

TRY CATCH

```
function sayHello() {  
  try {  
    console.log('Hi Simran');  
  }  
  catch(error) {  
    console.log('Opps! Error!');  
  }  
}
```

execute all the code inside of this block

@codewithsimran

The error that occurred

If any error occurs inside try { } block, handle it inside the catch block

```
function sayHello() {  
  try {  
    console.log('Hi Simran')  
    throw new Error('Damn error!')  
    console.log('executed success')  
  }  
  catch(error) {  
    console.log('Error', error)  
  }  
}
```

@codewithsimran

> Hi Simran

Error Damn Error!

So it executes code inside of try {} block until it finds an error, and then moves to catch block. That's why executed success is not printed.

finally

@codewithSimran

```
function sayHello() {
```

```
  try {
```

```
    ;
```

```
  catch (error) {
```

```
    ;
```

```
  finally {
```

```
    console.log("Always printed")
```

```
  }
```

```
  console.log("outside try catch")
```

```
}
```

→ No matter what part of try and catch gets executed, whatever code we write inside finally will always get executed

↳ This piece of code will never get executed. (outside try catch)

- ★ Try catch can be used to handle synchronous code
- ★ You can have nested try catch blocks