

התקנת חבילות וסביבה וירטואלית בפייתון

שפות-תיכנות בימינו בנויות בצורה מודולרית: על-גבי השפה הבסיסית, יש הרבה חבילות המרחיבות אותה בדרכים שונות. יש חבילות בסיסיות ועוד חבילות מתקדמות יותר המסתמכות עליהן. בקורס הזה למדנו על שתי חבילות שימושיות במיוחד, המאפשרות לנו לבצע אלגוריתמים מתקדמים בקלות: הספריה `cvxpy` – מאפשרת לפתור בעיות אופטימיזציה קמורות (ובפרט: בעיות של תיכנות ליניארי). הספריה `networkx` – מאפשרת להריץ אלגוריתמים על גרפים ורשתות.

התקנת חבילות דרך שורת הפקודה

רוב החבילות לא באות יחד עם ההתקנה של פייתון. הדבר מאפשר לכם להתקין רק את החבילות שאתם צריכים. יש הרבה דרכים להתקין חבילות בפייתון. הדרך הפשוטה ביותר היא להשתמש בפקודה `pip`. ראשית יש להתקין את `pip` על לינוקס. הפקודה שעובדת אצלי (אובונטו 20.04) היא:

```
sudo apt install python3-pip
```

אצלכם הפקודה עשויה להיות מעט שונה – חפשו את הפקודה המתאימה למערכת שלכם.

כדי לבדוק שהתקנתם את `pip` כמו שצריך, נסו להתקין את `numpy` – ספריה לעיבוד מספרים ומטריצות. אופן השימוש המדויק ב `pip` תלוי בגירסה של מערכת ההפעלה שלכם. הפקודה שעובדת אצלי היא:

```
pip install numpy
```

אם זה לא עובד, אפשר גם לנסות `pip3` במקום `pip`; או לנסות:

```
python -m pip install numpy
```

או לנסות `python3` במקום `python`. ואם אתם על לינוקס, נסו להוסיף `sudo` לפני הפקודה.

כדי לוודא שההתקנה עברה בהצלחה, אפשר פשוט לפתוח את שורת הפקודה של פייתון ולנסות לייבא את החבילה, למשל:

```
python
```

```
>>> import numpy;
```

אם לא קיבלתם הודעת שגיאה, כנראה שהחבילה הותקנה בהצלחה.

אם קיבלתם הודעת שגיאה, ייתכן שההודעה קשורה לבאג בגירסה האחרונה של החבילה; במקרה כזה, אפשר לבקש מ `pip` להתקין גרסה מוקדמת יותר, למשל:

```
python -m pip install numpy==1.19.3
```

התקנת חבילות דרך PyCharm

אפשר להתקין חבילות גם ישירות דרך סביבת העבודה PyCharm. יש לבחור בתפריט `File` - `Settings`, ולבחור בצד שמאל `Python Interpreter`. בצד ימין למטה רואים את רשימת החבילות המותקנות. אפשר ללחוץ על הסימן + מימין למעלה, ולבחור חבילה+גירסה להתקנה.

סביבה וירטואלית

שימו לב! אם לא נתקלתם בבעיה בתהליך ההתקנה הרגיל של חבילות, אין צורך שתטרחו להתקין סביבה וירטואלית. פרק זה נועד רק למשתמשים שנתקלו בבעיות בהתקנת החבילות.

אחד האתגרים בעבודה עם חבילות רבות הוא, שכל חבילה מסתמכת על חבילות אחרות, ולפעמים יש אי-התאמה בין גרסאות של חבילות דרושות. לדוגמה, ייתכן שעבור פרוייקט מסויים, אתם צריכים את חבילה א, המשתמשת בגרסה 1 של חבילה ג; אבל עבור פרוייקט אחר, אתם צריכים את חבילה ב, והיא משתמשת דווקא בגרסה 2 של חבילה ג. איזו גרסה של חבילה ג תתקינו? אתם יכולים בכל פעם להחליף גרסאות בעזרת pip, אבל זו טירחה מיותרת. במקום זה, אפשר ליצור שתי **סביבות וירטואליות** שונות, ולעבור מסביבה לסביבה לפי הצורך. ישנן הרבה שפות תכנות מודרניות התומכות במנגנון של סביבה וירטואלית; אחת מהן היא פייתון.

