

Experiment No.9

//Async await

```
function doAnything(data) {
    return new Promise(
        (resolve,reject) => {
            setTimeout(()=>{
                if(typeof data=='number') {
                    resolve(data*data);
                } else {
                    reject("An error occurred");
                }
            },1000);
        });
}

async function chainStuff() {
    let a = await doAnything(2);
    console.log(" in "+a);
    let b = await doAnything(a);
    console.log(" in "+b);
    let c = await doAnything(b);
    console.log(" in "+c);
    return c;
}

chainStuff().then(
    result=>{
        console.log("out r"+result);
    }).catch(error=>{
        console.log("out e"+error);
    });
```

Output:

```
[student@localhost Rushi]$ node async-await.js
in 4
in 16
in 256
out r256
```

//Promise

```
function doAnything(data) {  
  return new Promise(  
    (resolve,reject) => {  
      setTimeout(()=>{  
        if(typeof data=='number') {  
          resolve(data*data);  
        } else {  
          reject("An error occurred");  
        }  
      },1000);  
    });  
}  
doAnything(2).then(a=>{  
  console.log(a);  
  return doAnything(a);  
}).then(b=>{  
  console.log(b);  
  return doAnything(b);  
}).then(c=>{  
  console.log(c);  
}).catch(error=>{  
  console.log(error);  
});
```

Output:

[student@localhost Rushi]\$ node promise.js

4

16

256