async / await

- **async** functions always return a promise
- Can use the **await** keyword to resolve promises inside the function

async / await

```
async function sum(a, b) {
  return a + b;
}

sum(1, 2).then(function (result) {
  console.log(result);
  // 3
});
```

async / await

```
async function sum(a, b) {
  return a + b;
}

const result1 = await sum(1, 2);
const result2 = await sum(3, 4);
const result3 = await sum(result1, result2);
console.log(result3);
```

Converting

From callback to promise

```
function getLocation() {
  return new Promise(function (resolve, reject) {
    navigator
    .geolocation
    .getCurrentPosition(resolve, reject);
  })
}
```

Converting

From callback to promise

```
getLocation()
.then(function (location) {
  console.log(location);
})
.catch(function (error) {
  console.log(error);
})
```

async / await

```
async function printLocation() {
  const location = await getLocation();
  console.log(location);
}
```

async / await

Error handling:

```
async function printLocation() {
  try {
    const location = await getLocation();
    console.log(location);
  } catch (error) {
    console.log(error);
  }
}
```