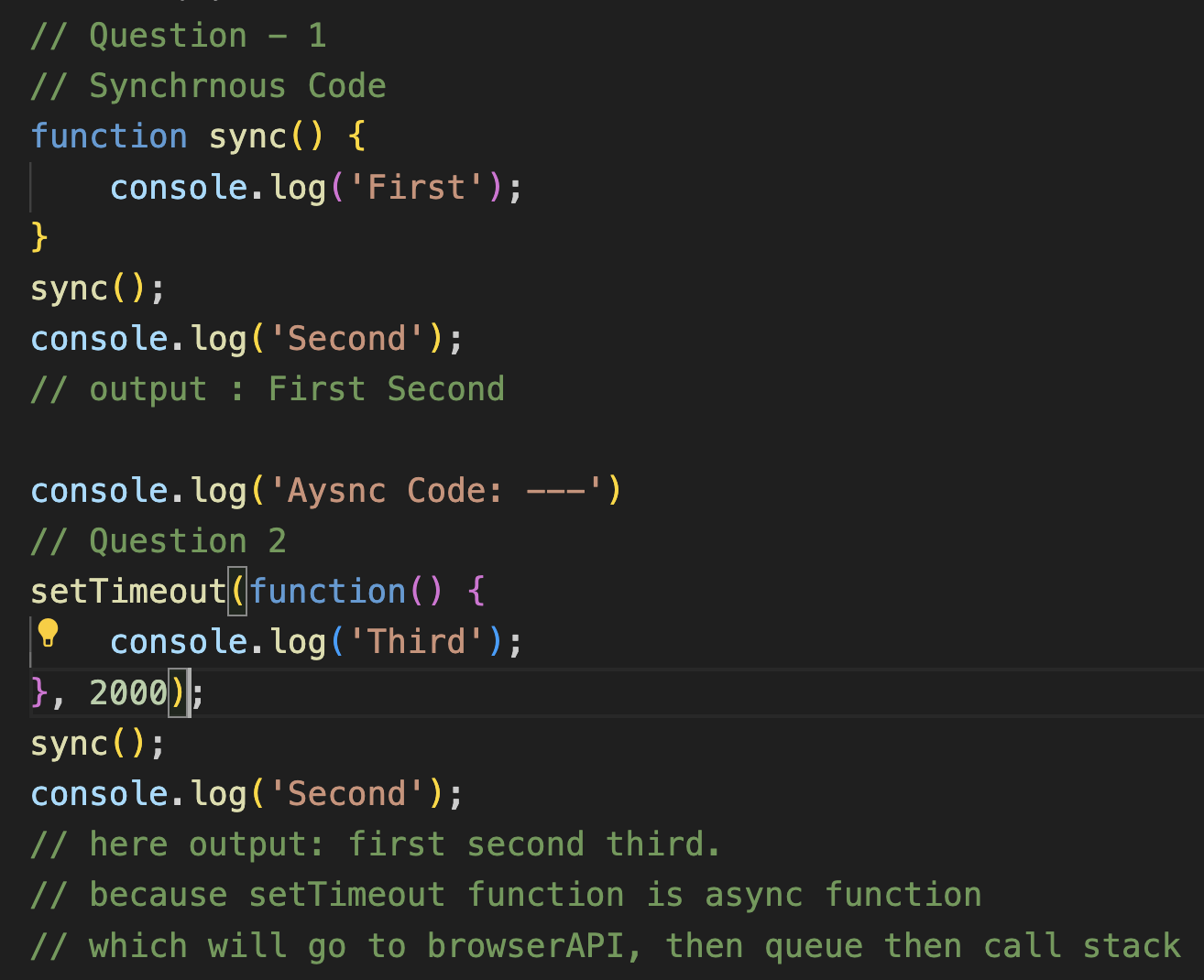
**JavaScript DOM + Modern JS**

**(Interview Concept)**

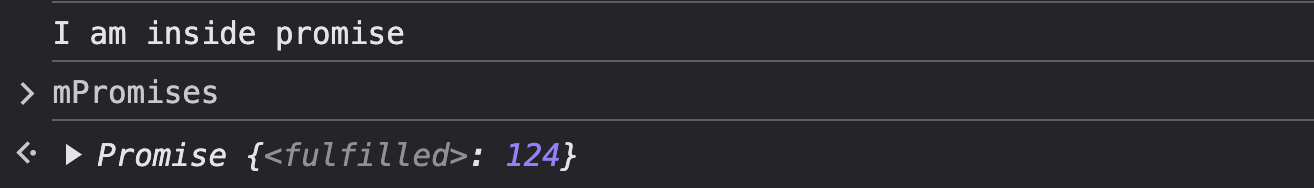
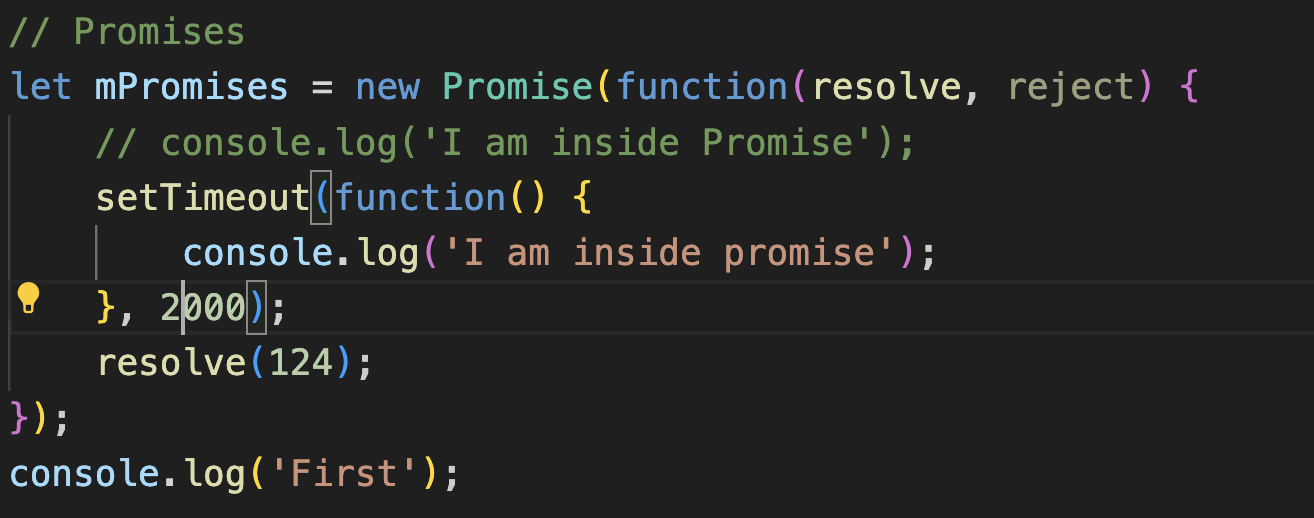
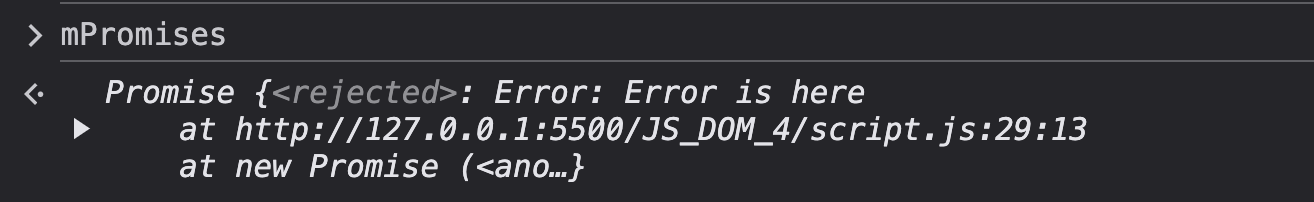
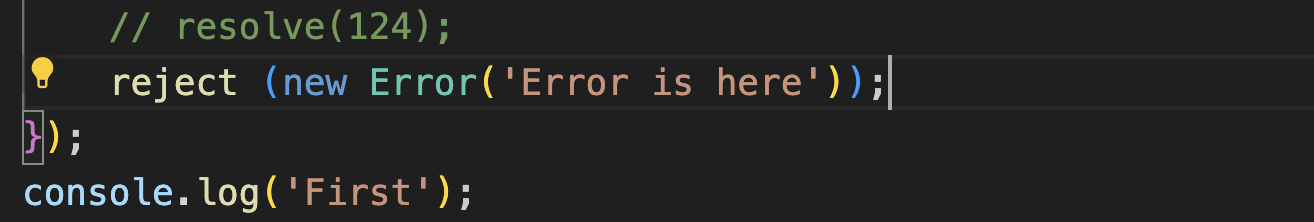
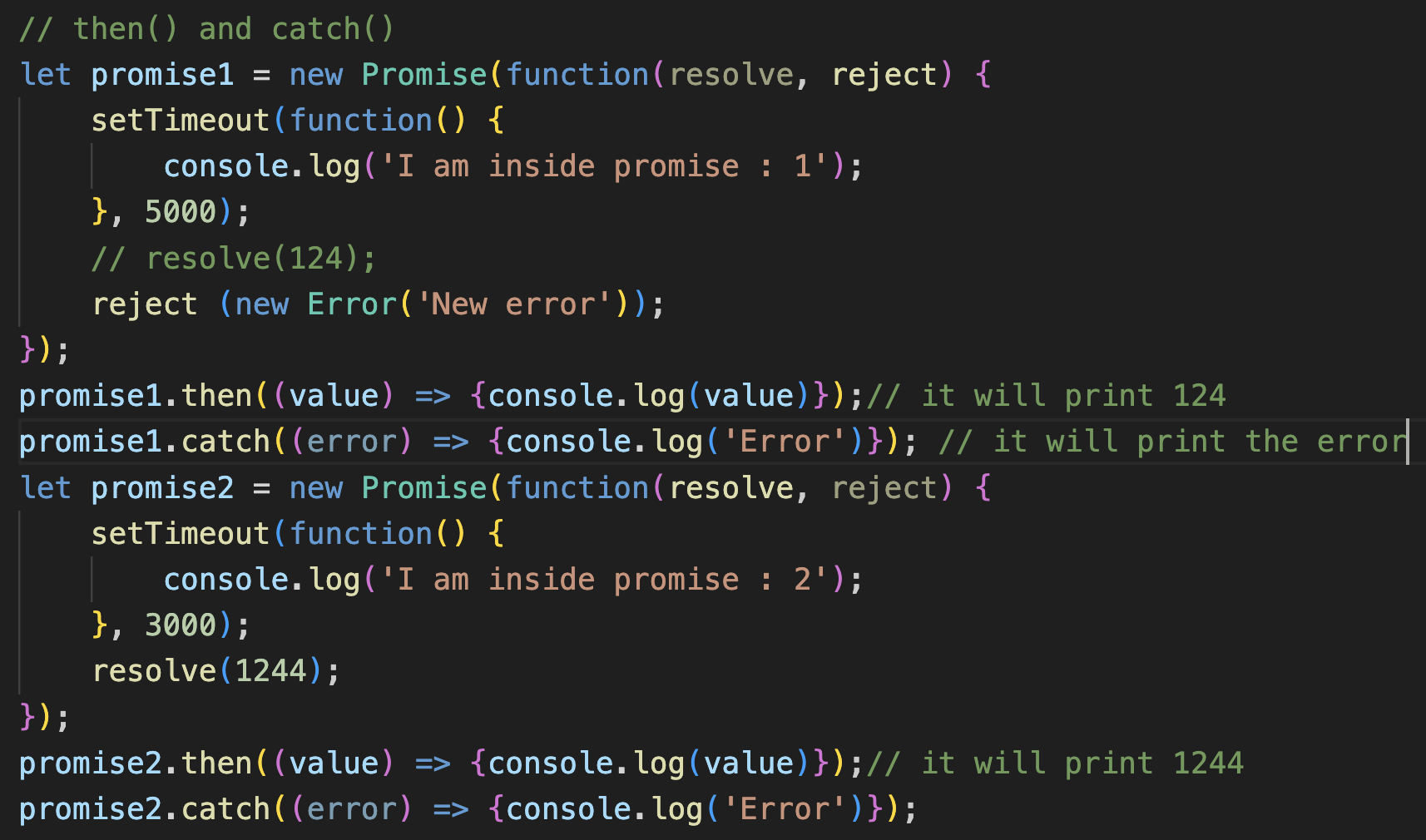
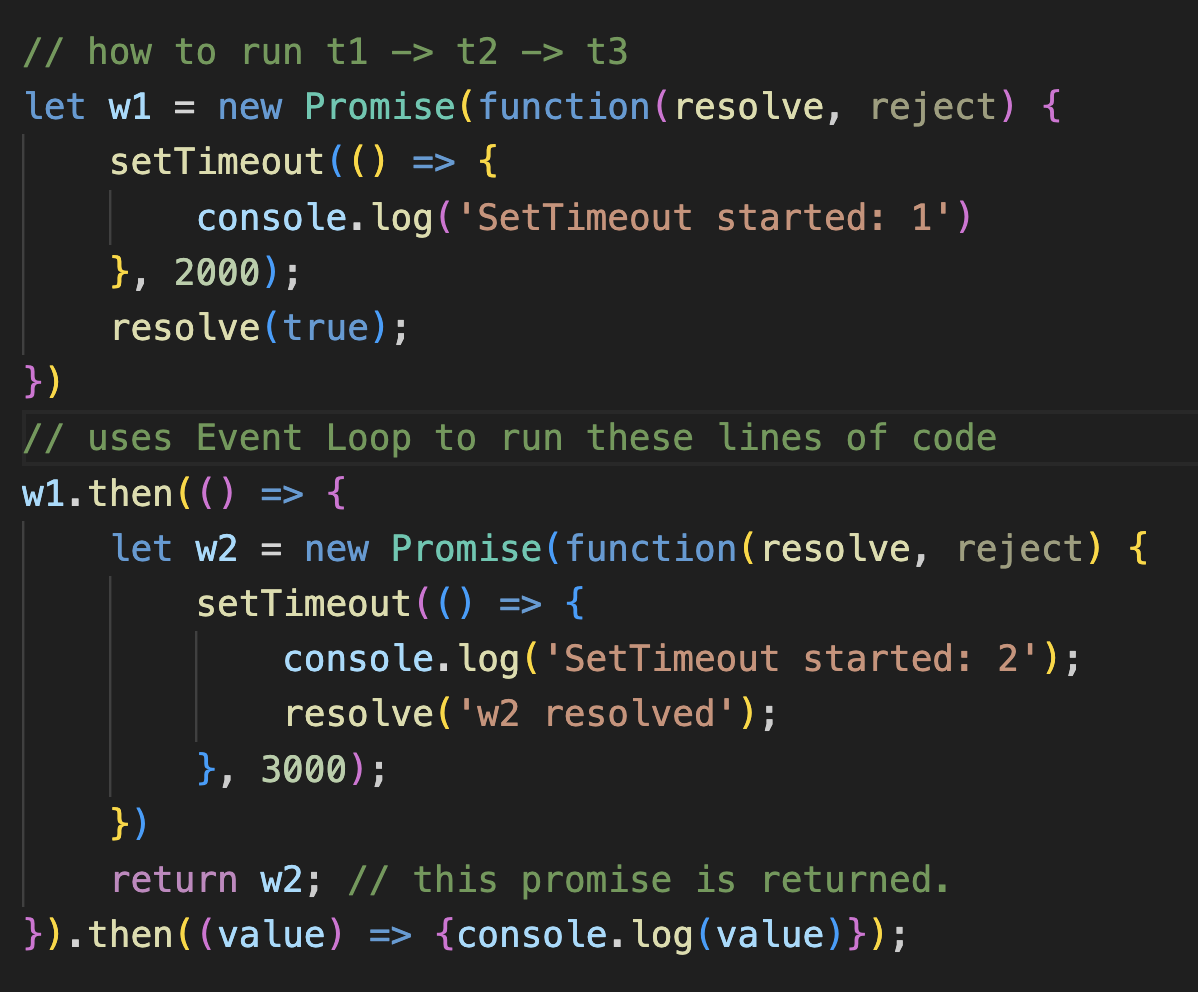
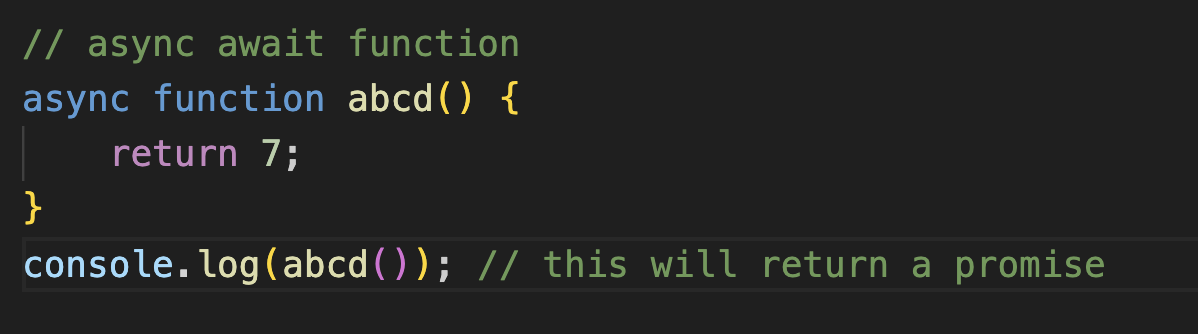
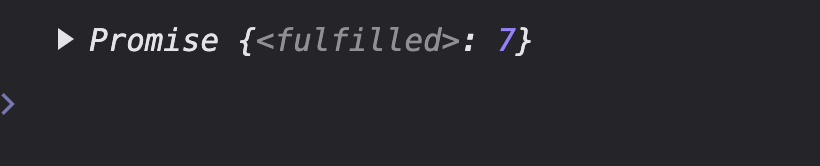
