

1. Write a function that takes a callback and executes it after printing a message.

```
function message(cb) {  
  console.log("Executing...");  
  cb();  
}  
message(() => console.log("Callback done"));
```

2. Use `setTimeout` to simulate an asynchronous operation.

```
setTimeout(() => {  
  console.log("Executed after 1 second");  
}, 1000);
```

3. Create a promise that resolves with the message "Hello World".

```
let p = new Promise((resolve, reject) => {  
  resolve("Hello World");  
});  
p.then(data => console.log(data));
```

4. Create a promise that rejects and handle the error using `.catch()`.

```
let p = new Promise((resolve, reject) => {  
  reject("Something went wrong");  
});  
p.catch(err => console.log(err));
```

5. Write a promise that resolves if a number is even and rejects if it is odd.

```
function checkEven(num) {  
  return new Promise((resolve, reject) => {  
    if (num % 2 === 0) {  
      resolve("Even");  
    } else {  
      reject("Odd");  
    }  
  });  
}  
checkEven(4).then(console.log).catch(console.log);  
checkEven(3).then(console.log).catch(console.log);
```

6.Chain multiple .then() methods to modify a resolved value.

```
Promise.resolve(2)
  .then(n => n + 2)
  .then(n => console.log(n));
```

7.Create an async function that returns a greeting message.

```
async function greet(name) {
  return "Hello, " + name;
}
greet("Asha").then(console.log);
```

8.Wait for 2 seconds using async/await and then print a message.

```
async function wait() {
  await new Promise(resolve => setTimeout(resolve, 2000));
  console.log("Waited!");
}
wait();
```

9.Fetch a post from an API and display its title using async/await.

```
async function fetchPost() {
  let response = await fetch("https://jsonplaceholder.typicode.com/posts/1");
  let data = await response.json();
  console.log(data.title);
}
fetchPost();
```

10.Fetch data safely using try/catch to handle errors.

```
async function safeFetch() {
  try {
    let response = await fetch("https://jsonplaceholder.typicode.com/posts/1");
    let data = await response.json();
    console.log("Title:", data.title);
  } catch (error) {
    console.log("Failed:", error);
  }
}
safeFetch();
```

11. Create a reusable async function that fetches a post by ID.

```
async function safeFetch() {  
  try {  
    let response = await fetch("https://jsonplaceholder.typicode.com/posts/1");  
    let data = await response.json();  
    console.log("Title:", data.title);  
  } catch (error) {  
    console.log("Failed:", error);  
  }  
}  
safeFetch();
```