



Algoritmų parinkimas.

Ciklai

Ką išmoksime šiandien

- Funkcija range() (Python)
- Sakinys „pass“
- Ciklas cikle (nested loops)

Funkcija `range()` (Python)

- **Tikslas:** Funkcija `range()` naudojama generuoti seką skaičių, kuri dažnai naudojama `for` cikluose.
- **Veikimas:** `range()` gali būti iškviesta su vienu, dviem arba trimis argumentais:
 - `range(n)`: Sugeneruoja skaičius nuo **0 iki $n-1$**
 - `range(start, stop)`: Sugeneruoja skaičius nuo **start iki stop-1**
 - `range(start, stop, step)`: Sugeneruoja skaičius nuo **start** iki **stop-1**, su nurodytu **step** (žingsniu).

Python - range() pavyzdys

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

```
for i in range(5):  
    print(i)
```

Rezultatas: 0, 1, 2, 3, 4

```
for i in range(2, 6):  
    print(i)
```

Rezultatas: 2, 3, 4, 5

```
for i in range(1, 10, 2):  
    print(i)
```

Rezultatas: 1, 3, 5, 7, 9

Sakinys „pass“

- **Tikslas:** `pass` sakinyss naudojamas kaip tuščias veiksmas, kuris leidžia išlaikyti sintaksės teisingumą, kai reikia tuščios kodo vietos.
- **Veikimas:** Jis nieko neatlieka, bet gali būti naudojamas, kai struktūra (pvz., ciklas ar funkcija) reikalauja kodo, tačiau nenorime ten dar nieko rašyti.
- **Naudojamas,** kai norime sukurti ***tuščią ciklą***, ***sąlygą*** ar ***funkciją***, kuri bus pildoma vėliau, kad išvengtume klaidų programoje.

Sakinio „pass“ pavyzdys

Cikle / „if“ sakinyje

```
for x in [0, 1, 2]:  
    pass  
  
if b > a:  
    pass
```

Funkcijoje / Klasėje

```
def myfunction():  
    pass  
  
class Person:  
    pass
```

Ciklas cikle (nested loops)

- **Apibrėžimas:** Įdėti ciklai (nested loops) yra ciklai, kurie yra įterpti vienas į kitą. Išorinysis ciklas apima vidinį ciklą, kuris gali būti vykdomas kiekvienos išorinės iteracijos metu.
- **Veikimas:** Vidinis ciklas vykdomas kiekvieną kartą, kai išorinė iteracija prasideda. Tai leidžia apdoroti daugiamatę informaciją, pavyzdžiui, dvimatį masyvą.
- Ciklas cikle dažnai **naudojamas** dirbant su dvimačiais masyvais, matrica, ar kitais kompleksiniais duomenų struktūromis.

Ciklo ciklo pavyzdys

```
for i in range(3): # Išorinė ciklo iteracija
    for j in range(2): # Vidinė ciklo iteracija
        print(f"i: {i}, j: {j}")
```

Rezultatas:

```
i: 0, j: 0
i: 0, j: 1
i: 1, j: 0
i: 1, j: 1
i: 2, j: 0
i: 2, j: 1
```

```
for (int i = 0; i < 3; i++) { // Išorinė ciklo iteracija
    for (int j = 0; j < 2; j++) { // Vidinė ciklo iteracija
        cout << "i: " << i << ", j: " << j << endl;
    }
}
```