




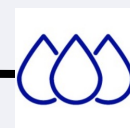


Nieko bendro su Lietuvos prezidentu.
Tiesiog **Aurimas Aleksandras** arba
Aurimas.
Python nuo **2013** .

 aurimas13
 aurimasaleksandras
 aurimasnausedas
 aurimasnausedas
 aurimas.nausedas@gmail.com



C&R Software
Collections & Recovery Simplified



Kas kas zoologijos sode?

Jūsų laikas spindėti!



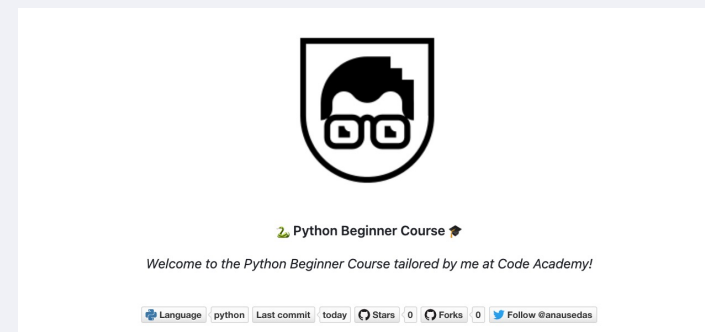
Klausimai:

- Jei galėtum būti bet koks pasaulio gyvūnas, koks Jūs būtumėte ir kodėl?
- Kas juokingiausia Jums kas nutiko per pastarąją savaitę?
- Jei būtumėte superherojus, kokia būtų jūsų supergalia ir kodėl?
- Koks keisčiausias jūsų kada nors išbandytas maistas ir koks jis buvo?
- Jei dabar galėtumėte teleportuotis į bet kurią pasaulio vietą, kur ir kodėl?



Kurso **GitHub** repozitorija kur bus talpinami uždaviniai/špargalkės/patarimai/papildoma medžiaga ir kitokios su kursu susijusios paskaitos atskirai nuo **Teams** paskaitų -

<https://github.com/aurimas13/Python-Beginner-Course>



Prieš pasinerdami daugiau į Python paskaitas ir uždavinius perskaitykite šituos patarimus, kuriuos rasite aukčiau pasidalintoje **GitHub** repozitorijoje, kad kuo greičiau pavyktų išmokti Python kalbą -

https://github.com/aurimas13/Python-Beginner-Course/blob/main/Tips/happy_learning.md



**Code
Academy**



1 LYGIS

1 paskaita. Kintamieji, if sąlyga

2023

Aurimas Aleksandras Nausėdas



Šiandien išmoksite

01

Kokie yra kintamųjų tipai, jų sudarymo taisyklės

02

Veiksmai su simbolių eilutėmis (string)

03

Konvertuoti skirtingus duomenų tipus

04

Priimti duomenis iš vartotojo konsolės

05

Sąlygos sakinius

06

Kodo komentavimo taisyklės



```
a = 5
print(a)

# 5

a = int(5)
print(a)

# 5

a = 7
print(a)

# 7

a = 10
print(a)

# 10

a = a - 4
print(a)

# 6
```

Sveikieji skaičiai – Integer (int)



```
a = 8.56  
b = 5  
c = a + b  
print(c)
```

```
# 13.56
```

```
a = float(5)  
print(a)
```

```
# 5.00
```

Skaičiai su kableliu – float



Operatorius Aprašymas

+	Sudėtis
-	Atimtis
*	Daugyba
/	Dalyba
%	Liekana (atlikus dalybą)
//	Sveikoji dalis (atlikus dalybą)
⌈ ⌋	Skaičiaus pakėlimas laipsniu

Matematinių veiksmų operatoriai



```
a = 5 + 2  
print(a)
```

```
# 7
```

```
b = 5 - 2  
print(c)
```

```
# 3
```

```
c = 5 * 2  
print(c)
```

```
# 10
```

```
b = 5 / 2  
print(c)
```

```
# 2.5
```

Veiksmai su kintamaisiais



Kintamųjų pavadinimų sudarymo taisyklės

Kintamųjų pavadinimai turi prasidėti raide arba pabraukimu, pvz:

- `_vardas`
- `vardas`

Likusioji kintamojo dalis gali būti sudaryta iš raidžių, skaičių ir pabraukimų:

- `pirmas1`
- `antras_skaicius`
- `_e5786`

Pavadinimuose svarbios didžiosios ir mažosios raidės:

- *Vardas* ir *vardas* būtų skirtingi kintamieji.

