Javascript Promises

Javascript Promises

A Promise is an object representing the eventual completion (or failure) of an asynchronous operation.

```
let promise = new Promise(function(resolve, reject) {
  // asynchronous task
});
```

States of a Promise

Pending : Initial state, neither fulfilled nor rejected.

Fulfilled : Operation completed successfully.

Rejected : Operation failed

Javascript Promises

Example

```
let myPromise = new Promise((resolve, reject) => {
  let success = true;
  if (success) {
    resolve("Task completed!");
  } else {
    reject("Something went wrong.");
  }
});

myPromise
  .then(result => console.log(result))
  .catch(error => console.log(error));
```

then() Method : Handles the **success** result of a Promise

.catch() Method : handles the **rejected** result of a Promise

Javascript Promises

Async & Await

Async/Await is a modern and cleaner way to handle Promises. It makes asynchronous code look more like synchronous code.

async Keyword: Declares a function that always returns a Promise**.**

await Keyword: Waits for the Promise to resolve or reject — it can only be used inside async functions.**.**

```
function fetchData() {
  return new Promise(resolve => {
      resolve("Data received");
  });
}
async function getData() {
  console.log("Fetching...");
  let result = await fetchData();
  console.log(result);
}
getData();
```

Javascript setTimeout & setInterval

setTimeout(): setTimeout(function, milliseconds)

setTimeout() executes a function after a specified delay (once).

```
setTimeout(() => {
  console.log("This runs after 3 seconds");
}, 3000);
```

setInterval() : setInterval(function, milliseconds)

setInterval() executes a function repeatedly after a specified time interval.

```
let count = 0;
let timer = setInterval(() => {
  count++;
  console.log(`Count: ${count}`);
  if (count === 5) {
    clearInterval(timer);  // Stops the interval
  }
}, 1000);
```

Javascript Fetch

The Fetch API allows you to make network requests with a promise-based syntax.

```
fetch( url, options )
  .then( response => response.json() ) // or .text(), .blob(), etc.
  .then(data => {
    // handle data
  })
  .catch(error => {
    // handle error
  });
```

```
fetch('https://jsonplaceholder.typicode.com/posts/1')
  .then(response => response.json())
  .then(data => {
    console.log(data);
  })
  .catch(error => {
    console.error('Error:', error);
  });
```

indiXpert

THANK YOU

PHONE NUMBER

(+91) 778 899 2897

WEBSITE

www.indixpert.com