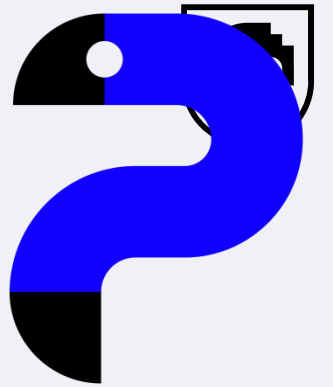




**Code
Academy**



1 LYGIS

3 paskaita. Boolean, data, laikas, išimtys

2023

Aurimas Aleksandras Nausėdas



Šiandien išmoksite

01

Boolean kintamuosius

02

Veiksmus su data ir laiku

03

Sugaudyti programos klaidas ir atlikti veiksmus



```
teisybe = True
print(teisybe)
# True

teisybe2 = False
print(teisybe2)
# False

loginis = bool()
print(loginis)
# False

print(1 == 1 or 2 == 2)
# True

print(2 < 1 or 3 > 6)
# False

logika = 1 == 1 or 2 == 2
print(logika)
# True

atvirkstine = not 2 * 2 == 4
print(atvirkstine)
# False
```

Loginis kintamasis (Boolean)

Dvi reikšmės = True (tiesa) arba False (klaidinga)



```
sarasas = ["Vienas", "Du", "Trys"]
skaicius = 123
kablelis = 5.56
zodynas = {"Mantas": 20}
loginis = True
```

```
print(type(sarasas))
print(type(skaicius))
print(type(kablelis))
print(type(zodynas))
print(type(loginis))
```

```
# <class 'list'>
# <class 'int'>
# <class 'float'>
# <class 'dict'>
# <class 'bool'>
```

Kaip patikrinti kintamojo tipą (funkcija type)



```
sarasas = [5, 8, "Lietuva", 95, "Žodis", True]

suma = 0

for x in sarasas:
    if type(x) is int:
        suma += x

print(suma)

# 108
```

**Programa, kuri paskaičiuoja ir
atspausdina sąrašo skaičių
(sveikų) sumą**



```
import datetime

x = datetime.datetime.today()
print(x)

# 2020-11-25 12:41:47.651313
```

Data, laikas (Datetime)

Datetime kintamasis gali išsaugoti datą ir/arba laiką.
Jis importuojamas per *import datetime*

Dabartinė data ir laikas



```
import datetime

x = datetime.date.today()
print(x)

# 2020-11-25
```

today() - Tik data



```
import datetime
```

```
x = datetime.datetime.today().time()  
print(x)
```

```
# 12:43:45.571871
```

time() - Tik laikas



```
import datetime
y = datetime.datetime(2020, 2, 29, 18, 20, 50)
print(y)
# 2020-02-29 18:20:50

print(y.strftime("%A, %d. %B %Y %I:%M%p"))
# Saturday, 29. February 2020 06:20PM
```

Datos formatavimas

<https://docs.python.org/3/library/datetime.html>



```
import datetime
now = datetime.datetime.now()
print(now)
print(now - datetime.timedelta(days=5))
print(now + datetime.timedelta(hours=5))
print(now + datetime.timedelta(days=20, hours=8))
```

```
# 2020-11-25 12:26:14.575023
# 2020-11-20 12:26:14.575023
# 2020-11-25 17:26:14.575023
# 2020-12-15 20:26:14.575023
```

Kaip pridėti ar atimti laiką



```
import datetime
ivesta_data = input("Įveskite datą: ")
data = datetime.datetime.strptime(ivesta_data, "%Y-%m-%d %H:%M:%S")
skirtumas = (datetime.datetime.now() - data)
print(skirtumas.days)
```

```
# Įveskite datą: 2004-03-29 00:00:00
# 5604
```

Kaip įvesti datą/laiką



```
import datetime

now = datetime.datetime.today()

print(now.year)
print(now.month)
print(now.weekday())
print(now.day)
print(now.hour)
print(now.minute)
print(now.second)
print(now.microsecond)

# 2020
# 11
# 2
# 25
# 12
# 13
# 45
# 760594
```

**Kaip iš datetime atskirai ištraukti
metus, mėnesį, valandas**



```
7 / 0
# ZeroDivisionError: division by zero

skaicius = int(input("Įveskite skaičių: "))
# Įveskite skaičių: k11
# ValueError: invalid literal for int() with base 10: 'k11'
```

```
dalinyys = 7
daliklis = 0
if daliklis == 0:
    print("Dalyba iš nulio negalima")
else:
    dalinyys / daliklis
print("Programa vykdoma toliau")

# Dalyba iš nulio negalima
# Programa vykdoma toliau
```