איך מגדירים סביבה וירטואלית בפייתון? קודם-כל מתקינים חבילה בשם venv, למשל באובונטו 20.04:

```
sudo apt install python3-venv
```

אחר-כך יוצרים סביבה וירטואלית ונותנים לה שם. נניח שנקרא לסביבה שלנו myenv:

```
python -m venv myenv
```

הפקודה הזאת יוצרת על המחשב שלכם תיקיה המייצגת את הסביבה הוירטואלית (סביבה היא בסה"כ תיקיה על המחשב). כדי להפעיל אותה, צריך לכתוב בלינוקס:

```
$ source myenv/bin/activate
```

בחלונות הפקודה מעט שונה:

```
> myenv\Scripts\activate.bat
```

בשני המקרים, אתם אמורים עכשיו לראות מצד שמאל של הסמן את שם הסביבה בסוגריים, למשל:

```
(myenv) $
```

כך שתדעו תמיד באיזו סביבה אתם נמצאים. עכשיו אתם יכולים להתקין גרסאות שונות של החבילות שאתם רוצים, והחבילות יותקנו בתוך הסביבה בלבד.

כדי לצאת מהסביבה הוירטואלית בחזרה לסביבה הכללית, משתמשים בפקודה deactivate. כך אפשר לעבור בין סביבות בקלות.

סביבה וירטואלית עם גרסאות שונות של פייתון

בעזרת סביבות וירטואליות, אפשר לעבוד גם עם גרסאות שונות של שפת פייתון עצמה. לשם כך יש להוריד את הגרסאות הרצויות מכאן: <https://www.python.org/downloads> ולזכור באיזו תיקיה כל גרסה הותקנה. אצלי, במערכת חלונות, כל הגרסאות מותקנות כאן:
C:\Users\user\AppData\Local\Programs\Python\PythonNN
כאשר NN הוא מספר הגרסה, למשל 37 (עבור גרסה 3.7).

כדי להתקין סביבה וירטואלית המסתמכת על גרסה מסוימת של פייתון, צריך פשוט לקרוא ל venv עם הפייתון המתאים. למשל, בסביבת חלונות:

```
C:\Users\user\AppData\Local\Programs\Python\Python37\python -m venv env37
```

```
C:\Users\user\AppData\Local\Programs\Python\Python39\python -m venv env39
```

יוצר שתי סביבות – אחת לגרסה 3.7 ואחת לגרסה 3.9.

אם נפעיל כל אחת מהסביבות, נראה שהגרסה של פייתון אכן שונה בכל אחת מהן:

```
> env37\Scripts\activate
(env37) > python -version
Python 3.7.4
> env39\Scripts\activate
(env39) > python -version
Python 3.9.0
```

בכל אחת מהסביבות אפשר להתקין את החבילות שרוצים לעבוד בהן עם אותה גרסה של פייתון.

התקנת סביבה וירטואלית דרך PyCharm

אפשר להתקין סביבה וירטואלית גם ישירות דרך סביבת העבודה PyCharm. יש לבחור בתפריט File Settings, - ולבחור בצד שמאל Python Interpreter. בצד ימין למעלה הולכים לשורה Project Interpreter, לוחצים על גלגל-השיניים מימין, ובחרים Add. בחרים environment, בחרים את גרסת הבסיס של פייתון (מבין הגרסאות שהתקנתם על המחשב) ובחרים תיקיה על המחשב שלכם שבה תשב הסביבה (כאמור, כל סביבה מיוצגת ע"י תיקיה על המחשב).

אחרי שיצרתם סביבה, אתם יכולים לעבור בין סביבו, ובחזרה לסביבה הראשית, ע"י בחירת האפשרות המתאימה בתפריט של Project Interpreter בצד ימין למעלה.

מקורות

- עבודה עם סביבה וירטואלית בפייתון: <https://docs.python.org/3/tutorial/venv.html>
- עבודה עם סביבה וירטואלית בפייצ'ארם: <https://www.jetbrains.com/help/pycharm/creating-virtual-environment.html>
- ניהול כמה גרסאות של פייתון במקביל: <https://www.freecodecamp.org/news/manage-multiple-python-versions-and-virtual-environments-venv-pyenv-pyvenv-a29fb00c296f/>
- עוד מידע על סביבות וירטואליות בסביבת חלונות: <https://www.freecodecamp.org/news/installing-multiple-python-versions-on-windows-using-virtualenv/>

סיכום: אראל סגל-הלוי.