

## Վոաջադրանքներ 07: setTimeout և setInterval`

---

Վոաջադրանք 1: Ունենք getRandomNumber(a = 1, b = 2) ֆունկցիան, որը ստանում է 2 պարամետր և յուրաքանչյուր 1000 միլիվայրկյան ինտերվալը(setInterval) մեկ գեներացնում է պատահական թիվ (Math.random()), եթե գեներացված թիվը գտնվում է a-ից b միջակայքում ապա Promise-ի resolve-ի միջոցով վերադարձնում է (return) արդյունքը:

```
async function Spinner(a, b) {
  ֆունկցիայի մեջ անհրաժեշտ է 5 անգամ կանչել
  getRandomNumber(a, b); ֆունկցիան, և վերջում համեմատել վերադարձած արժեքները, եթե
  բոլորը արժեքները միջակայք հավասար են, ապա alert-ի կամ document.write-ի միջոցով էկրանին
  ցույց տալ "WIN!!!!", եթե ոչ` "LOSE!!!!".
}
```

Spinner(1,1) -ի դեպքում հաղթելու հավանականությունը կլինի 100%, իսկ Spinner(1,2) -ի դեպքում` 6.25%

---

Վոաջադրանք 2: Ստեղծել սուրճի մեքենայի իմիտացիա, որը կստանա պատվերներ սարքերի սեսակի սուրճերի համար և կցուցադրի պատասխան գործընթացը ֆայլ առ ֆայլ:

Օգտագործողը կարող է պատվիրել երեք սեսակի սուրճ` espresso, latte կամ cappuccino: Եթե օգտվողը փորձի պատվիրել անթույլատրելի սեսակի սուրճ (օրինակ, "tea"), սուրճի մեքենան պետք է լուրջ կոդերի դրա մասին:

Ստեղծել makeCoffee(coffeeType) ֆունկցիա, որը ստանում է մեկ փոփոխական` սուրճի սեսակը(անունը): Եթե սուրճի սեսակը անթույլատրելի է, ֆունկցիան պետք է վերադարձնի սխալ: Օգտագործել console.log() հաղորդագրությունները ցուցադրելու համար:

Ցուցադրել սուրճ պատրաստելու ֆայլերը, ինչպիսիք են` "Տեղադրում ենք բաժակը", "Լցնում ենք ջուր", "Վվելացնում ենք սուրճ", "Վվելացնում ենք շաքարավազ", "Խառնում ենք": Պատվերները պետք է իրականացվեն սինխրոն: Անհրաժեշտ է օգտագործել async/await ափսոսանք գործողությունները կառավարելու համար:

Հուշում` վերհիշել Promise((resolve, reject) => {})

Ենթ: <https://prnt.sc/6lfd6NFlhYX8>

// espresso-ն պատրաստում են առանց շաքարավազի!!!