

**Osnovna
manipulacija slika.
Pillow modul. Rad s
virtualnim
okruženjima.**



Pillow module

- U pythonu je na raspolaganju mnogo biblioteka za procesiranje digitalnih slika (OpenCV, scikit-image, matplotlib...)
- Jedan od osnovnih je Python Pillow Module (PIL)
- Instalirati ga na PC (u virtualno okruženje):
 - `npr. pip install pillow`
- Osnovna operacije je učitavanje slike:
 - `Image.open()`
- Ovo stvara `image` objekt koji ima razne metode (`show`, `rotate`, ...)

Zadatak 1

- Downloadajte `zadatak1.zip` koji sadrži:
 - `vehicle.jpg`
 - `zad1.py`
- Dodajte kod u skriptu `zad1.py` koji će pomoću biblioteke Pillow:
 - Omogućiti učitavanje slike `vehicle.jpg`
 - Ispišite rezoluciju slike u terminal
 - zarotirajte sliku za 90 stupnjeva suprotnom od smjera kazaljke na satu te ju prikažite
 - Spremanje zarotirane slike u datoteku naziva `vehicle_90.jpg`

(za izradu ovog zadatka instalirajte biblioteku Pillow)



Zadatak 1

- Modificirajte `zad1.py` tako da omogući:
 - Cropanje zarotirane slike (odbaciti crni dio koji se dobio prilikom rotacije)
 - Koristite metodu `crop`
 - Spajanje osnovne ulazne slike sa zarotiranom slikom
 - Koristite metodu `Image.new`

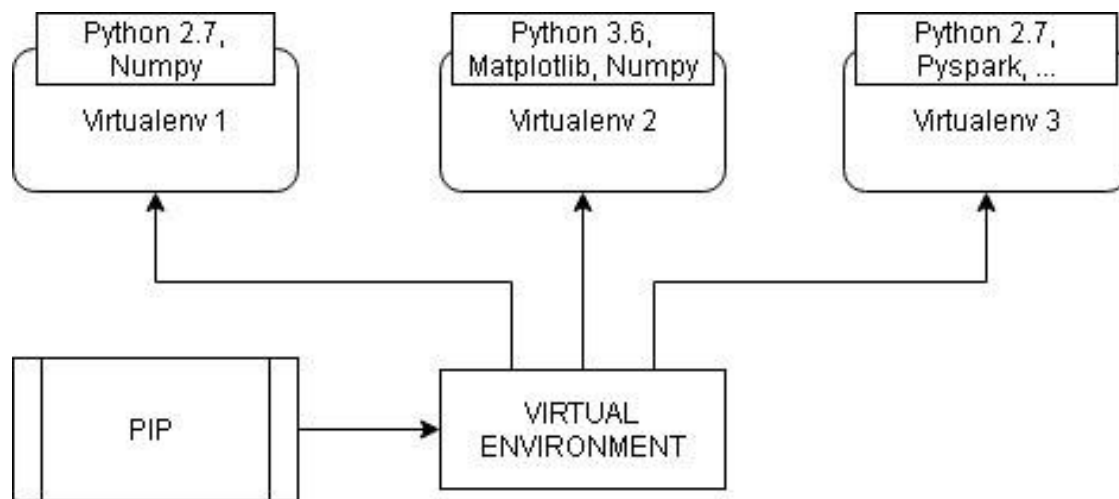


Zadatak 2

- Downloadajte `zadatak2.zip` koji sadrži direktorij `images`:
 - Tu su pohranje slike raznih naziva koje se nalaze u različitim (pod)direktorijima
 - Skriptu `zad2.py`
- dopunite skriptu `zad2.py` koja će:
 - pretražiti cijeli direktorij `images` te u terminal ispisati sve pronađene slike
 - svaku pronađenu sliku pretvoriti u grayscale i spremiti u `output` folder pod nazivom `img_x.jpg` pri čemu je `x` redni broj slike

Virtualna okruženja općenito

- skup određenih verzija biblioteka i Pythona
- zašto je to korisno?



