

1. Funkcijos yra veiksmas arba veiksmų rinkinys atskiras nuo pagrindinės programos. Viena iš funkcijų pavyzdžių būtų pvz. kvadrato ploto apskaičiavimo funkcija, kuri gauna kvadrato kraštinės ilgį ir pagal jį apskaičiuoja plotą.
2. Funkcijos gali priklausyti kintamajui arba būti atskiros. Mes kursime ATSKIRAS funkcijas, tačiau patys kiekvienoje programoje naudojame jau kūrėjų sukurtas funkcijas, kurios priklauso kintamiesiems kaip kad pvz. `console.log()`;
3. `console` yra kintamasis o `log()` yra funkcija, todėl rašom visada pagal tokį šabloną:
`kintamojoVardas.funkcijosVardas(parametras1, parametras2, ..., parametrasN)`
 - a. Po kintamojoVardo dedame TAŠKĄ

```
console.log(plotas);
```

b.

4. Tačiau, kaip minėjau mes kursime pakol kas funkcijas, kurios NEPRIKLAUSO kintamajui, o tokias funkcijas reikia pradėti po `window.onload` PASKUTINIO riestinio skliausto.
5. Norint sukurti funkciją rašome `function pavadinimas(parametras1, parametras2) { veiksmas }`
 - a. pavadinimas – funkcijos pavadinimas
 - b. skliaustuose yra parametrai – jie atskirti kableliu
 - c. veiksmas tarp riestinių skliaustų – tai tiesiog kokius veiksmus reikėtų atlikti tai funkcijai.

```
function kvadratoPlotas(a) {  
  var plotas;  
  plotas = a * a;  
  return plotas;  
}
```

6. Norint funkciją panaudoti, reikia pagrindinėje programoje **kreiptis** į ją. Jeigu funkcija kažką grąžina (turi `return`) tai turime funkcijos reikšmę priskirti kažkokiam kintamajui:

```
7. perimetras = kvadratoPerimetras(briaunosIlgis);
```

8. Skliausteliuose tiesiog įrašome ką paduodame funkcijai vietoj jos parametru (šiuo atveju paduodame **briaunosIlgis** kintamąjį vietoj parametro **a**)
9. O tada su `perimetras` kintamuoju galime elgtis kaip su paprastu kintamuoju.
10. Funkcijų nauda yra tokia, jog tą pačią funkciją galime panaudoti begalybę kartų PAGRINDINĖJE PROGRAMOJE.