Synchronous Vs Asynchronous Synchronous: Every statement of code get executed one by one So basically, a Statement has to wait for earlier statement to get executed Eg - Console. log ("I"); Console. log ("eat"); console. log ("ice-oueam"); It will point I first, then eat, after that ice-cream Asynchronous: It allows program to be executed immediately without blocking the code. Unlike the Synchronous method It doesn't wait for earlier statement to get executed first. Each task execute completed Independently

Console. 10g set Timeout (() =) {
 consoli.log ("eat"); }, 2000) Console. log ("Ice (ream") It will print " T" "Ice (ream") (will execute immediately)
"eat" (will point after 25) Hsynchronous Functions > It contains async keyword. How to use in Normal Function declaration async function name (arg) { How to use in an aviow function Cons function Name = async (arg) >> {

Lione always return
Asynchronous functions always return promises
Deromises
a chirt De
It doesn't mother what you reduced
The strengt value will always be fromse
It doesn't matter what you return.  The returned value will always be promise
$eg \rightarrow$
const get One = async = 2
Eg $\rightarrow$ Const get One = async $\rightleftharpoons$ $\Rightarrow$ $\S$ return 1;
2 and the second the second second
Const peromise = act (no ()
Const promise = get One (); Console. log (promise)
consecting (promise)
The a of the
The await keyword
The await keyword lets you wait for promise to resolve. Once promise is resolved it retwens the parameter parameter parameter
to resolve. Once promise
it retwens the payoned
it returns the parameter passed into
, with the same of
$C_{\bullet}$
Eg- Con

const get One = async  $\Rightarrow \xi$ get One (). then (value => {

console log (value) 3) 3/11 Now use of await keyword const test = async = > {
 const one = await get One ();
 Console log (one);
} test () We can only use await when we have async. Let's implement the fetch API code using asynct/await:

```
Fetch Data = async () => {
      quites = await Fetch ("http://--/quotes");
 const response = await quites ison();
 Console. log (response);
   Fetch Data ();
Me can also handle errors in async fawait
by using try and Catch.
const fetch Data = async () >>
    const quotes = await Fetch ("http:// ");
    (onst susponse = await, quotes, json ()
      Consol. log (response);
    catch (error) 5
         Console. jug (error).
    Fetch Data ();
```

Synchronous Vs Asynchronous Synchronous: Every statement of code get executed one by one So basically, a Statement has to wait for earlier statement to get executed Eg - Console. log ("I"); Console. log ("eat"); console. log ("ice-oueam"); It will point I first, then eat, after that ice-cream Asynchronous: It allows program to be executed immediately without blocking the code. Unlike the Synchronous method It doesn't wait for earlier statement to get executed first. Each task execute completed Independently

Synchronous Vs Asynchronous Synchronous: Every statement of code get executed one by one So basically, a Statement has to wait for earlier statement to get executed Eg - Console. log ("I"); Console. log ("eat"); console. log ("ice-oueam"); It will point I first, then eat, after that ice-cream Asynchronous; It allows program to be executed immediately without blocking the code. Unlike the Synchronous method It doesn't wait for earlier statement to get executed first. Each task execute completed Independently

Console. log ("T") Egset Timeout (() => { consoli. log ("eat"); } ,2000) Console. log ("Ice (ream") It will point " I" "Ice Cream" ( will execute immediately) "eat" (will print after 2s) Hsynchronous Functions > It contains async keyword. How to use in Normal Function declaration async function name (arg) { How to use in an avviow function Cons function Name = async (arg) > {

Asynchronous functions always retwen promises It doesn't matter what you retwern.

The retwened value will always be promise Console. log (promise) The await Keyword The await keyword lets you wait for promise to resolve. Once promise is resolved it retroins the parameter passed into Eg- Con

Eg > const get One = async  $\Rightarrow \xi$ netwin 1; 3 get One (). then (value => { consoli. log (value) 3); 1/1 Now use of await keyword const test = async = > {
 const one = await getOne ();
 Console.log (one);
} test() We can only use await when we have async. Let's implement the fetch API code using asynct/await:

Const Fetch Data = async () => { quites = await Fetch ("http://--/quotes"); const response = await quites , ison(); console. log (response); Fetch Data (); The can also handle errors in async await by using toy and Catch. const tetch Data = async () ⇒ 2 const quotes = await Fetch ("http:// "); (onst rusponse = await, quotes, json (); consoli log (response); catch (error) 5 Console. lug (croor). Fetch Data ();