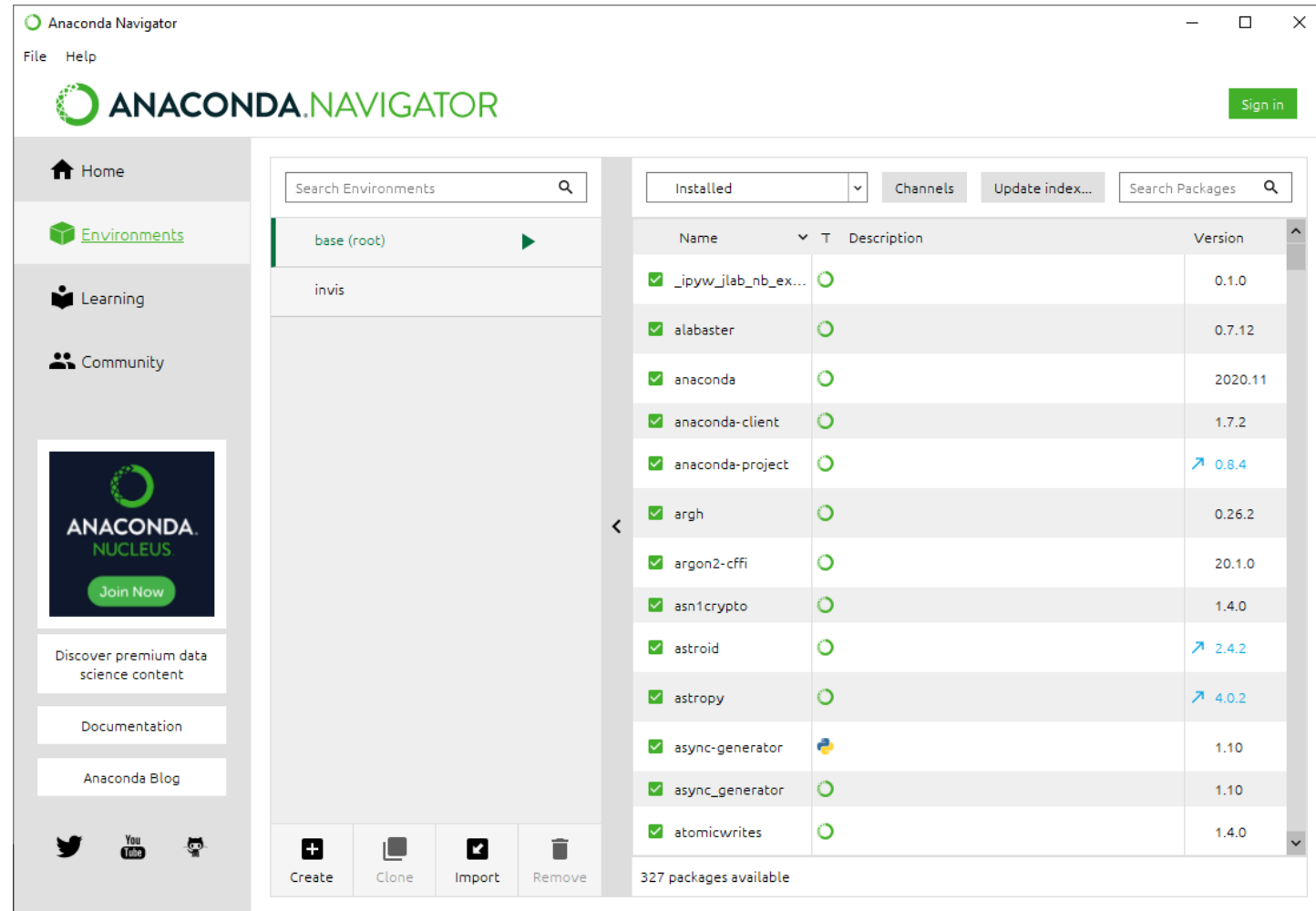
Anaconda distribucija

- Anaconda distribucija je vrlo popularna Pythona distribucija koja dolazi s velikim brojem paketa i sustavom za upravljanje instaliranim paketima i okruženjima
- Instalacija se može preuzeti na:
 - <https://www.anaconda.com/products/individual>
 - trenutna verzija Pythona je 3.8



