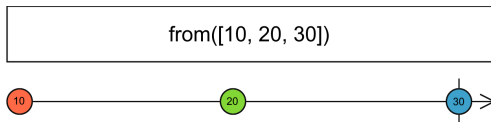
from

Crea un Observable a partir de un Array, un objeto similar a un Array, una Promesa, un objeto iterable o un objeto similar a un Observable

► Signatura

Descripción

Convierte prácticamente cualquier elemento en un Observable.



`from` convierte varios tipos de datos u objetos en Observables. También puede convertir una Promesa, un objeto similar a un Array o un objeto iterable en un Observable que emite los elementos de dicha Promesa, Array o iterable. Un String, en este contexto, se interpreta como un array de caracteres. Los objetos similares a Observables (contienen una función nombrada con el Símbolo ES2015 que corresponde a Observable) también se puede transformar mediante este operador.

Ejemplos

Crear un Observable a partir de una cadena

[StackBlitz](#)

```
import { from } from "rxjs";

const letter$ = from("RxJS mola");

letter$.subscribe(console.log);
// Salida: 'R', 'x', 'J', 'S', ' ', 'm', 'o', 'l', 'a'
```

Copy

Crear un Observable a partir de un Array de cadenas

[StackBlitz](#)

```
import { from } from "rxjs";

const fruit$ = from(["Fresa", "Cereza", "Mora"]);

fruit$.subscribe((fruit) => console.log(fruit));
// Salida: Fresa, Cereza, Mora
```

Copy

Crear un Observable a partir de un Map

[StackBlitz](#)

```
import { from } from "rxjs";

const language$ = from(
  new Map([
    ["Java", "Orientado a objetos"],
    ["Ruby", "Multiparadigma"],
    ["Haskell", "Funcional"],
  ])
);

language$.subscribe(console.log);
// Salida: ["Java", "Orientado a objetos"], ["Ruby", "Multiparadigma"], ["Haskell", "Funcional"]
```

Copy

Crear un Observable a partir de una promesa

[StackBlitz](#)

```
import { from } from "rxjs";

const promise$ = from(Promise.resolve("Prometo empezar a aprender RxJS"));

promise$.subscribe(console.log);
// Salida: 'Prometo empezar a aprender RxJS'
```

Copy

Crear un Observable a partir de un NodeList

[StackBlitz](#)

```
import { from } from "rxjs";

const node$ = from(document.querySelectorAll("p"));

node$.subscribe((node) => console.log(node));
// Salida: HTMLParagraphElement {tagName: "p", attributes: {...}}
```

Copy

Ejemplos de la documentación oficial

Convertir un array a un Observable

```
import { from } from "rxjs";

const array = [10, 20, 30];
const result = from(array);

result.subscribe((x) => console.log(x));

// Salida:
// 10
// 20
// 30
```

Copy

Convertir un iterable infinito (a partir de un generador) en un Observable

Copy

```
import { from } from "rxjs";
import { take } from "rxjs/operators";

function* generateDoubles(seed) {
  let i = seed;
  while (true) {
    yield i;
    i = 2 * i; // dóblalo
  }
}

const iterator = generateDoubles(3);
const result = from(iterator).pipe(take(10));

result.subscribe((x) => console.log(x));

// Salida:
// 3
// 6
// 12
// 24
// 48
// 96
// 192
// 384
// 768
// 1536
```

Con el planificador asyncScheduler