Išimtys, jų suvaldymas (su try/except/finally)



```
try:
    7 / 0
except:
    print("Dalyba iš nulio negalima")

# Dalyba iš nulio negalima

try:
    skaicius = int(input("Įveskite skaičių: "))
except:
    print("Įvestas klaidingas skaičius")

# Įveskite skaičių: k11
# Įvestas klaidingas skaičius

try:
    open('file.txt')
except:
    print("Nepavyksta atidaryti failo")

print("Programa vykdoma toliau")

# Nepavyksta atidaryti failo
# Programa vykdoma toliau
```

Klaidų suvaldymas naudojant try/except

Kuo naudingas try/except/finally naudojimas:

- Leidžia pakeisti klaidų pranešimus norimu tekstu
- Įvykus klaidai, programa nesustoja (apsaugo nuo lūžimo). Po neįvykdyto kodo, programa vykdoma toliau
- Leidžia nuspręsti, ką daryti, atsiradus klaidai (pvz., išmesti tam tikrą pranešimą, paleisti kitą funkciją ir t.t)



```
try:
    skaicius = int(input("Įveskite skaičių: "))
    print(7 / skaicius)
    open('file.txt')
except ZeroDivisionError:
    print("Dalyba iš nulio negalima")
except ValueError:
    print("Įvestas klaidingas skaičius")
except FileNotFoundError:
    print("Nepavyko atidaryti failo")

# Įveskite skaičių: k11
# Įvestas klaidingas skaičius

# Įveskite skaičių: 0
# Dalyba iš nulio negalima

# Įveskite skaičių: 7
# 1.0
# Nepavyko atidaryti failo
```

Kaip suvaldyti kelias išimtis



```
try:
    print(7 / 0)
except:
    print("Dalyba iš nulio negalima")
finally:
    print("Todėl įvykdysime daugybą: ")
    print(7 * 7)
print("Programa vykdoma toliau")
```

```
# Dalyba iš nulio negalima
# Todėl įvykdysime daugybą:
# 49
# Programa vykdoma toliau
```

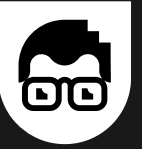
Finally blokas

Kodas, vykdomas nepaisant to, kas įvyksta try/except blokuose



```
while True:
    try:
        x = int(input("Įveskite skaičių: "))
        break
    except ValueError:
        print("Įvedėte ne skaičių. Bandykite dar kartą")
```

Kaip panaudoti try/except įvedant duomenis

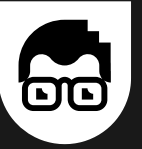


Užduotis nr. 1

Parašyti programą, kuri:

- Leistų vartotojui įvesti sveiką skaičių.
- Atspausdinti True, jei skaičius teigiamas
- Atspausdinti False, jei skaičius neigiamas ar lygus 0
- True/False reikšmei išsaugoti naudoti boolean tipo kintamąjį `ar_skaicius_teigiamas`

Patarimas: naudoti input, boolean, if/else

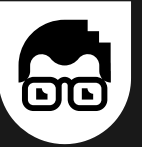


Užduotis nr. 2

Parašyti programą, kuri:

- Atspausdintų dabartinę datą ir laiką
- Atimtų iš dabartinės datos ir laiko 5 dienas ir juos atspausdintų
- Pridėti prie dabartinės datos ir laiko 8 valandas ir juos atspausdintų
- Atspausdintų dabartinę datą ir laiką tokiu formatu: 2019 03 08, 09:57:17

Patarimas: naudoti datetime, timedelta (from datetime import timedelta)



Užduotis nr. 3

Parašyti programą, kuri:

- Leistų vartotojui įvesti norimą datą ir laiką (pvz. gimtadienį)
- Paskaičiuotų ir atspausdintų, kiek nuo įvestos datos ir laiko praėjo:
- Metų
- Mėnesių
- Dienų
- Valandų
- Minučių
- Sekundžių
- Kadangi tiksliai galima paskaičiuoti tik dienas ir sekundes, metus, mėnesius ir t.t. paskaičiuokite apytiksliai.

Patarimas: naudoti `datetime`, `.days`, `.total_seconds()`

Skaičių suapvalinimo pavyzdys (kurio gali prireikti šioje užduotyje):

```
skaicius = 4.66  
  
print(round(skaicius))
```



Užduotis nr. 4

Pakeisti 1 ir 3 užduotis taip, kad neteisingai įvedus duomenis ar įvykus klaidoms, programos mestų norimas klaidas lietuvių kalba (panaudoti try/except)



Namų darbas

Užbaigti klasėje nepadarytas užduotis



Exception list

Galimų klaidų sąrašas

<https://docs.python.org/3/library/exceptions.html>

Datetime

Plačiau apie datetime biblioteką

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

Boolean

Plačiau apie boolean duomenų tipą

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

**Naudinga
informacija**