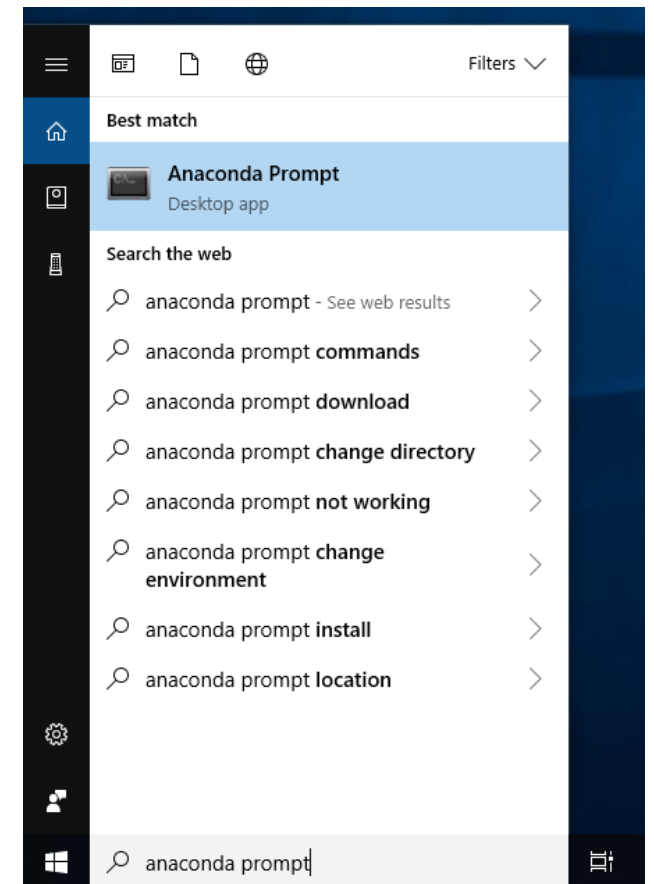
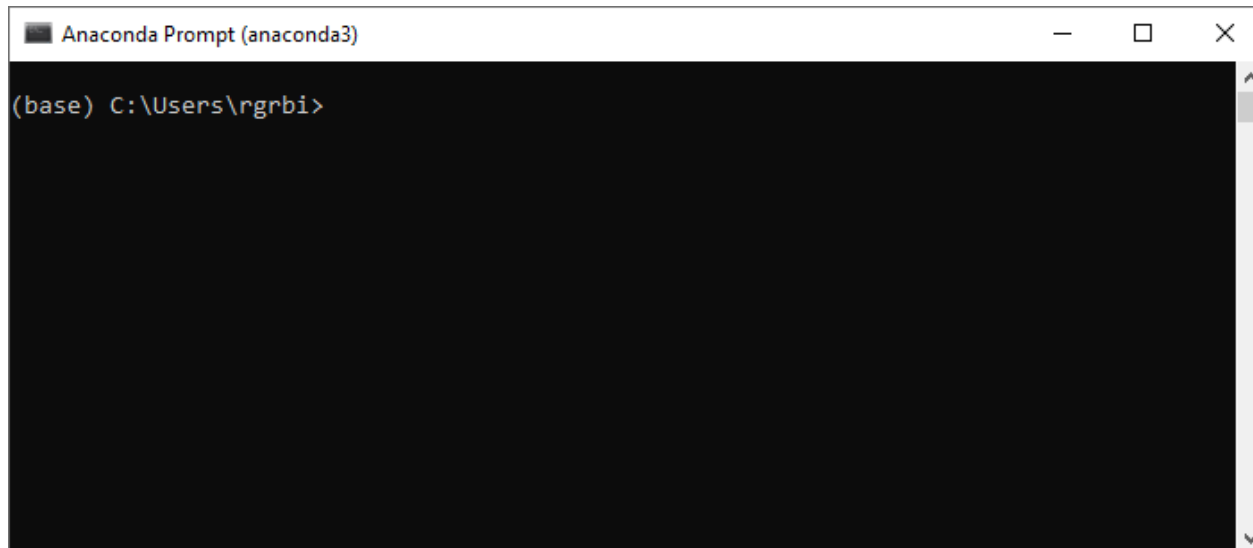
Anaconda navigator

- Anaconda navigator omogućuje upravljanje paketima, pokretanje IDE-a i slično
- kreiranje virtualnih okruženja moguće je pomoću taba Environments
- [Anaconda Navigator](#)



Anaconda prompt

- Kreiranje i instalaciju paketa moguće je napraviti i iz Anaconda Prompta
- naredbe započinju s `conda`
- [Getting started with conda](#)



Anaconda – rad s virtualnim okruženjima

- najprije deaktivirajte aktivno okruženje naredbom:
`$ conda deactivate`
- popis svih virtualnih okruženja dobiva se naredbom:
`$ conda info --envs`
- kreiranje novog virtualnog okruženja naziva „AIOpenCV”:
`$ conda create --name AIOpenCV python=3.8`
- aktiviranje kreiranog virtualnog okruženja
`$ conda activate LipikAI`

Anaconda – rad s virtualnim okruženjima

- brisanje virtualnog okruženja „my_env”
`$ conda env remove --name my_env`
- instalacija dodatnih paketa, npr. određene verzije OpenCV-a:
`$ conda install opencv=4.0.1`
- postoji i opcionalno instaliranje paketa pomoću naredbe pip
`$ pip install opencv-python`
- popis svih instaliranih paketa u trenutno aktivnom okruženju:
`$ conda list`
- ostale naredbe [Conda Cheat Sheet](#)

Koja je prednost korištenja virtualnog okruženja?