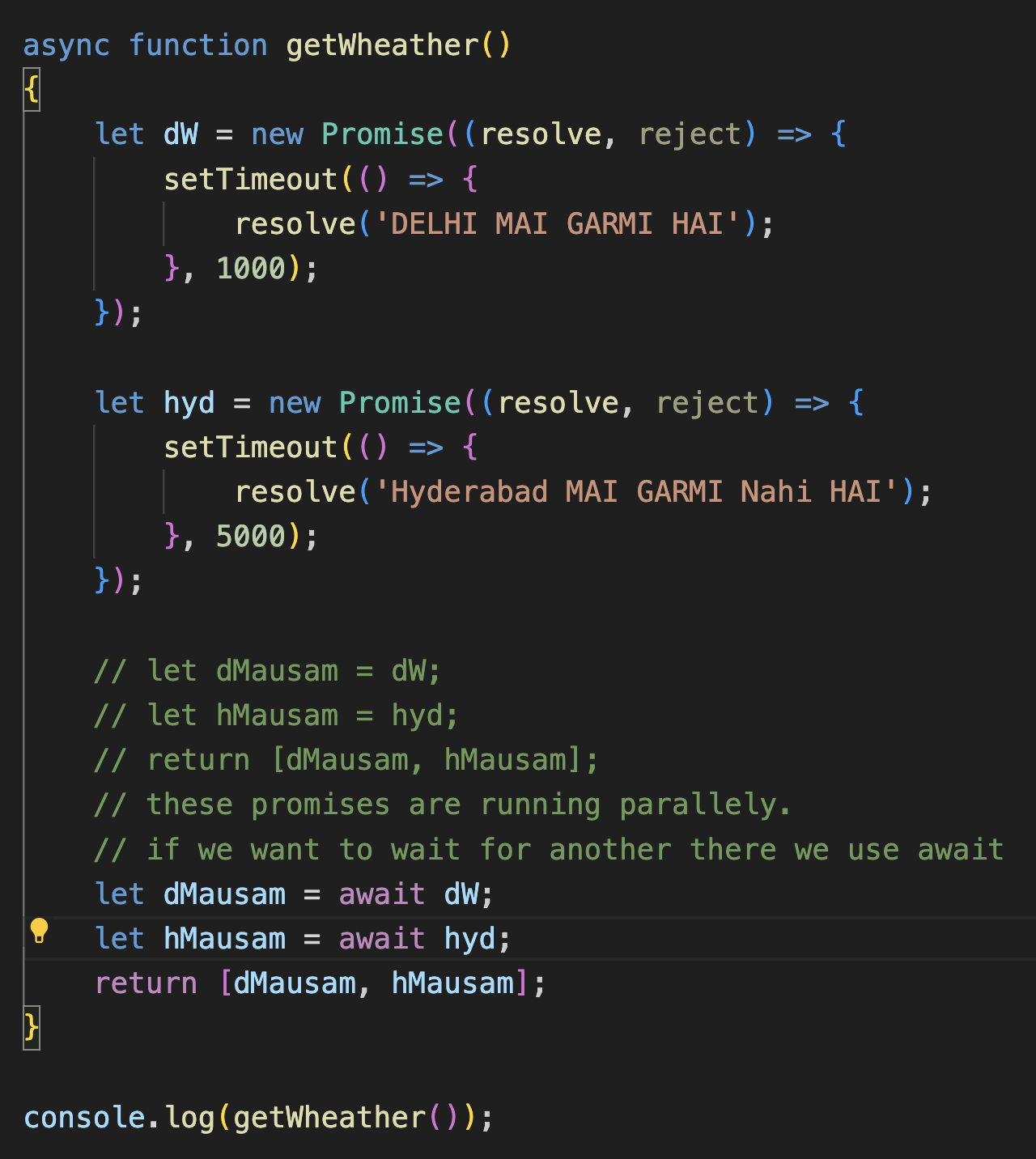
**Asynchronous JS**:

* These lines of code runs but we do not know when.
* They are run via the process of Event Loop.
* 
* Features of Async Code:
  + Clean and Concise.
  + Better error handling.
  + Debugging is easier.
  + can contain zero or more await expressions

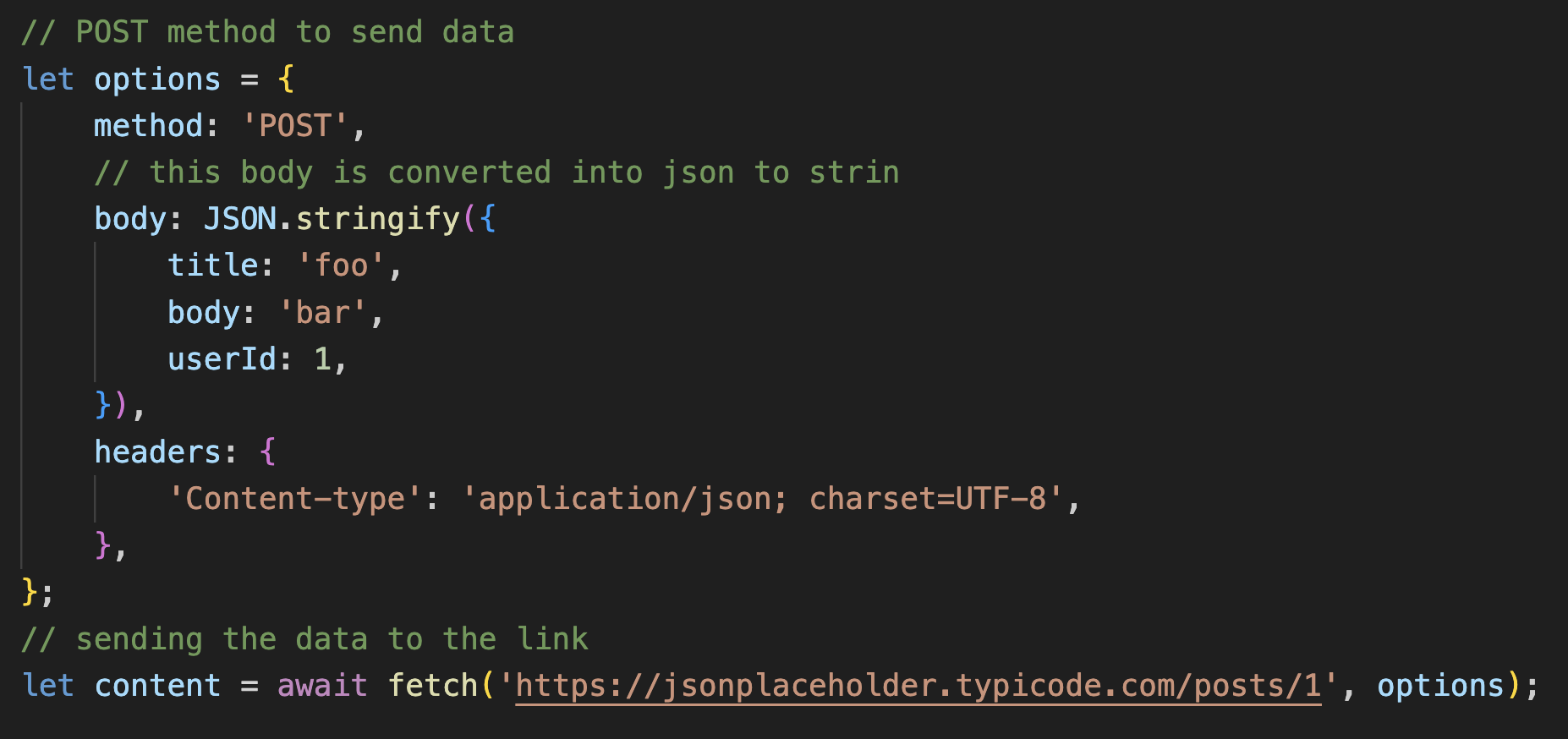
**API**:

* Application programming interface