Kintamaisiais negali būti python raktiniai žodžiai:

'False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield'

Python kalboje sudurtinius kintamųjų pavadinimus priimta sudarinėti taip:

- `first_block`, `vandens_temperatura`



```
a = 5  
a += 2  
print(a)
```

```
# 7
```

```
b = 12  
b /= 3  
print(b)
```

```
# 4.0
```

Paprastesnis veiksmų atlikimas



```
a = 2**2  
print(a)
```

```
# 4
```

```
b = 5**3  
print(b)
```

```
# 125
```

Kėlimas laipsniu



```
a = 32 / 6
print(a)

# 5.333333333333333

b = 32 // 6
print(b)

# 5

c = 32 % 6
print(c)

# 2
```

**Sveikojo skaičiaus ir
liekanos paieška (div/mod)**



```
zodis1 = "Labas "  
print(zodis1)  
  
# Labas  
  
zodis2 = str("vakaras")  
print(zodis2)  
  
# vakaras  
  
print(zodis1 + zodis2)  
  
# Labas vakaras
```

Simbolių eilutės (String) tipas



```
print("Labas \nvakaras")
```

```
# Labas
```

```
# vakaras
```

Nauja eilutė



C	o	d	e		A	c	a	d	e	m	y
0	1	2	3	4	5	6	7	8	9	10	11

```
zodis = "Code Academy"

print(zodis[5])

# A

print(zodis[-2])

# m

print(zodis[5:12])

# Academy

print(zodis[5:])

# Academy
```

Veiksmai su simbolių eilėmis (String) 1



C	o	d	e		A	c	a	d	e	m	y
0	1	2	3	4	5	6	7	8	9	10	11

```
print(zodis[:4])  
  
# Code  
  
print(zodis[5:12:1])  
  
# Academy  
  
print(zodis[5::2])  
  
# Aaey  
  
print(zodis[::-1])  
  
# ymedacA edoC
```

Veiksmai su simbolių eilėmis (String) 2



C	o	d	e		A	c	a	d	e	m	y
0	1	2	3	4	5	6	7	8	9	10	11

```
print(zodis.split())  
  
# ['Code', 'Academy']  
  
print(zodis.upper())  
  
# CODE ACADEMY  
  
print(zodis.replace('c', 'k'))  
  
# Code Akademy  
  
print(zodis.replace('Code', 'Music'))  
  
# Music Academy
```

Veiksmai su simbolių eilėmis (String) 3



```
a = 5

zodis = "Labas"
dar_vienas = "Šitas žodis"

print("a lygu: " + str(a) + ", žodis: " + zodis + ", dar vienas žodis - " + dar_vienas)

# Geresnis variantas:
print(f"a lygu {a}, žodis: {zodis}, dar vienas žodis - {dar_vienas}")
```

Geras būdas formuoti stringus iš kintamųjų



```
d = "Žodis "  
e = 5  
print(d+e)  
  
# TypeError: can only concatenate str (not "int") to str  
  
e = str(e)  
print(d+e)  
  
# Žodis 5  
  
a = "250"  
b = 4  
print(a * b)  
  
# ???
```

Veiksmai su skirtingais tipais (konvertavimas)



```
a = input("Įveskite pirmą žodį ")
b = input("Įveskite antrą žodį ")
print("Jūsų sakinys: ", a + b)

# Įveskite pirmą žodį Python
# Įveskite antrą žodį programavimas
# Jūsų sakinys: Python programavimas
```

String kintamųjų įvedimas ir išvedimas

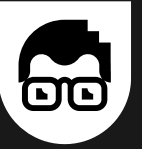


```
a = int(input("Įveskite pirmą skaičių "))
b = int(input("Įveskite antrą skaičių "))
print("Jūsų skaičių suma: ", a + b)
```

```
# Įveskite pirmą skaičių 5
# Įveskite antrą skaičių 6
# Jūsų skaičių suma: 11
```