- s novijim verzijama nekada se mijenja API, potpisi funkcija i sl.
- oslobađaju programera mukotrpnog posla vezanog za instalaciju/deinstalaciju paketa i odgovarajućih verzija
- dovoljno je za određeni projekt napraviti virtualno okruženje s odgovarajućim paketima, za drugi projekt drugo okruženje itd.
- npr. funkcija `cv2.findContours()`:
 - OpenCV 3.x: vraća tri vrijednosti
 - OpenCV 4.x: vraća dvije vrijednosti

Koja je prednost korištenja virtualnog okruženja?

- virtualno okruženje se lako može dijeliti na način da se verzije paketa eksportaju u YAML datoteku:

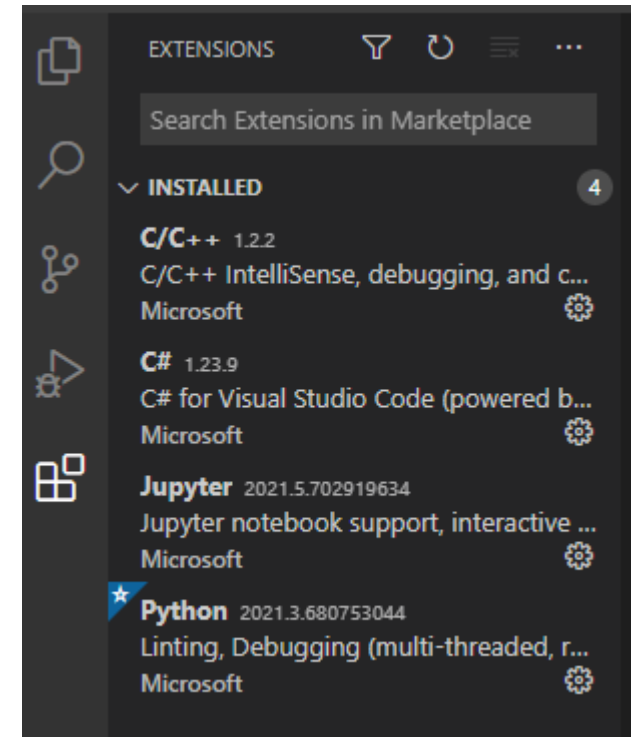
```
$ conda env export --name ENVNAME > envname.yml
```

- sada se ta datoteka može učitati na drugom računalu i automatski će se instalirati sve potrebne verzije biblioteka:

```
$ conda env create --file envname.yml
```

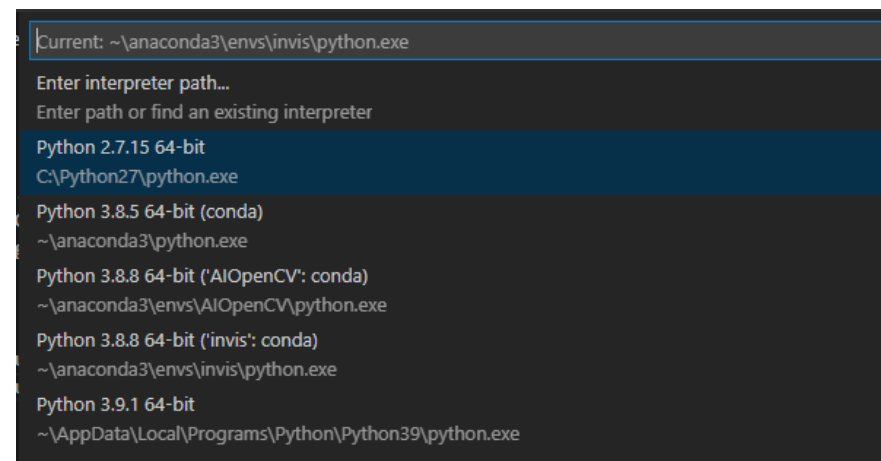
Visual Studio Code

- solidno integrirano razvojno okruženje (eng. Integrated Development Environment – IDE)
- Instalacija se može preuzeti na:
 - <https://code.visualstudio.com/download>
- Nakon instalacije potrebno je dodati Python u Extensions
- Ponovo pokrenite VS Code



Visual Studio Code – odabir interpretera

- odaberite željeni Python interpreter pomoću (vidi sliku):
 - kratice: **Ctrl + Shift + P**
 - **Python: Select Interpreter**



U padajućem meniju dostupni su:

- sistemski Python
- Python iz kreiranih virtualnih okruženja

Odaberite željeno virtualno okruženje

Visual Studio Code – odabir interpretera

- kako bi se automatski pokrenuo odgovarajući interpreter potrebno je odrediti odgovarajući terminal
- pritisnite **Ctrl + Shift + P** i odaberite **Terminal: Select Default Profile**
- sada je sve spremno za rad u virtualnom okruženju
- u slučaju teškoća konzultirati:

https://code.visualstudio.com/docs/python/environments#_choose-a-debugging-environment