ასინქრონული პროგრამირება - promise

სინქრონული არის პროგრამირება რომელიც ელოდება ჯერ წინა მოქმედებების შესრულებას და შემდეგ ხდება ეს

let x=10;

x=5;

შესრულდა იქსებისთვის ველიუს გაწერა ერთმანეთის მიყოლებით მაგრამ შეიძლება რაიმე მოქმედებას წამები და წუთები დაჭირდეს ასამუშავებლად და სანამ არ დასრულდება სხვა ვერაფერი მოხდეს და მოგვიწიოს ლოდინი სანამ არ დასსრულდება

ამის თავიდან ასაცილებლად გამოიყენება ასინქრონული პროგრამირება როცა კიარელოდები არამედ შენ შენ საქმეს აკეთებ და თან სხვა მოქმედებაც გაშვებბულია

let x=10;

x=5;

asicnvhFunction()

if (x>3){

    Math.max(1,2,3)

}

აქ ფუნქციის ასამუშავებლად ჯერ დროა საჭირო და შესაბამისად არის ასინქრონული და საკმარისია რომ დაიწყოს მოქმედება და გადადის მაშინვე სსხვა ბრძანებებზე როცა მორჩება მორჩება.

რაიმე საიტზე შესასვლელად საიტს ჩასატვირთად ჭირდება დრო რა დროსაც იტვირთება მთლიანი კოდი და ასინქრონული პროგრამირება რომ არა სანამ საბოლოოდ არ დასრულდება კოდების ჩატვირთვა იქამდე მოგვიწევდა ლოდინი

Axios - როგორც chrome არის ბრაუზერი ასევეა axios ჯავასკრიპტის ბრაუზერი

axios.get("https://www.youtube.com/");

იუთუბზე შესასვლელად შეიძლება დიდიხანი დაჭირდეს ამიტომაც არ უნდა დაველოდოთ და უნდა დავარიგოთ აქსიოსი თუ როგორ მოიქცეს იმშემთხვევაში თუ რეზულტატი დაბრუნდა იუთუბიდან და გავუშვათ.

function getVideo(website){

    //ალგორითი

}

let resilt=axios.get("https://www.youtube.com/");

resilt.then(getVideos);

რასაც აქსიოსი დააბრუნებს ჩავწერთ resilt ცვლადში და შემდეგ გამოვიძახებთ ამ ცვლადს თუშესრულდები მაშინ აამუშავე ფუნქცია getvideo

Axios.get აბრუნებს ობიექტს რომელსაც ქვია then (მაშინ) და მას გადაეცემა callback ფუნქცია სადაც არგუმენტად მთლიან ვებსაიტს მოგვცემს და მასზე რასაც გვინდა იმას ვიზამთ

და ამ ყველაფერს ეწოდება promise (დაპირება) რადგან axios.get\_ს დრო ჭირდება, მისი return ველიუ მთლიანი ვებსაიტი კიარაა არამედ promise ანუ დაპირება.

რადგან მოთხოვნა გაგზავნილი გვაქვს და აქსიოსმა დაგვიბრუნა ფრომისი შედეგად ველოდებით ორ პასუხს ან დადებითს fulfilled (.then(fn) ) ან უარყოფითს rejected (catch (fn) )

ანუ თუ აქსიოსმა დაgetა საიტი მაშინ სრულდება შესაბამისი ფუნქცია

ხოლო თუ ვერ დაგეთა მაშინ სრულდება catch ფუნქცია რითიც ვიჭერთ ერორებს

საბოლოოდ axios.get მეთოდი აბრუნებს ჯავასკრიპტის promise ობიექტს რომელსაც გააჩნია ორი მეთოდი catch და then რომლებიც იმის მიხედვით შესრულდებიან promise შესრულდება თუ არა თუ შესრულდა ჯავასკრიპტი then ფუნქციას ჩააწვდის მთლიან ვებსაიტს, ხოლო თუ ვერშესრულდა catch ფუნქციას გამოიძახებს და არგუმენტად მოგვცემს erors და შიგნით ეწერება რა დაერორდა

function getVideo(website){

    //ალგორითი

}

function errodLoad(){

    //ალგორითი

}

axios.get("https://www.youtube.com/")

     .then((getVideo) =>{

    })

     .catch((errodLoad) =>{

    })

console.log("bla bla bla")

სანამ საიტზე შევალთ და აქსიოსის მოქმედებები განხორციელდება კონსოლი მაინც დაიწერება

ეს არის აქსიოსი ფუნქციონალი დაუმეტესად ასე არ ვიყენებთ მას. აქსიოსი გამოიყენება ბექენდთან დასაკავშირებლად და რამის გასატანებლად ან მისაღებად.

ანუ ინფორმაციის გაცვლა გამოცვლისთვის გამოიყენება axios და ჯეისონით ვხელმძღვნალელობთ მასზე (ჯეისონს სტრინად მოაქვს მონაცემები)

API გვეხმარება ბრაუზერთან ან საიტთან კომუნიკაციაში. მაგალითად ტელევიზორის ეიპიაი არის პულტი რითიც ვცვლით არხებს ხოლო ბრაუზერის ეიპიაით საქონლელზე ინფორმაციის მიღება ან რაიმე მსგავსი