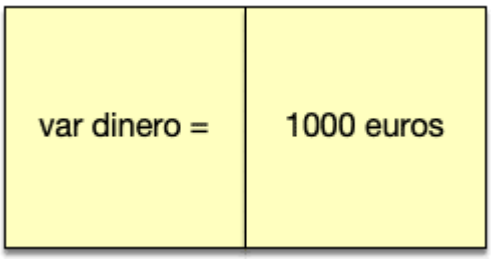
Tabla de Contenidos

- El concepto de Variable
- El concepto de Promesa y la programación asíncrona
- Javascript Observable y la programación asíncrona
- Otros artículos relacionados

¿Que es un JavaScript Observable?. Este es una pregunta muy habitual hoy en día . Los Observables son conceptos que cuesta realmente entender en profundidad ya que están ligados a la programación asíncrona y relacionados con el concepto de Promesas. Para entender correctamente los Observables hay que entender de forma correcta como funcionan los conceptos de programación asíncrona a alto nivel. Vamos a explicarlos:

El concepto de Variable

Imaginemonos que nosotros queremos irnos de viaje y el viaje nos cuesta 1000 euros . ¿Podemos o no podemos irnos de viaje? . Es una pregunta sencilla y todo dependerá de si tenemos el dinero disponible en nuestra cuenta corriente . En caso afirmativo nos podemos ir de viaje ya que tenemos guardados los 1000 euros necesarios. Este concepto aunque no se vea con claridad es el concepto de variable (un valor que se almacena y que puede variar) . Hoy podemos tener 1000 euros y mañana 900 o 1100.



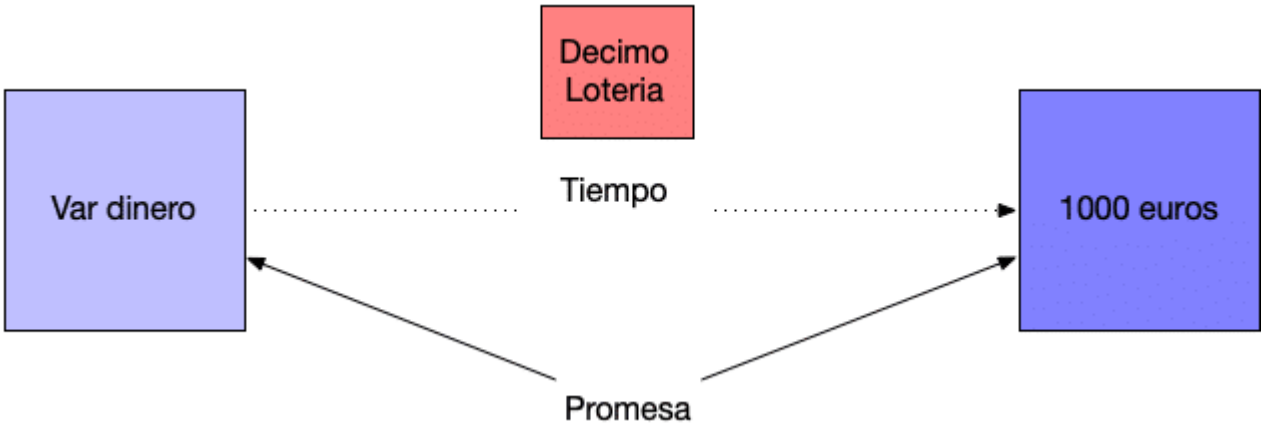
En código sería :

```
var numero=5;
console.log(numero);
```

Este mensaje saldrá por la consola de forma inmediata.

El concepto de Promesa y la programación asíncrona

El concepto de Promesa es algo más complejo , pero vamos a intentarlo explicar con los 1000 euros y el viaje. Supongamos que seguimos queriendo irnos de viaje y no disponemos del dinero. La realidad es que no podremos irnos de viaje. Sin embargo tenemos un décimo de lotería que ha sido premiado con 1000 euros . ¿Entonces podremos viajar? . La realidad es que sí pero no …. tendremos que ir a la administración de lotería a confirmar el premio y la administración tardará unos días en procesarlo y darnos el dinero . Esto es lo que habitualmente se conoce como una Promesa en programación es una variable que en el momento actual no tiene valor pero en un futuro lo tendrá y podremos usarle.



