

JS Pagrindai

- 1. Kintamieji ir šabloninės eilutės:** Sukurk programą, kuri imituoja pirkinių čekio generavimą. Priima tris kintamuosius: prekės pavadinimą (pvz., „Pienas“), kiekį (pvz., 2) ir vieneto kainą (pvz., 1.5). Naudodamas šablonines eilutes, išvesk čekį, pvz., „Prekė: Pienas, kiekis: 2 vnt., vieneto kaina: 1.5 €, viso: 3 €“. Taip pat pridėk 21% PVM ir išvesk galutinę sumą su mokesčiais.

- 2. Sąlygos ir loginiai operatoriai:** Parašyk funkciją, kuri imituoja orų programėlės rekomendaciją. Ji priima tris parametrus: temperatūrą (pvz., 15°C), ar lyja (true/false), ir vėjo greitį (pvz., 20 km/h). Funkcija grąžina patarimą:
 - ✓ Jei temperatūra > 20°C ir nelyja – „Puikus oras pasivaikščiojimui!“
 - ✓ Jei lyja arba vėjo greitis > 30 km/h – „Geriau likti namuose.“
 - ✓ Kitais atvejais – „Apsirenk pagal orą ir viskas bus gerai!“

- 3. Ciklai ir skaičiavimai:** Sukurk programą, kuri imituoja automobilio kelionę ir apskaičiuoja, kiek kilometrų jis nuvažiuos per tam tikrą laiką. Naudok šiuos duomenis (duomenis galima keisti pagal pageidavimus):
 - Pradinė kuro talpa (pvz., 50 litrų) įvedama kaip kintamasis.
 - Mašina naudoja 7 litrus kuro 100 kilometrų
 - Mašina važiuoja pastoviu 90 km/h greičiu.
 - Naudodamas ciklą, apskaičiuok ir kiekvieną valandą išvesk:
 - ✓ Kiek kilometrų nuvažiuota iš viso.
 - ✓ Kiek kuro liko bake.
 - Ciklas sustoja, kai kuro lieka mažiau nei 7 litrai (neužtenka kitam 100 km).

Pavyzdinis išvesties formatas:

Po 1 val.: nuvažiuota 90 km, liko 43.7 l kuro

Po 2 val.: nuvažiuota 180 km, liko 37.4 l kuro

...

Po X val.: nuvažiuota Y km, liko Z l kuro – kelionė baigta!

4. **Masyvai ir atsitiktinumas:** Sukurk masyvą su 5 istoriniais ginklais (pvz., „kardas“, „kalavijas“ ir t.t.). Parašyk funkciją, kuri atsitiktinai parenka vieną ginklą (naudodamas `Math.random()`) ir išveda pranešimą, pvz., „Tau paskirtas [ginklas] kovai!“
-
5. **Objektai ir metodai:** Sukurk objektą, aprašantį piratą (vardas, laivas, lobio kiekis aukso monetomis). Pridėk metodą `raid`, kuris padidina lobio kiekį atsitiktiniu skaičiumi (pvz., nuo 10 iki 50 monetų) ir grąžina pranešimą, pvz., „[Vardas] apiplėšė ir gavo [kiekis] monetų!“

Funkcijos ir jų taikymas

6. **Funkcijos:** Parašyk funkciją, kuri priima slaptažodį (string) ir grąžina objektą su: `ilgis` (length), ar yra skaičių (true/false), ar yra didžiųjų raidžių (true/false) ir stiprumo įvertinimą („Silpnas“, „Vidutinis“, „Stiprus“ pagal kriterijus: `ilgis > 8`, skaičiai, didžiosios raidės).

```
console.log(checkPassword("abc"));
Išvestis: { length: 3, hasNumbers: false, hasUppercase: false, strength:
  "Silpnas" }
console.log(checkPassword("Pass123"));
Išvestis: { length: 7, hasNumbers: true, hasUppercase: true, strength:
  "Vidutinis" }
console.log(checkPassword("SuperSecure123"));
Išvestis: { length: 14, hasNumbers: true, hasUppercase: true, strength:
  "Stiprus" }
```

7. **Aukštesnio lygio funkcijos:** Sukurk funkciją, kuri priima masyvą su darbų objektais (kiekvienas turi `task` – darbo pavadinimą ir `done` – būseną kaip true/false) ir kitą funkciją, apibrėžiančią, kaip keisti būseną (pvz., pažymėti kaip atliktą). Grąžink naują masyvą su atnaujintais darbais.

Duomenų pvz:

```
const myTasks = [
```

```
    { task: "Išplauti indus", done: false },
    { task: "Pavalgyt", done: false },
  ];
  Išvestis: [{ task: "Išplauti indus", done: true }, { task: "Pavalgyt",
done: true }]
```

8. Aukštesnio lygio funkcijos: Sukurk funkciją, kuri priima masyvą su maisto kalorijomis (skaičiais) ir kitą funkciją, apibrėžiančią, kaip keisti kalorijas (pvz., pridėti 10%). Grąžink naują masyvą su atnaujintomis kalorijomis.

```
Duomenų pvz:
const dailyFoods = [200, 150, 300];
Išvestis: [220, 165, 330]
```

9. Numatyti parametrai + Masyvų metodai (reduce): Parašyk funkciją, kuri priima masyvą su pirkinių kainomis ir nuolaidą (su numatyta reikšme 0%), o grąžina bendrą sumą po nuolaidos.

```
Duomenų pvz:
calculateTotal([10, 20, 30]) => Išvestis: "60.00 eurų";
calculateTotal([10, 20, 30], 20) => Išvestis: "48.00 eurų"
```

10. Uždarymai (Closures) – moko funkcijų, kurios „prisimena“ ankstesnius duomenis.

Užduotis: Sukurk funkciją, kuri priima pradinį treniruočių dienų masyvą (pvz., ["Pirmadienis"]) ir grąžina kitą funkciją. Ta nauja funkcija leidžia pridėti vieną treniruočių dieną ir išveda visas esamas dienas kaip eilutę.

```
const myPlan = planWorkouts(["Antradienis"]);
console.log(myPlan("Ketvirtadienis"));
išvestis => Treniruotės: Antradienis, Ketvirtadienis
console.log(myPlan("Šeštadienis"));
išvestis => Treniruotės: Antradienis, Ketvirtadienis, Šeštadienis
```