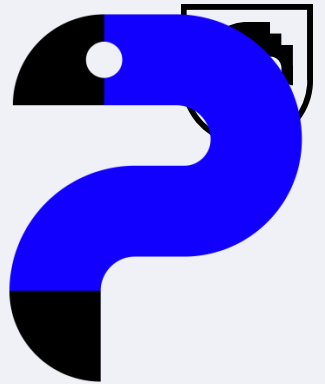




**Code  
Academy**



1 LYGIS

# **17 paskaita. Virtuali aplinka, paleidžiamieji failai**



# Šiandien išmoksite

01

Sukurti virtualią darbinę aplinką

02

Instaliuoti norimas bibliotekas į virtualią aplinką

03

Kurti EXE failus



```
C:\Users\Vartotojas>pip list
```

```
Package Version
```

```
-----
```

```
decorator 4.0.10
```

```
docutils 0.12
```

```
pefile 2018.8.8
```

```
pip 18.1
```

```
PyInstaller 3.4
```

```
pytz 2016.7
```

```
pywin32-ctypes 0.2.0
```

```
setuptools 40.6.2
```

## Virtuali aplinka (VENV)

Kaip pažiūrėti, kokie paketai (moduliai) įdiegti sistemoje:

Windows komandinėje eilutėje (Win + c, m, d + Enter):



```
C:\Users\Vartotojas>cd Desktop  
  
C:\Users\Vartotojas\Desktop>mkdir Projektas  
  
C:\Users\Vartotojas\Desktop>cd Projektas  
  
C:\Users\Vartotojas\Desktop\Projektas>python -m venv venv
```

## Kaip sukurti virtualią programavimo aplinką



```
C:\Users\Donoras\Desktop\Projektas>venv\Scripts\activate.bat  
(venv) C:\Users\Donoras\Desktop\Projektas>pip list
```

```
Package Version  
-----  
pip 18.1  
setuptools 40.6.2
```

```
(venv) C:\Users\Donoras\Desktop\Projektas>python sveiki.py  
(venv) C:\Users\Donoras\Desktop\Projektas>deactivate
```

## Kaip aktyvuoti/deaktyvuoti virtualią programavimo aplinką



```
(venv) C:\Users\Donoras\Desktop\Projektas>pip install pyinstaller  
(venv) C:\Users\Donoras\Desktop\Projektas>pip list
```

```
Package Version  
-----  
altgraph 0.16.1  
future 0.17.1  
macholib 1.11  
pefile 2019.4.18  
pip 18.1  
PyInstaller 3.4  
pywin32-ctypes 0.2.0  
setuptools 40.6.2
```

## Kaip įdiegti paketus į virtualią programavimo aplinką

17 paskaita. Virtuali aplinka, paleidžiamieji failai



```
(venv) C:\Users\Donoras\Desktop\Projektas>pip freeze
```

```
altgraph==0.16.1  
future==0.17.1  
macholib==1.11  
pefile==2019.4.18  
PyInstaller==3.4  
pywin32-ctypes==0.2.0
```

```
(venv) C:\Users\Donoras\Desktop\Projektas>pip freeze > requirements.txt
```

## Kaip sukurti reikalavimų failą



```
(venv) C:\Users\Donoras\Desktop\Projektas>pip install -r requirements.txt
```

```
(venv) C:\Users\Donoras\Desktop\Projektas>pip list
```

```
Package Version
```

```
-----
```

```
altgraph 0.16.1
```

```
future 0.17.1
```

```
macholib 1.11
```

```
pefile 2019.4.18
```

```
pip 18.1
```

```
PyInstaller 3.4
```

```
pywin32-ctypes 0.2.0
```

```
setuptools 40.6.2
```

## Kaip įdiegti paketus iš reikalavimų failo





```
C:\Users\Donoras\Desktop\Projektas>python -m venv venv --system-site-packages
```

```
C:\Users\Donoras\Desktop\Projektas>venv\Scripts\activate.bat
```

```
(venv) C:\Users\Donoras\Desktop\Projektas>pip list
```

```
Package Version
```

```
-----
```

```
altgraph 0.16.1
```

```
Babel 2.3.4
```

```
decorator 4.0.10
```

```
docutils 0.12
```

```
future 0.17.1
```

```
macholib 1.11
```

```
pefile 2018.8.8
```

```
pip 18.1
```

```
PyInstaller 3.4
```

```
pytz 2016.7
```

```
pywin32-ctypes 0.2.0
```

```
setuptools 40.6.2
```

