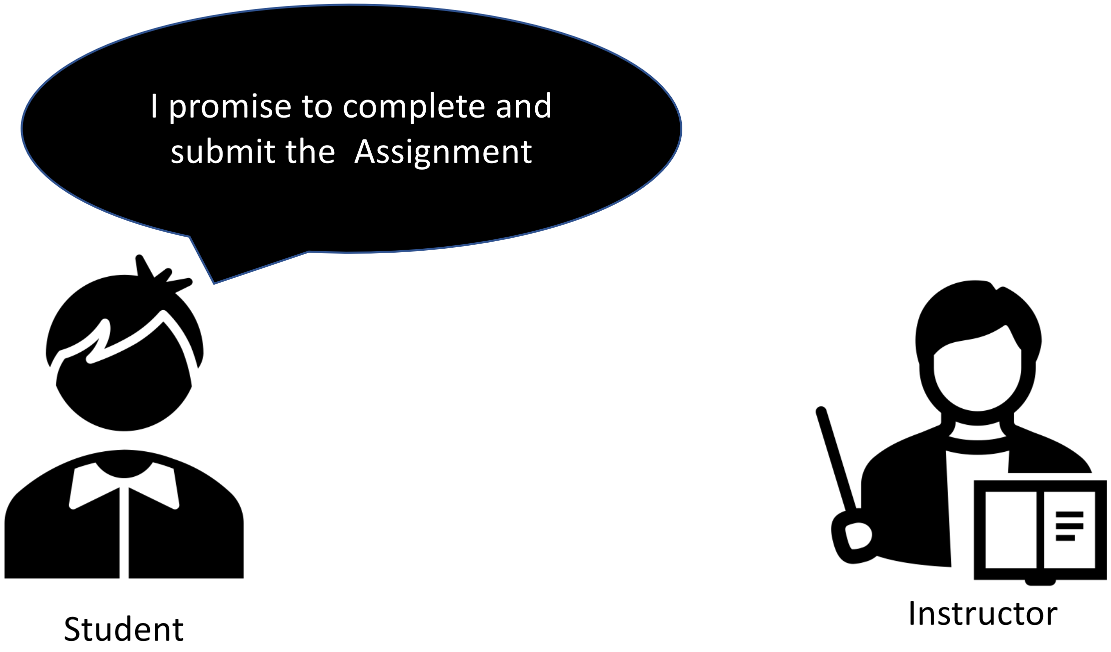
JavaScript Promises

Promises are events that happen in the future

JS Promises

- A JS **promise** is a proxy for value that is not known when the promise is created.
- The promise will return a value in the future.
  - Getting data over the internet
  - Submitting your assignments

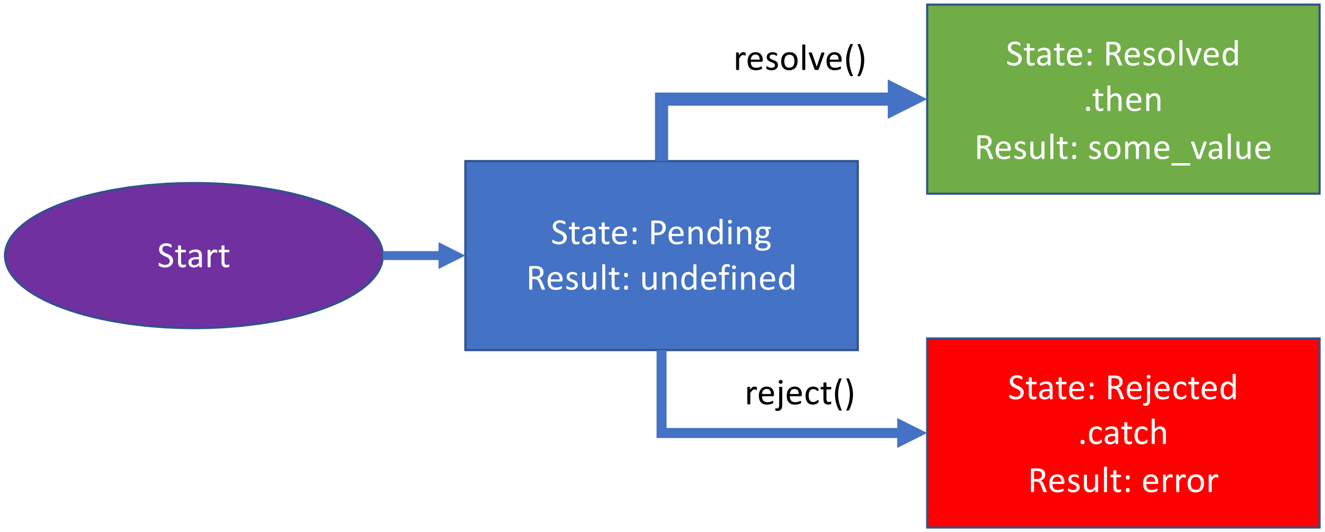
A Simple Promise Analogy



A Simple Promise Analogy

- Suppose you **promise** to submit an assignment to your instructor.
  - There are several possible outcomes:
    - **Pending** – You decide to do the assignment
    - **Resolved** – you complete and submit the assignment
    - **Rejected** – Oops you didn’t submit the homework on time
      - **Catch** – Talk to your instructor and get an extension

A Simple Promise Flowchart



Only three possible states:

1. Pending
   - Waiting to be invoked
2. Resolved
   - Successfully completed
3. Rejected
   - Did not complete
   - Typically used with error handling

// Simple Promise Analogy
// Declare Your Promise and do the Assignment

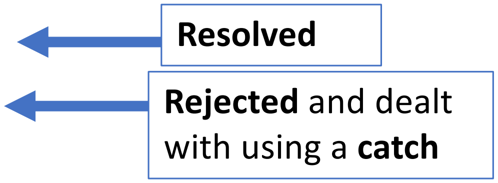
// A *pending* promise

```
const promise = doAssignment(‘Assignment#1’)


// if you complete your homework then we say that the promise has
// been resolved
doAssignment(’Assignment#1’)
          .then(submittedAssigment)
```

Resolved

// Simple Promise Analogy
//Declare Your Promise, submit the assignment
// And have a contingency plan in place

```
// declare pending promise
const promise = doAssignment(‘Assignment#1’)

// if you complete your homework then we say that the promise has
// been resolved (.then)
// if you do not submit your assignment (status: rejected) we have a
contingency plan (catch)
doAssignment(’Assignment#1’)
          .then(submittedAssigment)
          .catch(AskforExtension)
```



Resolved

Rejected and dealt with using a **catch**

Simple Promise in JS

```
# Consider the following example, you have 1000ms to complete your assignment.
# Guess you shouldn’t have procrastinated
const doAssignment = true;
const my_promise = new promise(function(resolve, reject) =>{
        // Do the Assignment in 1000 ms?
        setTimeOut(() => {
                //Did you complete the assignment?
        },1000)
})
```

Promise in JS

```
const doAssignment = true;
const my_promise = new promise(function(resolve, reject) =>{
        // Do the Assignment in 1000 ms?
        setTimeOut(() => {
                if (”Assignment is completed”){
                        resolve(“Your assignment has been submitted”)
                } else {
                        reject(“No assignment submitted”)
                        //error handling ask for an extension
        }
        }, 1000)
})
```