# Why promise is needed

```
//asynchronous : occuring at the same time
        const f1 = () => {
          setTimeout(() => {
            return 5;
          }, 5000);
        };

        const f2 = (x) => {
          console.log(x + 6);
        };

        let n1 = f1();
        f2(n1);
```

# Use callback to solve the issue

```
const f1 = (fnc) => {
    setTimeout(() => {
     fnc(5);
    }, 5000);
   };

   const f2 = (x) => {
    console.log(x + 6);
   };

   f1(f2);
```

# Use promise and .then

```
const f1 = () => {
   return new Promise((resolve, reject) => {
    setTimeout(() => {
     resolve(5); //use resolve instead of return
    }, 5000);
   });
  };

  const f2 = (x) => {
   console.log(x + 6)
  };

  f1().then((a) => f2(a));
```

# Fetch with async await

```
<body>
  <ul id="userDiv"></ul>
  <script>
  //   const url = "https://jsonplaceholder.typicode.com/users/";
   const url = "students.json"
   const showUsers = async () => {
    try {
     const response = await fetch(url);
     const json = await response.json();
     json.map((element) => {
      let li = document.createElement("li");
      li.innerHTML = element.name;
      userDiv.append(li);
     });
    } catch (error) {
     console.log(error);
    }
   };
   showUsers();
  </script>
 </body>
```