ES6 - Promises

**Promises** are a clean way to implement async programming in JavaScript (ES6 new feature).

Prior to promises, Callbacks were used to implement async programming

## **Understanding Callback**

A function may be passed as a parameter to another function. This mechanism is termed as a **Callback**.

A Callback would be helpful in events.

<script>

function notifyAll(fnSms, fnEmail) {

console.log('starting notification process');

fnSms();

fnEmail();

}

notifyAll(function() {

console.log("Sms send ..");

},

function() {

console.log("email send ..");

});

console.log("End of script");

//executes last or blocked by other methods

</script>

**OUTPUT:**

starting notification process

Sms send ..

Email send ..

End of script

## **Understanding AsyncCallback**

Consider the above example.

To enable the script, execute an asynchronous or a non-blocking call to notifyAll() method. We shall use the **setTimeout()** method of JavaScript. This method is async by default.

The setTimeout() method takes two parameters −

* A callback function.
* The number of seconds after which the method will be called.

<script>

function notifyAll(fnSms, fnEmail) {

setTimeout(function() {

console.log('starting notification process');

fnSms();

fnEmail();

}, 2000);

}

notifyAll(function() {

console.log("Sms send ..");

},

function() {

console.log("email send ..");

});

console.log("End of script"); //executes first or not blocked by others

</script>