# Promises

ES6 came with many new features, but one of the best features was the official introduction of Promises. Promises allow you to write clean non-callback-centric code without ever having to worry about callback hell.

Even if you never write your own promise, knowing how they work is incredibly important, since many newer parts of the JavaScript API use promises instead of callbacks. Checkout [the full video](https://www.youtube.com/watch?v=DHvZLI7Db8E) to learn how to define and use promises.

Example 1 -

let userLeft = false;

let userMultiTaskingWhileWatchingVid = false;

function userOriented(errorCallback, callback) {

  if (userLeft) {

    callback({

      message: "User Left",

      emotion: ":/"

    });

  } else if (userMultiTaskingWhileWatchingVid) {

    errorCallback({

      message: "User was distracted",

      emotion: ":("

    });

  } else {

    callback({

      message: "Thumbs up and Subscribe"

    });

  }

}

userOriented(

  msg => {

    document.querySelector("#root").innerHTML = `${msg.message} ${msg.emotion}`;

  },

  msg => {

    document.querySelector("#root").innerHTML = `${msg.message}`;

  }

);

# writing above function using promises  
function userOriented() {

  return new Promise((solve, Unsolve) => {

    if (userLeft) {

      Unsolve({

        message: "User Left!"

      });

    } else if (userMultiTaskingWhileWatchingVid) {

      Unsolve({

        message: "User Multitasking!"

      });

    } else {

      solve({

        message: "Thumbs up and subscribe"

      });

    }

  });

}

userOriented()

  .then(msg => {

    document.querySelector("#root").innerHTML = msg.message;

  })

  .catch(msg => {

    document.querySelector("#root").innerHTML = msg.message;

  });

Example 2 -

const recordVideoOne = new Promise((Resolve, Reject) => {

  Resolve("Video one recorded!");

});

const recordVideoTwo = new Promise((Resolve, Reject) => {

  Resolve("Video two recorded!");

});

const recordVideoThree = new Promise((Resolve, Reject) => {

  Resolve("Video three recorded!");

});

Promise.all([recordVideoOne, recordVideoTwo, recordVideoThree]).then(message =>

  console.log(message)

);

# Froms

<https://scrimba.com/p/p7P5Hd/cW8Jdfy>

class Main extends React.Component {

    constructor() {

      super();

      this.state = {

        firstName: "",

        lastName: "",

      };

      this.handleChange = this.handleChange.bind(this);

    }

    handleChange(e) {

      this.setState({

        [e.target.name]: e.target.value

      });

    }

    render() {

      return (

        <form>

          <input

            type="text"

            name="firstName"

            placeholder="Type first name"

            onChange={this.handleChange}

          />

          <br/> <br/>

          <input

            type="text"

            name="lastName"

            placeholder="Type last name"

            onChange={this.handleChange}

          />

          <h3>{this.state.firstName} {this.state.lastName}</h3>

        </form>

      );

    }

}

ReactDOM.render(<Main />, document.getElementById('root'));