

# Advanced JavaScript

## Callbacks

Function passed as an argument to another function.

Allows a function to invoke another function.

```
let message;
function displayConsole(output) {
  console.log(output);
}
function displayDOM(output) {
  document.getElementById("myH1").innerHTML = output;
}
function setMessage(text, callBack) {
  message = text;
  callBack();
}

setMessage("Hi Mom", displayConsole)
```

## Promises

```
let p = new Promise((resolve, reject) => {
  let a = 1 + 1;
  if (a == 2) {
    resolve('Success');
  } else {
    reject('failed')
  }
})
p.then((message) => {
  console.log('This is in the then' + message)
}).catch((message) => {
  console.log('This is in the catch' + message)
})
```

## Async Await

```
async function doWork() {
  const response = await makeRequest('Google');
  console.log('Response Received')
  const processedResponse = await processRequest(response);
  console.log(processResponse)
}
doWork();
```

## Fetch API

```
fetch('https://reqres.in/api/users', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({
    name: 'User 1'
  })
}).then(res => {
  return res.json()
})
.then(data => console.log(data))
.catch(error => console.log('error'))
```

## axios

```
const axios = require('axios'); // For Node.js; in the browser, you can include Axios via a script tag
axios.get('https://reqres.in/api/users')
  .then(response => {
    // Handle the successful response
    console.log('Data:', response.data);
  })
  .catch(error => {
    // Handle any errors that occurred during the request
    console.error('Error:', error);
  });
```