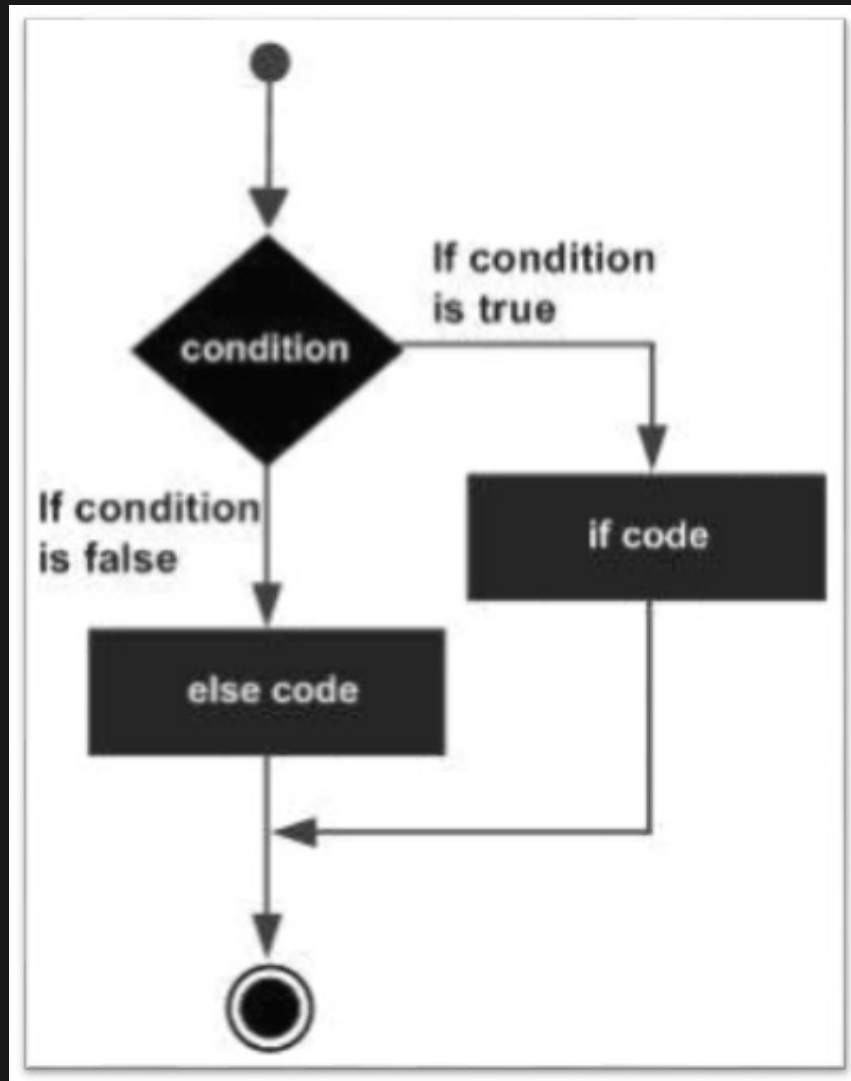
```
h = float(input("Įveskite skaičių "))
print(h)
```

Integer, float kintamųjų įvedimas ir išvedimas



Loginiai operatoriai

Operatorius	pavadinimas	pavyzdys	rezultatas
=	priskyrimas	x = 5, y = "Žodis"	–
==	lygu	5 == 5	True
!=	nelygu	8 != 8	False
and	ir	5 == 5 and 8 == 5	False
or	arba	7 == 7 or 3 == 6	True
>	daugiau	5 > 0	True
<	mažiau	8 < 0	False
>=, <=	daugiau arba lygu	8 >= 8, 5 <= 0	True, False



IF sąlygos taikymas



```
if 5 > 0:
    print("5 yra daugiau už 0")
# 5 yra daugiau už 0
if 5 < 0:
    print("5 yra daugiau už 0")
print("Programa baigta")

# Programa baigta
```

**Jeigu (IF) [sąlyga], tuomet
[veiksmai]**



```
skaicius = 25
if skaicius < 100:
    print("1: Skaičius yra mažesnis už 100")
if skaicius > 10:
    print("2: Skaičius yra didesnis už 10")
if skaicius < 10:
    print("3: Skaičius yra mažesnis už 10")

# 1: Skaičius yra mažesnis už 100
# 2: Skaičius yra didesnis už 10

skaicius = 60
if skaicius < 70:
    print("Skaičius yra mažesnis už 70")
    if skaicius > 15:
        print("Skaičius yra tarp 15 ir 70")

# Skaičius yra mažesnis už 70
# Skaičius yra tarp 15 ir 70

skaicius = 10

# Skaičius yra mažesnis už 70
```

