

Handling Multiple Promises

using the `promise.all()` function to handle multiple promises

`Promise.all()` takes an iterable object of promises. In this section, we will use arrays. Once all of them are fulfilled, it returns an array of fulfilled values. If any of the promises in the array fails, `Promise.all()` also fails. Study the following code:

```
var loan1 = new Promise( (resolve, reject) => {
  setTimeout( () => resolve( 110 ) , 1000 );
});
var loan2 = new Promise((resolve, reject) => {
  setTimeout( () => resolve( 120 ) , 2000 );
});
var loan3 = new Promise( (resolve, reject) => {
  reject( 'Bankrupt' );
});

Promise.all([ loan1, loan2, loan3 ]).then( value => {
  console.log(value);
}, reason => {
  console.log(reason);
} );
```



Now, let's solve some exercises.