1. Promise — vadalar.

let promise = new Promise(function(resolve, reject) {

setTimeout(() => resolve("done!"), 1000);

});

// resolve runs the first function in .then

promise.then(

result => alert(result), // shows "done!" after 1 second

error => alert(error) // doesn't run

);

1. Synchronous and Asynchronous tasks 1.
2. async and await
   * await doim async funksiya ichida ishlaydi

async function main() {

let promise = new Promise((resolve, reject) => {

setTimeout(() => resolve("done!"), 1000)

});

let result = await promise; // wait until the promise resolves (\*)

alert(result); // "done!"

}

main();

1. fetch — serverlarga ma'lumotlarni olib kelish uchun turli xil so'rovlarni yuborish.

So'rov yuborishning turlari:

* + POST (create)
  + GET (read)
  + PUT (update)
  + DELETE (delete)

// Basic syntax

let promise = fetch(url, [options])

// JSda misol uchun

let response = await fetch(url, options); // resolves with response headers

let result = await response.json(); // read body as json

* + **url** – the URL to access.
  + **options** – optional parameters: method, headers etc.
  + Agar option'lar berilmasa serverga GET so'rov yuborilgan bo'ladi

So'rov yuborilganda natijada quyidagicha ma'lumotlar qaytishi mumkin:

* + **status** – HTTP status code, e.g. 200.
  + **ok** – boolean, true if the HTTP status code is 200-299.

let response = await fetch(url);

if (response.ok) { // if HTTP-status is 200-299

// get the response body (the method explained below)

let json = await response.json();

} else {

alert("HTTP-Error: " + response.status);

}

**POST** method orqali ma'lumot yuborish

let user = {

name: 'John',

surname: 'Smith'

};

let response = await fetch('/article/fetch/post/user', {

method: 'POST',

headers: {

'Content-Type': 'application/json;charset=utf-8'

},

body: JSON.stringify(user)

});

let result = await response.json();

alert(result.message);

**Amaliyot**

async function showAvatar() {

// read our JSON

let response = await fetch('/user.json');

let user = await response.json();

// read github user

let githubResponse = await fetch(`https://api.github.com/users/${user.name}`);

let githubUser = await githubResponse.json();

// show the avatar

let img = document.createElement('img');

img.src = githubUser.avatar\_url;

img.className = "promise-avatar-example";

document.body.append(img);

// wait 3 seconds

await new Promise((resolve, reject) => setTimeout(resolve, 3000));

img.remove();

return githubUser;

}

showAvatar();

1. CRUD - Create Read Update Delete

* Manba:

[JSONPlaceholder](https://jsonplaceholder.typicode.com/guide/)

Uyga vazifa:

1. O'tilgan mavzularni qayta ko'rib chiqish, mazularga oid qo'shimcha izlanish