

**Python virtualna
okruženja.
Visual Studio Code
razvojno okruženje.**



Sadržaj

- Instalacija Anaconda distribucije
- Kreiranje i rad s conda virtualnim okruženjem
- Instalacija dodatnih biblioteka u virtualno okruženje
- Instalacija Visual Studio Code IDE
- Podešavanje Visual Studio Code IDE-a

(ove upute vrijede za Windows OS)

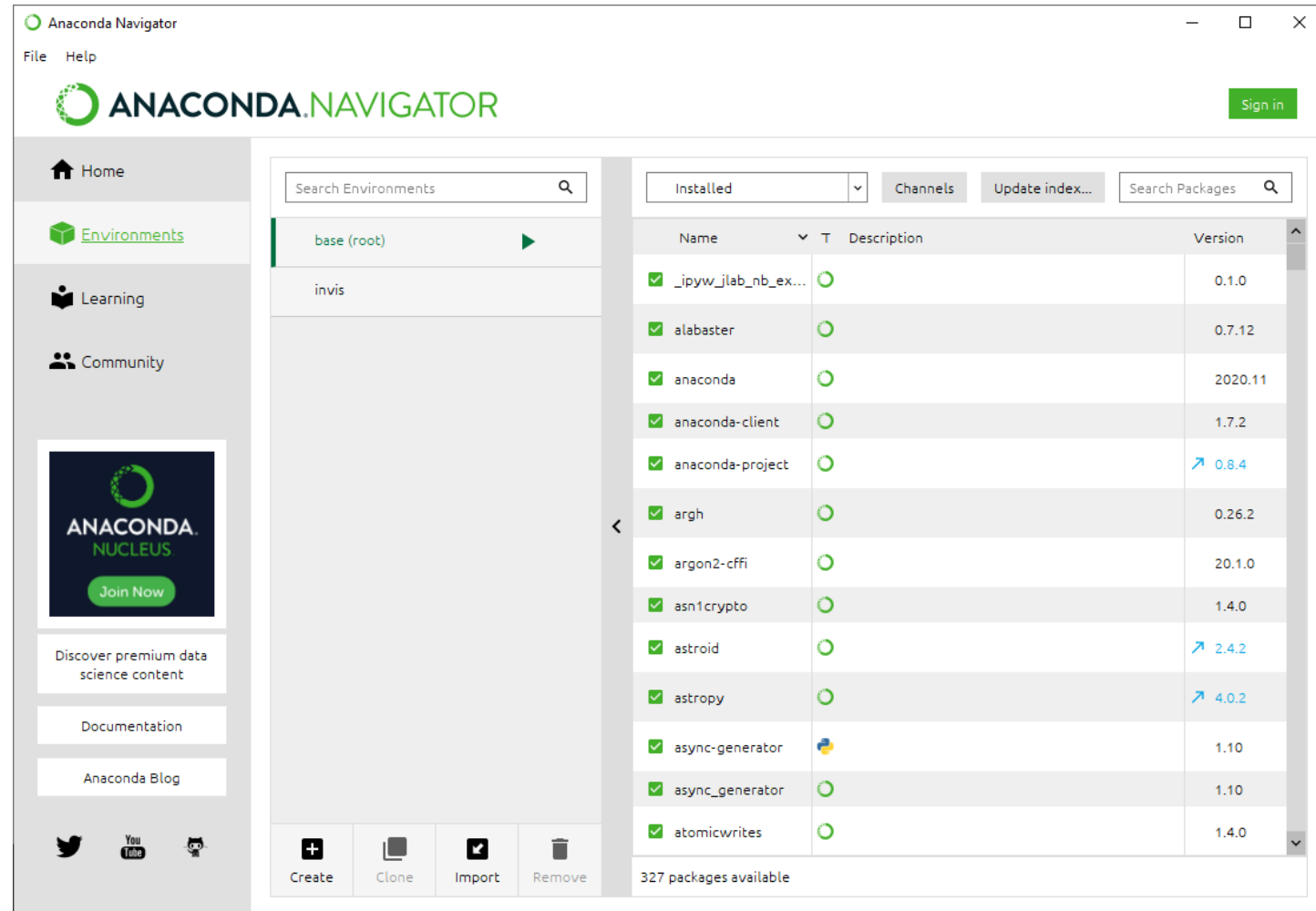
Anaconda distribucija

- 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



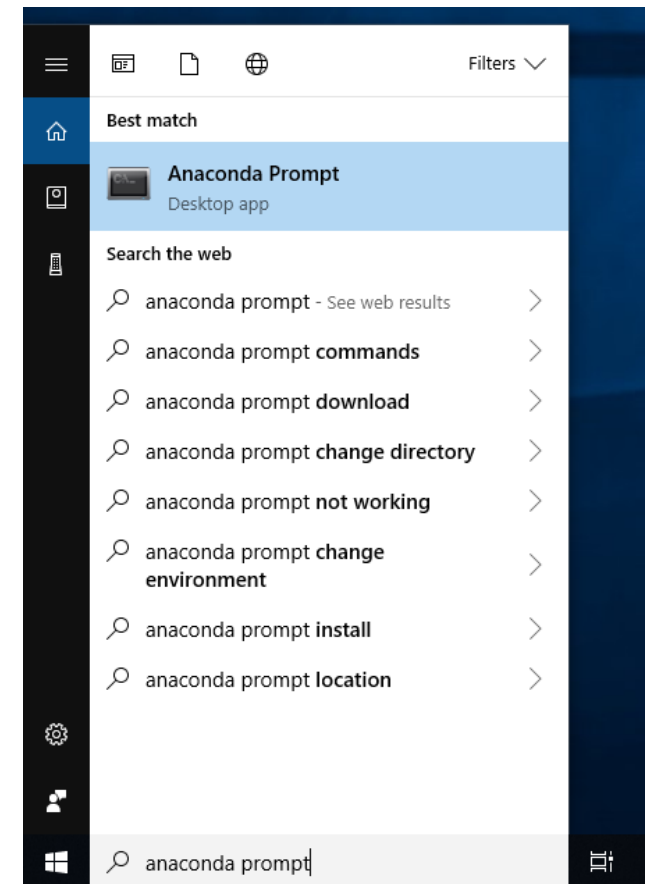
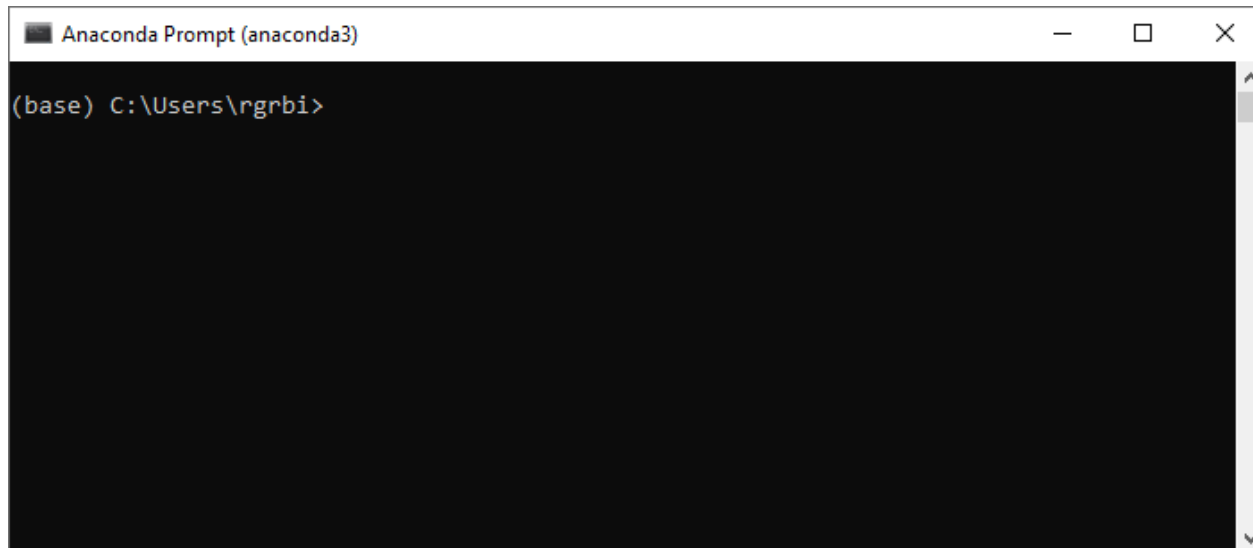
Anaconda navigator