Prácticamente todas las llamadas AJAX son Promesas , veamos un ejemplo en código:

```
var promesa= new Promise (function(resolve,reject) {

  setTimeout(function() {
```

```
    resolve(1000);

  },2000);


})

promesa.then(function(resultado) {

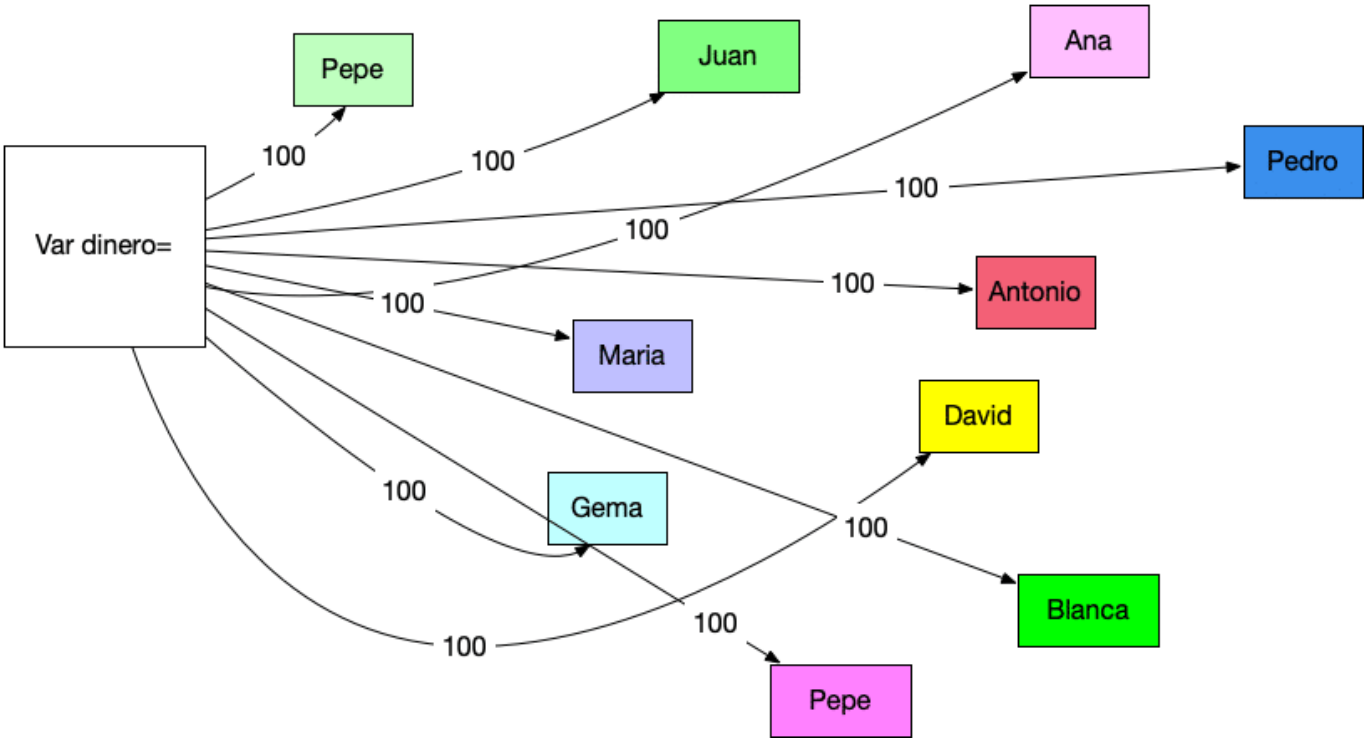
  console.log(resultado);

})
```

Este mensaje saldrá por la consola al cabo de 2 segundos

Javascript Observable y la programación asíncrona

Supongamos ahora que nos queremos ir de viaje , no tenemos el dinero y no nos ha tocado la lotería . Hemos tenido bastante mala suerte, pero todavía nos queda una opción podemos recurrir a nuestros amigos y que cada uno de ellos nos preste 100 euros para irnos de viaje .



Tendremos que hacer 10 llamadas de teléfono y esperar pacientemente a quedar con con cada persona para que nos de los 100 euros cuando hayamos quedado con todos nos encontraremos que finalmente podemos viajar. Ese es el concepto de JavaScript Observable , hace referencia a la combinación de multiples peticiones asíncronas que pueden terminar en diferentes momentos temporales . Según vayamos recibiendo el dinero podremos ir pagando el viaje. Veámoslo en código:

```
<html>
<head>
<script type="text/javascript"
src="https://unpkg.com/rxjs/bundles/rxjs.umd.min.js">
</script>
<script type="text/javascript" src="eventos.js">
</script>
</head>
<body>
```

```
<input type="button" value="pulsar" id="miboton"/>
</body>
</html>
```

En estos momentos simplemente tenemos una página html con un botón vamos a añadir un observable

```
window.onload=function() {

  var boton = document.getElementById('miboton');

  var pulsaciones = rxjs.fromEvent(boton, 'click');


    pulsaciones.subscribe(e => {
      console.log(100);

    });

}
```

Ahora cada vez que pulsamos el botón nos llegan 100 euros El observable esta ligado al botón y cada pulsación del botón es un elemento asíncrono que se ejecuta . En este sentido es idéntico a la petición de 100 euros que hacemos a nuestros 10 amigos . Los observables son conceptos fundamentales que aprender en nuestro día a día .

Otros artículos relacionados

1. Fetch API Promesas Ajax y conceptos claves
2. Curso Ajax JSON Promesas y buenas prácticas
3. jQuery Promise y AJAX
4. JavaScript Streams vs Promises