PROMISE AND PROMISES.ALL

GENERAL EXAMPLE BEFORE UNDERSTANDING PROMISES

- Its your girlfriend birthday and you went to pizza shop to give an order of 50 pizzas for evening party.

- Its a promise and you passed it to pizzashop(pending)
- Here two scenario's occur

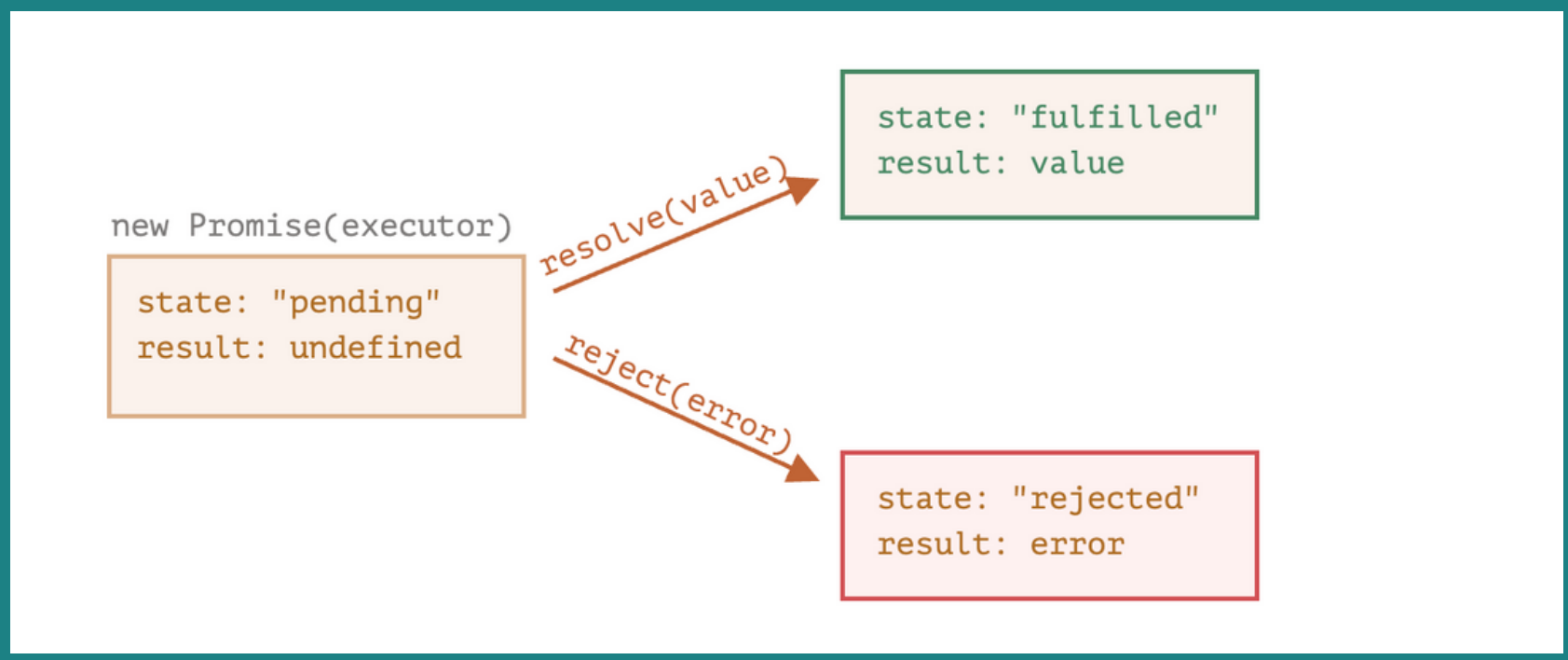
  - Pizza shop delivered 50 pizzas(fulfilment:resolve).
  - Pizza shop not delivered (rejected) due to high demand

PROMISE

In JavaScript, a promise is a good way to handle asynchronous operations. It is used to find out if the asynchronous operation is successfully completed or not.

A promise may have one of three states.

- Pending
- Fulfilled
- Rejected



- when you request data from the server by using a promise, it will be in a pending state.
- When the data arrives successfully, it will be in a fulfilled state.
- If an error occurs, then it will be in a rejected state.

```
let promise = new Promise(function(resolve, reject){
     //do something
});
```

we will have a promise example

```
let promise = new Promise(function (resolve, reject) {
    if (count) {
        resolve("There is a count value.");
    } else {
        reject("There is no count value");
    }
});
```

Promise is made and after a while promise will be fullfiled or rejected

Back to pizza shop examplei

- **We are calling to pizza shop at the evening to know the status .**

- **Here we are making a call to pizza shop to know output (resolve or rejected which is already know by pizza shop will be intimated)**

- In similar way of calling to pizza shop to know the status of resolve or reject ,the fulfilled promise has to be called with then and catch block to actually get the resolve and reject ouput

```
promise.then(
  (result) => {
     console.log(result);
  },
  (error) => {
     console.log(error);
  }
);
```

```
promise
  .then(success)
  .catch(error);
```

working on it is the only way to understand better

Promise.all()

The name its says all, which means when all promises which are passed into Promise.all() is fulfilled . Then promise is fullfilled.

NOTE: Even one promise is rejected, the promise.all will be rejected

```
const promise1 = Promise.resolve(3);
const promise2 = 42;
const promise3 = new Promise((resolve, reject) => {
  setTimeout(resolve, 100, 'foo');
});

Promise.all([promise1, promise2, promise3]).then((values) => {
  console.log(values);
});
// Expected output: Array [3, 42, "foo"]
```