Դաս 10: Promis-ներ

Promise-ներ՝

Promise-ը օբյեկտ է, որը ներկայացնում է գործողություն, որը կամ կավարտվի հաջողությամբ, կամ կձախողի ապագայում: **Promise**-ները լուծում են **քաոսային կոդը** խնդիրը:

Promise-ը ստեղծվում է **new Promise** կոնստրուկտորի միջոցով: Այս կոնստրուկտորը որպես արգումենտ ստանում է **executor** կոչվող ֆունկցիա, որն ինքն իր հերթին ստանում է երկու արգումենտ՝

resolve - կանչվում է, երբ գործողությունը հաջողությամբ ավարտվում է:

reject - կանչվում է, երբ գործողությունը ձախողվում է:

```
const myPromise = new Promise((resolve, reject) => {
    // Օրինակ ասինխրոն գործողություն
    let success = true;
// Կայացած տվյալ, որը կարող է փոփոխվել

    if (success) {
        resolve("Գործողությունը հաջողվեց!");
    } else {
        reject("Գործողությունը ձախողվեց...");
    }
});
```

Ինչու են պետք resolve և reject-ը

resolve(value)

Կանչվում է, երբ գործողությունը հաջողությամբ ավարտվում է: Որպես արգումենտ այն ստանում է արժեքը (օրինակ՝ տվյալներ սերվերից), որը հետագայում կօգտագործվի:

reject(reason)

Կանչվում է, երբ գործողությունը ձախողվում է: Որպես արգումենտ այն փոխանցում է սխալի մասին տեղեկություն (օրինակ՝ սխալի հաղորդագրություն կամ օբյեկտ):

Promise-ի վերամշակում

Promise-ը կարող է ունենալ երեք հիմնական վիճակ.

Pending (**սպասող**) — Սկզբնական վիճակ:

Fulfilled (**կատարված**) — Երբ կանչվում է **resolve():**

Rejected (**մերժված**) — Երբ կանչվում է **reject():**

Օգտագործվում են **.then** և **.catch** մեթոդները՝ համապատասխանաբար հաջող և սխալի դեպքում:

```
myPromise
  .then((result) => {
    console.log("Հաջողությամբ արդյունք:", result);
  })
  .catch((error) => {
    console.error("Սխալի հաղորդագրություն:", error);
  });
```

Օրինակ՝ պատկերացնենք, որ ասինխրոն գործողությունը ստուգում է ֆայլի առկայությունը:

```
const checkFile = new Promise((resolve, reject) => {
  let fileExists = false; // Փորձեք փոխել true և false

  if (fileExists) {
    resolve("Ֆայլը հայտնաբերվեց:");
  } else {
    reject("Ֆայլը բացակայում է:");
  }
});

checkFile
  .then((message) => {
    console.log("Հաջողություն:", message);
  })
  .catch((error) => {
```

```
    console.error("Սխալ:", error);
  });
```

Բացի այդ կարող ենք օգտագործել **.finally()** մեթոդը, որը կատարում է նույն ֆունկցիոնալը ինչպես **switch/case**-ում:
Promise-ները կարելի է կանչել շղթայաբար՝

```
function step1() {
  return new Promise((resolve) => {
    setTimeout(() => {
      console.log("Քայլ 1");
      resolve();
    }, 1000);
  });
}

function step2() {
  return new Promise((resolve) => {
    setTimeout(() => {
      console.log("Քայլ 2");
      resolve();
    }, 1000);
  });
}

function step3() {
  return new Promise((resolve) => {
    setTimeout(() => {
      console.log("Քայլ 3");
      resolve();
    }, 1000);
  });
}
```

```
step1()
.then(step2)
.then(step3)
.then(() => console.log("Բոլոր քայլերը ավարտված են:"));
```