

# مدیریت کتابخانه‌ها با استفاده از pip

در درسنامه ماژول‌ها با کتابخانه‌های استاندارد پایتون مانند `math` آشنا شدید. این کتابخانه‌ها به صورت پیش فرض در کنار پایتون قرار دارند اما کتابخانه‌های کاربردی پایتون در زمینه هوش مصنوعی معمولاً کتابخانه‌های استاندارد پایتون نیستند و برخی از کتابخانه‌های مفید (که به طور پیش فرض همراه پایتون نیستند) را باید به سیستم اضافه کنیم. مدیریت کردن این کتابخانه‌های افزوده معمولاً دشوار است.

می‌توانیم از pip جهت مدیریت این بسته‌ها استفاده کنیم. برای اطلاع از نحوه نصب pip به [این‌جا](#) مراجعه کنید.

با اجرای دستور زیر می‌توانید یک کتابخانه را به سیستم اضافه کنید. (این دستور را باید در `cmd` و یا `terminal` اجرا کنید).

```
1 | > pip3 install package-name
```

برای مثال می‌توانید کتابخانه `numpy` را که در فصل بعدی با آن آشنا خواهید شد، با استفاده از همین دستور نصب کنید. (توجه کنید که چون از پایتون ۳ استفاده می‌کنیم باید در ترمینال هم از `pip3` استفاده کنیم).

همچنین با استفاده از دستور زیر می‌توانید همه کتابخانه‌هایی که در سیستم موجود است را به همراه نسخه‌ای که دارند، مشاهده کنید.

```
1 | > pip3 list
2 | pycairo (1.16.2)
3 | pycrypto (2.6.1)
4 | pycups (1.9.73)
5 | pygame (1.9.6)
6 | pyglet (1.3.2)
7 | Pygments (2.3.1)
8 | pygobject (3.26.1)
9 | pymacaroons (0.13.0)
10 | PyNaCl (1.1.2)
11 | pyRFC3339 (1.0)
12 | python-apt (1.6.3+ubuntu1)
13 | python-dateutil (2.7.5)
14 | python-debian (0.1.32)
15 | pytz (2018.3)
16 | pyxdg (0.25)
17 | PyYAML (3.12)
```

در لیست بالا می‌توانید برخی از کتابخانه‌هایی که خروجی داده شدند را مشاهده کنید. مقدار درون پرانتز، نسخه آن کتابخانه را نشان می‌دهد.

همچنین با استفاده از دستور `uninstall` می‌توانید یک `package` نصب شده را از سیستم حذف کنید.

```
1 | > pip3 uninstall numpy
```

برای مثال در نمونه‌ی بالا کتابخانه `numpy` را که نصب کردیم، حذف می‌کنیم.

برای اطلاع از قابلیت‌های بیشتر `pip` می‌توانید با نوشتن دستور زیر در ترمینال، در خروجی اطلاعات مفیدی در این مورد دریافت کنید.

```
1 | > pip3 --help
```