Errors in JavaScript

There is a native JavaScript constructor Error in JavaScript.

Go to console and type

> Error

f Error() { [native code] }

@codewithsimran

So you can create an Error with this constructor function as follows

new Error('any message')

↳ new error instance

But this dosent mean that we have an actual error, we need to actually throw an error

> throw new Error()

Uncaught Error → now we have an actual error

@codewithSimran

And when we throw something, the execution of our program stops.

```
console.log('1')
throw new Error()
console.log('2')
```

@codewithsimran

```
> 1
> Uncaught f Error()
```

2 is not even printed because the execution of our program will stop

When we create an error, we can access some properties on it

```
const someError = new Error('Damn Error!')
> someError.name
    → Error
> someError.message
    → Damn Error!
> someError.stack
  "Error: Damn Error
      at <anonymous>:1:17"
```

@codewithsimran

The error occured on the global execution context

Now let's define an error inside of a function

```
> function randomError()
  {
      const someError = new Error('Damn Error!')
  }
randomError();
> Error: Damn Error!
      at randomError (<anonymous>:...:)
      at <anonymous>
```

② we called randomError and the error occured inside it

① first we had execution context

We can also check @codeWithSimran

```
som randomError().stack
....
```

So we can trace where exactly the error occured

We also have other error in JS

- new SyntaxError → example £,
- new ReferenceError → when something is not defined

So how are errors handled in Javascript behind the scenes?

When an error occurs, it check the execution context on top of stack if there is a catch (to handle the error), if not it goes one level down and checks the next execution context if there is a catch

@codeWithSimran

If there is no catch anywhere, the onerror() function gets run on the browser

So the runtime catch : onerror() will handle the error for us

But again, this will stop the execution of the program and hence we should have our own ~~of~~ catch statements