* A kind of interface which connects frontend to backend or way to communicate with backend.
* A way to communicate between different software component.
* Example to send request we use API, then to take response we use API.

**Promises**:

* Anything we want to execute parallelly we use promises which is a kind of async code.
* It takes two parameters resolve and reject which means if its gets completed successfully means resolve else rejected.
* 
* 
* Here two states of this promise are first one is Resolve (Fulfilled), Reject and Pending.
* 
* 
* The promise object represents the eventual completion (or failure) of an asynchronous operation and its resulting value.
* Three states of promise are
  + Pending
  + Fulfil
  + Reject.
* Two methods which are applied on promise after the task or promise is completed.
  + then(): Fulfilled value when returned is managed by then()
  + catch(): error is handled by catch().
  + 
* Basically, for running the async code we use Promises (so that we can run synchronous code fastly).
* 
* Writing again then and then is called Promise Chaining. But we do not need to write promise chaining.
* Here we use **async await** function.
  + Special syntax used to work with promises.
  + Because every async function returns a promise.
  + How to write async await function.
  + 
  + 
* 
* The await keyword can only be used inside an async function.
* The await keyword makes the function pause the execution and wait for a resolved promise before it continues:

**Fetch API**

* API is used to retrieve the data or send the data.
* Needs an URL to run.
* 
* Example using image API and we are using GET call:
  + 
* Sending some data using POST call in [options]
* 
* stringify(): converts a JavaScript value to JSON string.

**Closures**:

* a closure is a combination of a function bundled together (enclosed) with references to its surrounding state (the lexical environment).
* A closure gives you access to an outer function’s scope from an inner function.
* 