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



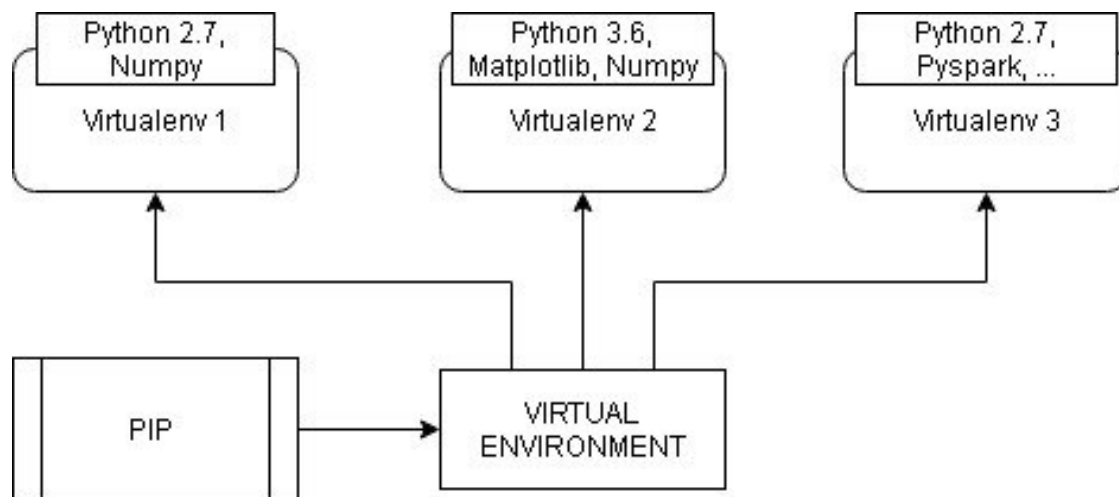
Anaconda prompt

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



Virtualna okruženja općenito

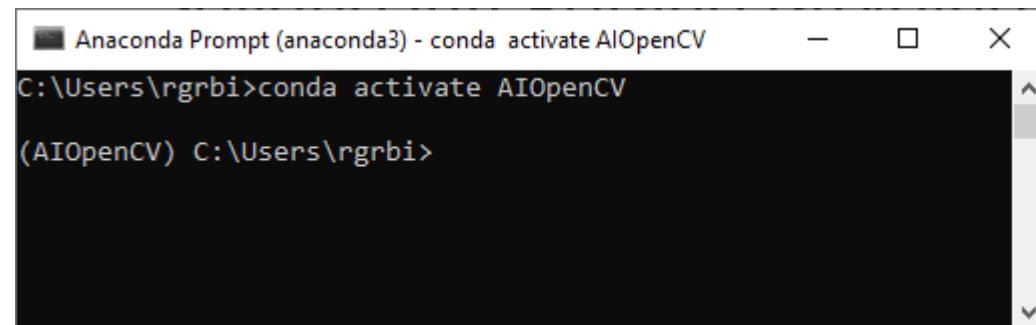
- skup određenih verzija biblioteka i Pythona
- zašto je to korisno?



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 AIOpenCV`

Trenutno aktivno okruženje

A screenshot of the Anaconda Prompt terminal window. The title bar reads "Anaconda Prompt (anaconda3) - conda activate AIOpenCV". The terminal shows the command "C:\Users\rgrbi>conda activate AIOpenCV" being executed, followed by the prompt "(AIOpenCV) C:\Users\rgrbi>". A blue arrow points from the text "Trenutno aktivno okruženje" to the "(AIOpenCV)" prefix in the terminal output.

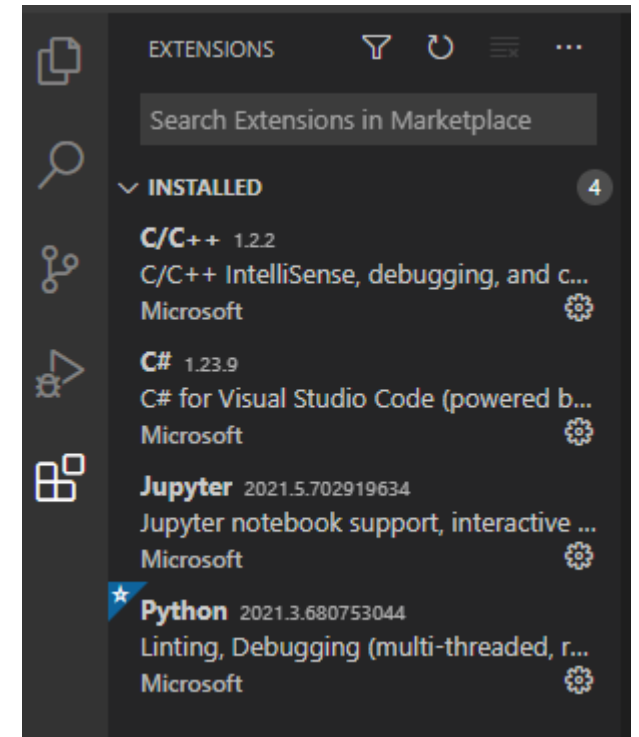
```
Anaconda Prompt (anaconda3) - conda activate AIOpenCV
C:\Users\rgrbi>conda activate AIOpenCV
(AIOpenCV) C:\Users\rgrbi>
```

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](#)