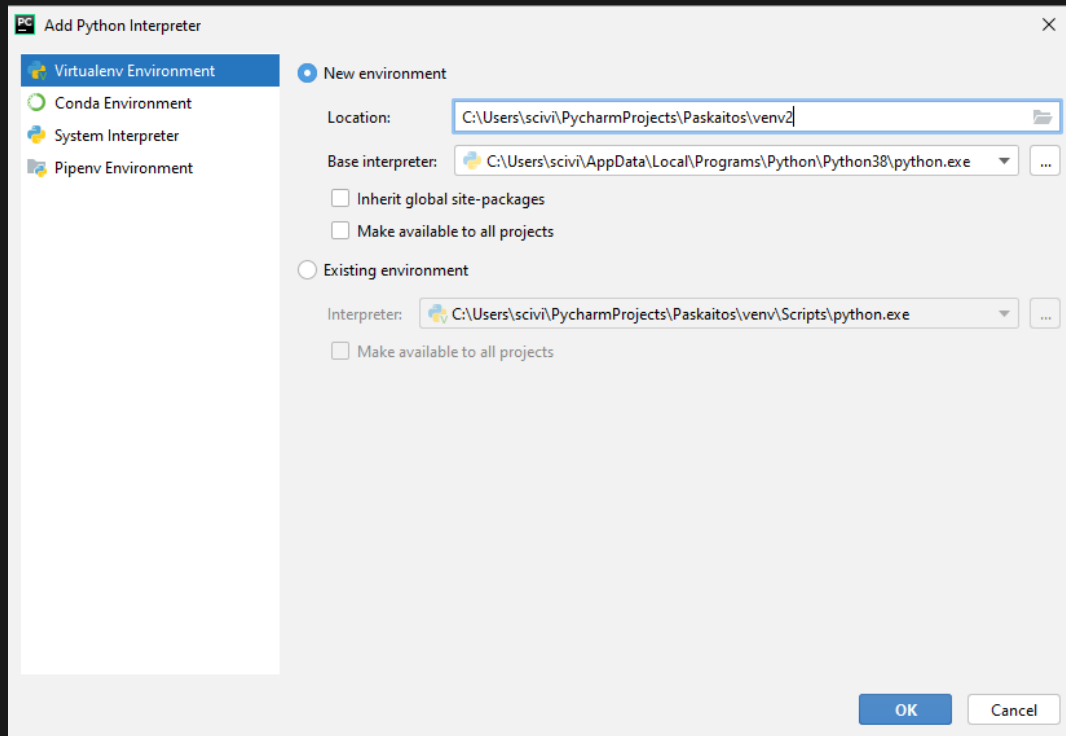
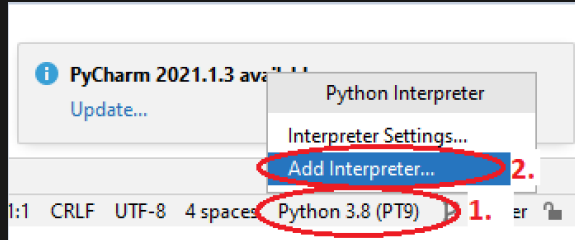
## Kaip sukurti virtualią aplinką su paketais iš sistemos aplinkos



```
(venv) C:\Users\Donoras\Desktop\Projektas>pip list --local
Package Version
-----
pip 18.1
setuptools 40.6.2
SQLAlchemy 1.3.3
```

## Kaip pažiūrėti tik virtualios aplinkos paketus

17 paskaita. Virtuali aplinka, paleidžiamieji failai



# Kaip sukurti virtualią aplinką PyCharm projekte



## VENV pastabos

- Į virtualią aplinką nededame jokių projekto (.py ir kitų) failų
- Virtualios aplinkos katalogo nededame į versijų valdymo sistemų repozitorijas
- Dedame requirements failą, kad kiti kodu besinaudojantys asmenys galėtų susikurti savo virtualias aplinkas



```
pyinstaller --onefile sveiki.py
```

## EXE failų kūrimas

Reikia turėti įsirašius PyInstaller biblioteką. Tada terminale paleidžiame šią eilutę, kurioje nurodomas argumentas "--onefile" (bus sukurtas 1 exe file) ir nurodomas python programos pavadinimas iš kurio norima padaryti exe failą



```
from tkinter import *  
  
pagrindinis = Tk()  
uzrasas = Label(pagrindinis, text="Sveikas,  
pasauli!")  
uzrasas.pack()  
pagrindinis.mainloop()
```

```
pyinstaller --onefile -w sveiki2.py
```

## Padarome EXE failą iš programos su grafine sąsaja

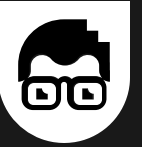


```
from tkinter import *

pagrindinis = Tk()
uzrasas = Label(pagrindinis, width=40, height=10, text="Sveikas, pasauli!")
pagrindinis.title("Mano programa")
pagrindinis.iconbitmap(r'sveikinimasis.ico')
uzrasas.pack()
pagrindinis.mainloop()
```

```
pyinstaller --onefile -w --icon=sveikinimas.ico sveikinimas.py
```

## Kaip pakeisti lango ikoną, pavadinimą, lango dydį



## Užduotis nr. 1

Išbandyti šioje pamokoje aprašytus žingsnius:

- Sukurti naują projektą su .py failu
- Jame sukurti virtualią aplinką
- Ją aktyvuoti
- Į ją įdiegti, pavyzdžiui, pyinstaller paketą
- Susikurti iš virtualios aplinkos requirements.txt failą

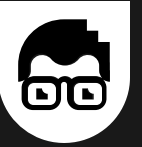




## Užduotis nr. 2

Sukurti paleidžiamąjį failą iš programos, kuri:

- Leistų vartotojui įvesti metus nuo ir metus iki
- Atspausdintų visus keliamuosius metus pagal duotą rėžį
- Paleidžiamasis failas turi turėti norimą ikoną



### Užduotis nr. 3

- Padaryti paleidžiamąjį failą iš 11 paskaitos 4 užduoties (pilna programa su vartotojo sąsaja)
- Programa turi turėti programos lango ikoną ir norimą pavadinimą
- Paleidžiamasis failas turi turėti norimą ikoną



## Namų darbas

Užbaigti klasėje nepadarytas užduotis



### **Ikonų biblioteka**

Nemokama ikonų biblioteka

<https://www.flaticon.com/>

### **Ikonų modifikavimas**

Nemokamas ikonų formatavimas

<https://icoconvert.com/>

# **Naudinga informacija**