Async - Await

1. What is async ?

Keyword

* (always returns a Promise.)

```
// always returns a promise
async function getData() {
  return "Namaste";
}

const data = getData();
console.log(data);
```

Promise

```
Elements  Console  Sources
top  Filter  Default levels
No issues
▸ Promise {<fulfilled>: 'Namaste'}  index.js:18
```

2. Using await with async :

Keyword

* (async - await are used to handle Promises.)

```
const p = new Promise((resolve, reject) => {
  resolve("Promise Resolved Value!!");
});

async function handlePromise() {
  const val = await p;
  console.log(val);
}
handlePromise();
```

(await is used in-front of a Promise.)

(await is only be used inside an async Function.)

3. Difference B/w Promises and async - await

```
const p = new Promise((resolve, reject) => {
  setTimeout(() => {
    resolve("Promise Resolved Value!!");
  }, 10000);
});

// await can only be used inside an async function
async function handlePromise() {
  console.log("Namaste JavaScript");
  // JS Engine was waiting for promise to resolved
  const val = await p;
  console.log(val);
}
handlePromise();

// function getData() {
// // JS Engine will not wait for promise to be resolved
//   p.then(res => console.log(res));
//
//   console.log("Namaste JavaScript");
// }
//
// getData();
```

Promise

(Using async - await)

(Using Promises)

(waits for 10sec and then prints.) ?

(prints Namaste JS immediately.) ?

```
const p = new Promise((resolve, reject) => {
  setTimeout(() => {
    resolve("Promise Resolved Value!!");
  }, 10000);
});

// await can only be used inside an async function
async function handlePromise() {
  console.log("Namaste JavaScript");
  // JS Engine was waiting for promise to resolved
  const val1 = await p;
  console.log(val1);

  const val2 = await p;
  console.log("Namaste JavaScript 2");
  console.log(val2);
}
handlePromise();
```

val 1

val 2

```
Hello World!!                          index.js:20
Namaste JavaScript                     index.js:23
Promise Resolved Value!!               index.js:24
Namaste JavaScript 2                   index.js:27
Promise Resolved Value!!               index.js:28
```

(why did we got this after 10 sec ?) * ?

(Understand using Sources Tab and Debugging and ChatGPT.)

```
const p1 = new Promise((resolve, reject) => {
  setTimeout(() => {
    resolve("Promise Resolved Value!!");
  }, 10000);
});

const p2 = new Promise((resolve, reject) => {
  setTimeout(() => {
    resolve("Promise Resolved Value!!");
  }, 5000);
});

// await can only be used inside an async function
async function handlePromise() {
  console.log("Hello World!!");
  // JS Engine was waiting for promise to resolved
  const val1 = await p1;
  console.log("Namaste JavaScript");
  console.log(val1);

  const val2 = await p2;
  console.log("Namaste JavaScript 2");
  console.log(val2);
}
handlePromise();
```

val 1

val 2

```
Hello World!!                          index.js:20
Namaste JavaScript                     index.js:23
Promise Resolved Value!!               index.js:24
Namaste JavaScript 2                   index.js:27
Promise Resolved Value!!               index.js:28
```

(why did we got this after 10 sec and not 5 sec ?) * ?

(Understand using Sources Tab and Debugging and ChatGPT.)

```
const p1 = new Promise((resolve, reject) => {
  setTimeout(() => {
    resolve("Promise Resolved Value!!");
  }, 5000);
});

const p2 = new Promise((resolve, reject) => {
  setTimeout(() => {
    resolve("Promise Resolved Value!!");
  }, 10000);
});

// await can only be used inside an async function
async function handlePromise() {
  console.log("Hello World!!");
  // JS Engine was waiting for promise to resolved
  const val1 = await p1;
  console.log("Namaste JavaScript");
  console.log(val1);

  const val2 = await p2;
  console.log("Namaste JavaScript 2");
  console.log(val2);
}
handlePromise();
```

val 1

val 2

```
Hello World!!                          index.js:26
Namaste JavaScript                     index.js:29
Promise Resolved Value!!               index.js:30
Namaste JavaScript 2                   index.js:33
Promise Resolved Value!!               index.js:34
```

(why did we got this after 5sec ?) ?

(And this after 10sec ?) ?

(Understand using Sources Tab and Debugging and ChatGPT.)