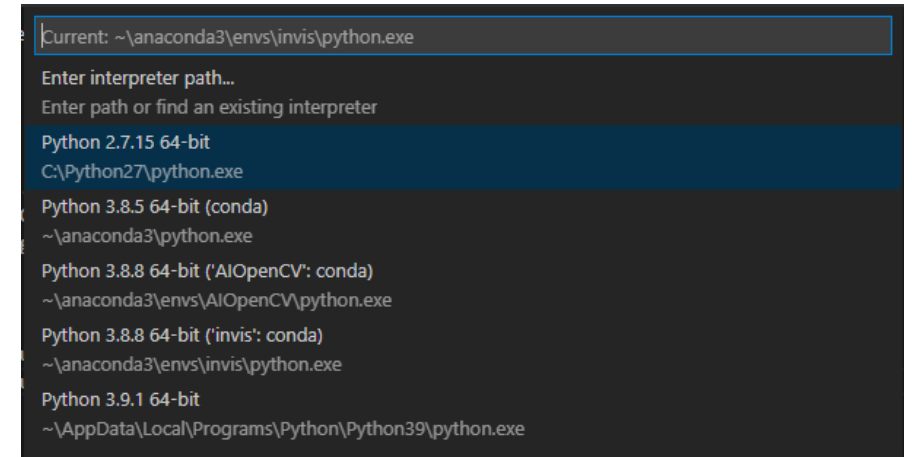
Visual Studio Code

- 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

- kreirajte folder na računalu naziva `opencv_primjer`
- zatim ovaj folder odaberite kao aktivni unutar VS Code pomoću **File → Open Folder**
- u ovaj folder ćete spremati programski kod i datoteke vezane za ovu vježbu
- 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 AIOpenCV

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 AIOpenCV virtualnom okruženju
- u slučaju teškoća konzultirati:

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

Pokretanje primjera

- kreirajte novi file unutar VS Code-a
- napravite neku sliku (npr. u Paintu) naziva `primjer.png`
- kopirajte navedeni kod u skriptu naziva `example.py`
- pokrenite kod Ctrl + F5

```
import cv2
```

```
# učitaj sliku
```

```
img_src = cv2.imread("primjer.png")
```

```
img_gray = cv2.cvtColor(img_src, cv2.COLOR_BGR2GRAY)
```

```
ret, img_gray_th = cv2.threshold(img_gray, 150, 255, cv2.THRESH_BINARY)
```

```
# detektiraj konture
```

```
cnts, hierarchy = cv2.findContours(image=img_gray_th, mode=cv2.RETR_TREE,  
method=cv2.CHAIN_APPROX_NONE)
```

```
# prikazi konture na originalnoj slici
```

```
cv2.drawContours(image=img_src, contours=cnts, contourIdx=-1,  
color=(0, 255, 0), thickness=2, lineType=cv2.LINE_AA)
```

```
cv2.imwrite("rezultat.png", img_src)
```

```
cv2.imshow("rezultat", img_src)
```

```
cv2.waitKey(0)
```

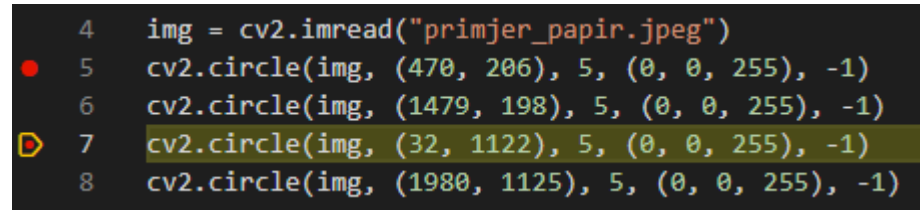
```
cv2.destroyAllWindows()
```

Visual Studio Code – debugging

- VS Code omogućuje relativno lako otklanjanje pogreški i uvid u vrijednosti pojedinih varijabli i sl. u debugging načinu rada
- postavite breakpointa na određenu liniju koda kako biste zaustavili izvršavanje koda na točno toj liniji (breakpoint je crvena točka lijevo od oznake reda koda)
- pokrenite debugging pomoću F5 ili iz izbornika Run->Start Debugging
- možete li za prethodni primjer reći kolika je visina i širina slike?
Pogledajte dio lijeve trake s nazivom VARIABLES

Primjer breakpointa

Program zaustavljen na
liniji 7



```
4  img = cv2.imread("primjer_papir.jpeg")
5  cv2.circle(img, (470, 206), 5, (0, 0, 255), -1)
6  cv2.circle(img, (1479, 198), 5, (0, 0, 255), -1)
7  cv2.circle(img, (32, 1122), 5, (0, 0, 255), -1)
8  cv2.circle(img, (1980, 1125), 5, (0, 0, 255), -1)
```

The image shows a code editor with a dark background. Line 7 is highlighted in yellow, and a red dot (breakpoint) is placed to its left. A blue arrow points from the text 'Primjer breakpointa' to the red dot. Another blue arrow points from the text 'Program zaustavljen na liniji 7' to the yellow highlight on line 7.

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
```