

Python Pagrindai: W02

- Kintamieji (Variables)
- Duomenų tipai
- Sąrašai (Lists)

Python Pagrindai: Kintamieji

Kintamasis – tai duomenų konteineris, kuriame programuotojas išsaugoja tam tikrą reikšmę (skaičių, tekstą, vartotojo vardą ir t.t), ir vėliau panaudoja išsaugotą informaciją programoje atsiradus poreikiui.

Python Pagrindai: Kintamieji

Kintamųjų pavyzdžiai:

```
user_name = "John Doe"
```

```
user_age = 32
```

```
user_height = 1.87
```

```
is_admin = False
```

```
is_active = True
```

Python Pagrindai: Kintamieji

Kintamųjų pavadinimo taisyklės

- Kintamojo vardas turi prasidėti raide arba pabraukimo ženklu (“_”)
- Kintamojo vardas gali būti sudarytas iš raidžių, pabraukimo ženklų ir skaičių
- Kintamojo vardas negali sutapti su Python rezervuotais žodžiais (Python keywords).

Python Pagrindai: Duomenų Tipai

Python duomenų tipai skirstomi į dvi dideles grupes:

- Primityvai
- Kolekcijos

Python Pagrindai: Duomenų Tipai

Primityvų grupei priklauso:

- Sveiki skaičiai (Integers)
- Skaičiai su kableliu (Floats)
- Simbolių eilutės (Strings)
- Loginės (Būlio) išraiškos (Booleans)

Python Pagrindai: Duomenų Tipai

Kolekcijų grupei priklauso:

- Sąrašai (Lists)
- Įrašai (Tuples)
- Žodynai (Dictionaries)
- Aibės (Sets)

Python Pagrindai: Lists

Sąrašo sukūrimas:

```
students = ["John", "Jim", "Tony"]
```

```
temperatures = [12.5, 8.0, 10.8, 10.5, 11.2]
```


Python Pagrindai: Lists

Sąrašo indeksavimas:

- Indeksavimas prasideda nuo 0.
- Parašykite paprastą programą ir įsitikinkite:

```
students = ["John", "Jim", "Tony"]  
print("Pirmas studentas:", students[0])  
print("Antras studentas:", students[1])  
print("Trečias studentas:", students[2])
```

Python Pagrindai: Lists

Naudingos sąrašo funkcijos:

- `len(some_list)`: grąžina sąrašo ilgį
- `max(some_list)`: grąžina sąrašo maksimumą
- `min(some_list)`: grąžina sąrašo minimumą
- `sum(some_list)`: grąžina sąrašo elementų sumą
- `sorted(some_list)`: grąžina naują sąrašą su elementais surūšiuotais eilės tvarką.

Praktinė Užduotis

1) Sukurkite sąrašą

```
student_scores = [98.95, 75.87, 89.35, 89.20, 63.50]
```

2) Išveskite maksimalų balų skaičių, panaudodajant max() funkciją

3) Išveskite mažiausią balų skaičių, panauduoiant min() funkciją

4) Apskaičiuokite balų vidurkį, panauduoiant funkcijas len() ir sum()

5) Išveskite į ekraną balų sąrašą, surūšiuotą eilės tvarką (pradedant nuo mažiausio balo), panauduoiant sorted() funkciją.

Python Pagrindai: Lists

Ciklo operatorius **for**

naudojamas tam, kad pritaikyti kiekvienam sąrašo elementui tam tikrą operaciją arba veiksmą.

Veismų pavyzdžiai:

- Sąrašo elementų spausdinimas
- Funkcijos reikšmės apskaičiavimas sąrašo elemento reikšmei
- Sąrašo elemento išsaugojimas duomenų bazėje

Python Pagrindai: Lists

Pavyzdys 1: Specialybės dalykų išvedimas

```
subjects = ["Algebra", "Computer Science", "Physics"]  
for subj in subjects:  
    print(subj)
```

Pavyzdys 2: Specialybės dalykų išvedimas

```
subjects = ["Algebra", "Computer Science", "Physics"]  
for idx in len(subjects):  
    print(subjects[idx])
```

Pavyzdys 3: Specialybės dalykai su eilės numeriais:

```
subjects = ["Algebra", "Computer Science", "Physics"]  
for subj_no, subj_name in enumerate(subjects, start=1):  
    print(f"{subj_no}. {subj_name}")
```

Praktinė Užduotis

Parašykite Python programa, kurioje turite

- a) Sukurti grupiokų sąrašą
- b) ir atspausdinkite visus studentus eilės tvarka, nurodant eilės numerį kartu su grupioko vardu.

Pavyzdys:

1. Tomas
2. Marius
3. Lina