

- The **Promise** object is created using the **new Promise** constructor.
- The **Promise** object represents the eventual completion (or failure) of an asynchronous operation, and its resulting value.
- The **Promise** object takes a function as its first parameter, called **executor function**. - `new Promise((resolve, reject) => { awf })`
- The **executor** function takes 2 parameters , **resolve** and **reject ()**, and as well **executor** initiates some **asynchronous work function**, in our case function named ``awf``.
- **Asynchronous work function**, when finished calls either **resolve** or **reject** and passes to them error or data.
- **resolve** and **reject** will at the end return either **data** or **error** that **asynchronous work function** gave to them, as a wrapped in a **Promise**.
- If resolved successfully data from resolve (which is a Promise) can be accessed with `.then()` .
- If an error occurs data from reject (which is a Promise) can be accessed with `.catch()` .

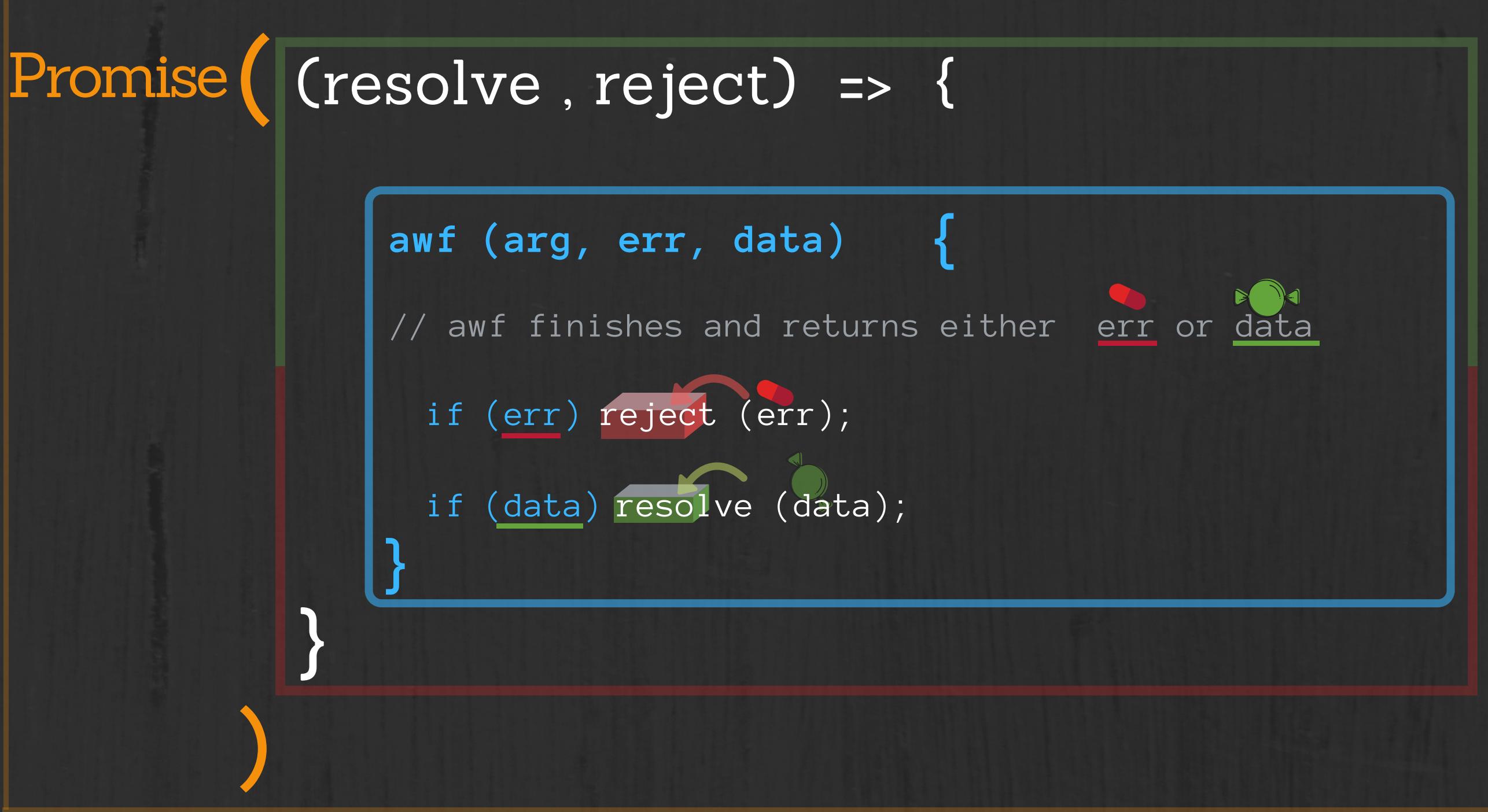
```
return new Promise ( (resolve , reject) => {  
    awf (arg, err, data) {  
        // awf finishes and returns either err or data.  
        if (err) reject (err);  
        if (data) resolve (data);  
    }  
})
```

- **resolve** and **reject** are methods, to which future result or error will be passed.
- They are methods, but we can think of them as 2 empty boxes. One of them will be filled and sent in the future.

```
return new Promise( (resolve , reject) => {  
       
    awf (arg, err, data) {  
        // awf finishes and returns either err or result.  
  
        if (err) reject (err);  
  
        if (data) resolve (data);  
    }  
}  
)
```

- Function that does our asynchronous work, in our case '**awf**' has 2 empty variables, **err** and **data**.
- After finishing it's work, '**awf**' saves the result in it's **data** variable. If unable to get the result it saves the Error in it's **err** variable.
- If there is an error, we invoke **reject** giving it **error** value, and if work was resolved sucessfully we invoke **resolve** giving it the **data received**.
- Same as putting something into the box and sending it off. **awf** is the one that gets something and puts it in the right box (**reject** or **resolve**).

```
return new Promise((resolve, reject) => {  
    awf (arg, err, data) {  
        // awf finishes and returns either err or data  
        if (err) reject (err);  
        if (data) resolve (data);  
    }  
})
```



(**resolve** , **reject**) => { // resolve and reject are methods to which future result or error will be passed.

```
awf (arg, err, data) { //`arg` - some additional value passed to the `awf`, e.g. URL
    if (err) reject (err); // if error occurs , `error` is placed in the `reject` , and it can be accessed later with .catch() method
    if (data) resolve (data); // if action resovles succesfully, `data` is placed in the `resolve` and can be accessed later with .then() method.
}
```

```
return new Promise((resolve , reject) => {  
    awf (arg, err, data) {  
        // awf finishes and returns either err or data  
        if (err) reject (err);  
        if (data) resolve (data);  
    }  
})
```