

جزوه برنامه نویسی پیشرفته

امیرحسین قلی زاده

دانشگاه علوم و فنون آریان ترم دوم 1403 - 1404

جزوه آموزشی – جلسه پنجم: محیط مجازی، پکیج منیجر، و کتابخانه sys

استاد: امیرحسین قلی زاده | دانشگاه: علوم و فنون آریان | نیمسال تحصیلی: نیمسال دوم 1403 – 1404

Venv: محیط مجازی در پایتون

چرا به محیط مجازی نیاز داریم؟

در پروژه‌های مختلف ممکن است به نسخه‌های متفاوتی از یک کتابخانه نیاز داشته باشیم. محیط مجازی:

- از تداخل نسخه‌ها جلوگیری می‌کند
- پروژه‌ها را مستقل نگه می‌دارد
- مسیر نصب پکیج‌ها را جدا از پایتون سیستم نگه می‌دارد

ایجاد محیط مجازی

```
1. python -m venv myenv
2.
```

ساختار پوشه بعد از ایجاد:

```
1. myenv/
2. |— bin/ (Linux/Mac)  |— Scripts/ (Windows)
3. |— lib/
4. |— pyvenv.cfg
5.
```

فعال‌سازی محیط مجازی

سیستم عامل	دستور
Windows	myenv\Scripts\activate
macOS/Linux	source myenv/bin/activate

فعال شدن موفق یعنی نام محیط (مثل myenv) در ترمینال ظاهر می‌شود.

غیرفعال‌سازی:

```
1. deactivate
2.
```

Pip: مدیر بسته پیش‌فرض پایتون

نصب کتابخانه:

```
1. pip install numpy
2.
```

نصب از فایل requirements.txt:

```
1. pip install -r requirements.txt
2.
```

به‌روزرسانی پکیج:

```
1. pip install --upgrade requests
2.
```

حذف پکیج:

```
1. pip uninstall matplotlib
2.
```

مشاهده پکیج‌های نصب شده:

```
1. pip list
2.
```

ساخت فایل requirements.txt:

```
1. pip freeze > requirements.txt
2.
```

Uv: نصب سریع‌تر جایگزین pip

[uv](#) یک ابزار بسیار سریع‌تر برای مدیریت پکیج‌هاست که جایگزین pip، venv و virtualenv شده است.

ویژگی‌های کلیدی uv :

- بسیار سریع‌تر از pip
- نصب، ایجاد محیط مجازی، و مدیریت dependency
- نصب پکیج بدون نیاز به C compiler یا wheel build

نصب uv:

```
1. curl -Ls https://astral.sh/uv/install.sh | sh
2.
```

یا:

```
1. pip install uv
2.
```

📦 ساخت محیط مجازی و نصب پکیج:

```
1. uv venv
2. source .venv/bin/activate
3. uv pip install pandas
4.
```

uv pip دقیقاً مثل pip کار می‌کند، اما سریع‌تر است.

کتابخانه sys: ابزار ارتباط با سیستم

کتابخانه sys اطلاعاتی در مورد مفسر پایتون و محیط سیستم ارائه می‌دهد.

وارد کردن:

```
1. import sys
2.
```

دستورات مهم:

دستور	کاربرد
sys.argv	آرگومان های خط فرمان
sys.exit()	خروج از برنامه
sys.path	مسیرهای جستجوی ماژول ها
sys.version	نسخه ی پایتون
sys.platform	سیستم عامل) مثل 'win32' یا 'linux'

مثال:

```

1. # script.py
2. import sys
3.
4. print("Python version:", sys.version)
5. print("Platform:", sys.platform)
6. print("Arguments:", sys.argv)
7.

```

اجرا:

```

1. python script.py hello world
2.

```

خروجی:

```

1. Python version: 3.11.2 ...
2. Platform: win32
3. Arguments: ['script.py', 'hello', 'world']
4.

```

تمرینات پیشنهادی

1. یک محیط مجازی بساز و در آن flask نصب کن.
2. با استفاده از pip یک پکیج نصب و حذف کن.
3. با sys برنامه‌ای بنویس که تمام آرگومان‌های خط فرمان را برعکس چاپ کند.
4. تفاوت pip و uv را از نظر سرعت بررسی کن.
5. یک فایل requirements.txt بساز و در یک سیستم دیگر پروژه را راه‌اندازی کن.

جمع‌بندی

ابزار / کتابخانه	کاربرد اصلی
venv	جداسازی محیط پروژه‌ها
pip	نصب و مدیریت کتابخانه‌ها
uv	جایگزین بسیار سریع برای pip/venv
sys	دسترسی به اطلاعات سیستم و آرگومان‌ها

موفق باشید!