**Jeigu (IF) [sąlyga], tuomet
[veiksmas]**



```
skaicius = 56
if skaicius == 50:
    print("1: Skaičius yra lygus 50")
else:
    print("2: Skaičius nelygus 50")

# 2: Skaičius nelygus 50
```

Sąlyga ELSE (jei ne, tuomet)



```
skaicius = 0

if skaicius > 0:
    print("Teigiamas skaičius")
elif skaicius == 0:
    print("Nulis")
else:
    print("Neigiamas skaičius")

# Nulis
```

Sąlyga ELIF (jei sąlyga netenkinama ir jei)



```
skaicius = -5

match skaicius:
    case skaicius if skaicius > 0:
        print('Teigiamas skaičius')
    case 0:
        print('Nulis')
    case _:
        print('Neigiamas skaičius')
```

Sąlyga match case (nuo Python 3.10 versijos)

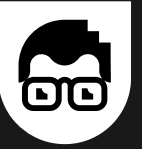


Kodo komentavimas

```
# Ši eilutė nebus vykdoma
```

```
"""  
Sveiki, draugai,  
Ši pastraipa nebus vykdoma.  
"""
```

- Komentuota eilutė (PyCharm programoje – CTRL+//)
- Komentuota pastraipa (Doctrings)



Užduotis nr. 1

Parašyti programą, kuri:

- Leistų įvesti skaičius a ir b (int arba float)
- Išvestų į ekraną „a mažesnis už b“, jei taip yra
- Išvestų į ekraną „a lygu b“, jei taip yra
- Išvestų į ekraną „a didesnis už b“, jei taip yra

Patarimas: naudoti if, elif, else sąlygas



Užduotis nr. 2

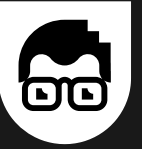
Parašyti programą, kuri su eilute "Zen of Python" darytų šiuos veiksmus:

- Atspausdintų paskutinį antro žodžio simbolį
- Atspausdintų pirmą trečio žodžio simbolį
- Atspausdintų tik pirmą žodį
- Atspausdintų tik paskutinį žodį
- Atspausdintų visą frazę atbulai
- Atskirtų žodžius ir juos atspausdintų
- Žodį "Python" pakeistų į "Programming" ir atspausdintų naują sakinį

Patarimas: naudoti string karpymo įrankius, funkcijas `split()`, `replace()`

The Zen of Python:

```
import this
```



Užduotis nr. 3

Parašyti programą, kuri su eilute "Zen of Python" darytų šiuos veiksmus:

- Atspausdintų paskutinį antro žodžio simbolį
- Atspausdintų pirmą trečio žodžio simbolį
- Atspausdintų tik pirmą žodį
- Atspausdintų tik paskutinį žodį
- Atspausdintų visą frazę atbulai
- Atskirtų žodžius ir juos atspausdintų
- Žodį "Python" pakeistų į "Programming" ir atspausdintų naują sakinį

Patarimas: naudoti string karpymo įrankius, funkcijas `split()`, `replace()`



Užduotis nr. 4

Programoje išbandyti daugiau string funkcijų:

- upper()
- casefold()
- capitalize()
- count()
- find()
- ir t.t.

Visas jas galite rasti čia: https://www.w3schools.com/python/python_ref_string.asp

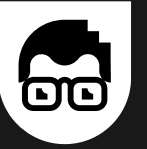


Užduotis nr. 5

Parašyti programą, kuri:

- Leistų įvesti pirmą skaičių
- Leistų įvesti antrą skaičių
- Paklaustų, kokį matematinį veiksmą reiktų atliktų
- Atspausdintų rezultatą: pasirinktų skaičių suma, daugybą ar pan.

Patarimas: naudoti input(), if, print



Namų darbas

Užbaigti klasėje nepadarytas užduotis



Išspręsti paskaitos uždaviniai (įkelti ketivrtadienį)

<https://github.com/aurimas13/Python-Beginner-Course/tree/main/Programs>

String functions

Visos built in string funkcijos

https://www.w3schools.com/python/python_ref_string.asp

**Naudinga
informacija**