

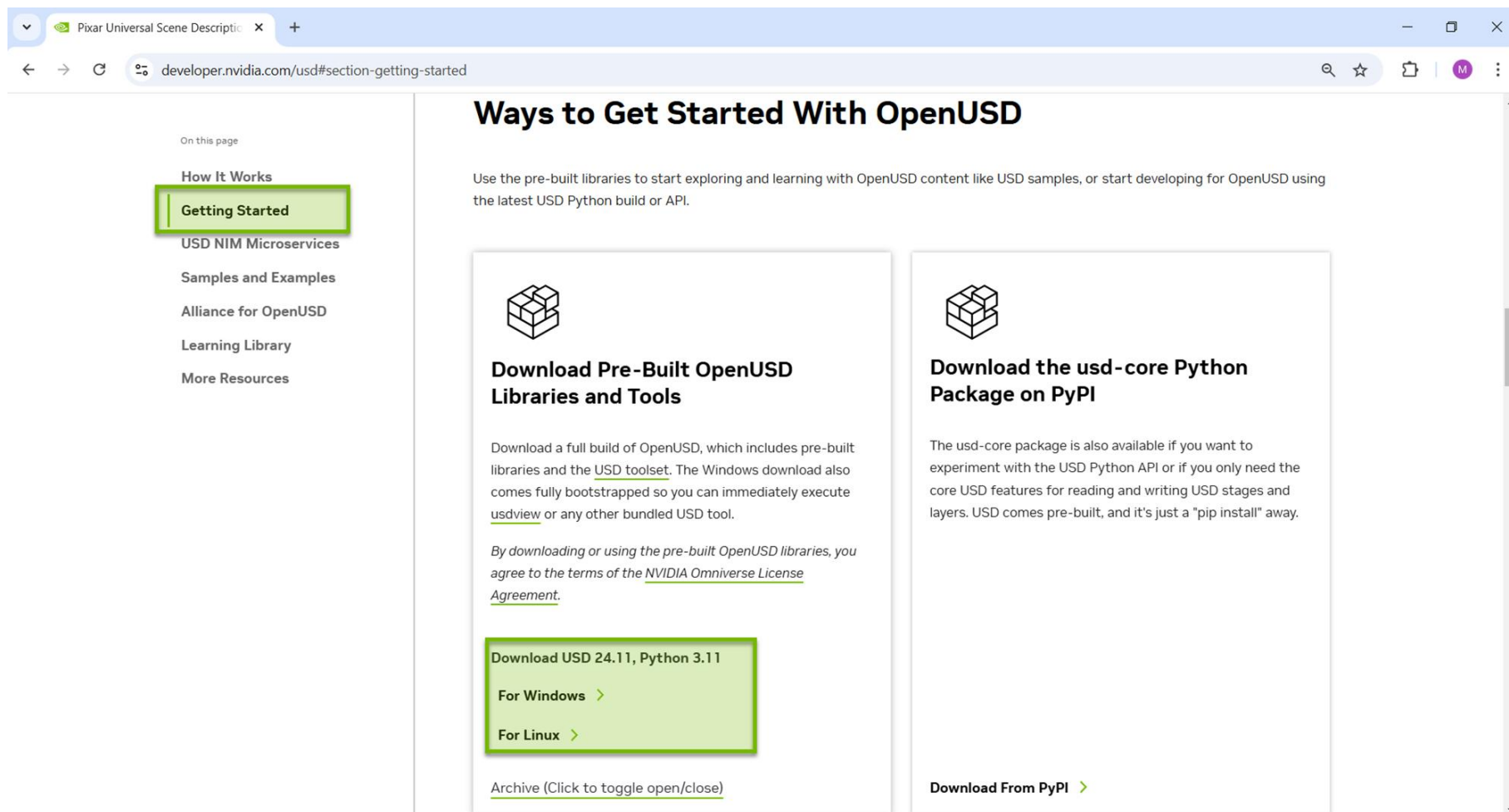
Downloading the Libraries and Tools

[Installing usdview](#)

[OpenUSD Git](#)

Download USD and Python

- 前往 NVIDIA 的 [OpenUSD 開發者資源頁面](#)
- 點選右側 Getting Started
- 下載對應作業系統的 OpenUSD 模組包和工具




The screenshot shows a web browser window with the URL `developer.nvidia.com/usd#section-getting-started`. The page title is "Ways to Get Started With OpenUSD". On the left, a sidebar lists navigation options: "On this page", "How It Works", "Getting Started" (highlighted with a green box), "USD NIM Microservices", "Samples and Examples", "Alliance for OpenUSD", "Learning Library", and "More Resources". The main content area has a heading "Ways to Get Started With OpenUSD" and a subheading "Use the pre-built libraries to start exploring and learning with OpenUSD content like USD samples, or start developing for OpenUSD using the latest USD Python build or API." Below this, there are two main sections. The first section, "Download Pre-Built OpenUSD Libraries and Tools", includes a description of the full build, a link to the [NVIDIA Omniverse License Agreement](#), and a green box with links for "Download USD 24.11, Python 3.11", "For Windows >", and "For Linux >". The second section, "Download the usd-core Python Package on PyPI", includes a description of the package and a link for "Download From PyPI >".

On this page

- How It Works
- Getting Started**
- USD NIM Microservices
- Samples and Examples
- Alliance for OpenUSD
- Learning Library
- More Resources

Ways to Get Started With OpenUSD

Use the pre-built libraries to start exploring and learning with OpenUSD content like USD samples, or start developing for OpenUSD using the latest USD Python build or API.



Download Pre-Built OpenUSD Libraries and Tools


Download a full build of OpenUSD, which includes pre-built libraries and the USD toolset. The Windows download also comes fully bootstrapped so you can immediately execute [usdview](#) or any other bundled USD tool.

By downloading or using the pre-built OpenUSD libraries, you agree to the terms of the [NVIDIA Omniverse License Agreement](#).

Download USD 24.11, Python 3.11

- For Windows** >
- For Linux** >

[Archive \(Click to toggle open/close\)](#)



Download the usd-core Python Package on PyPI

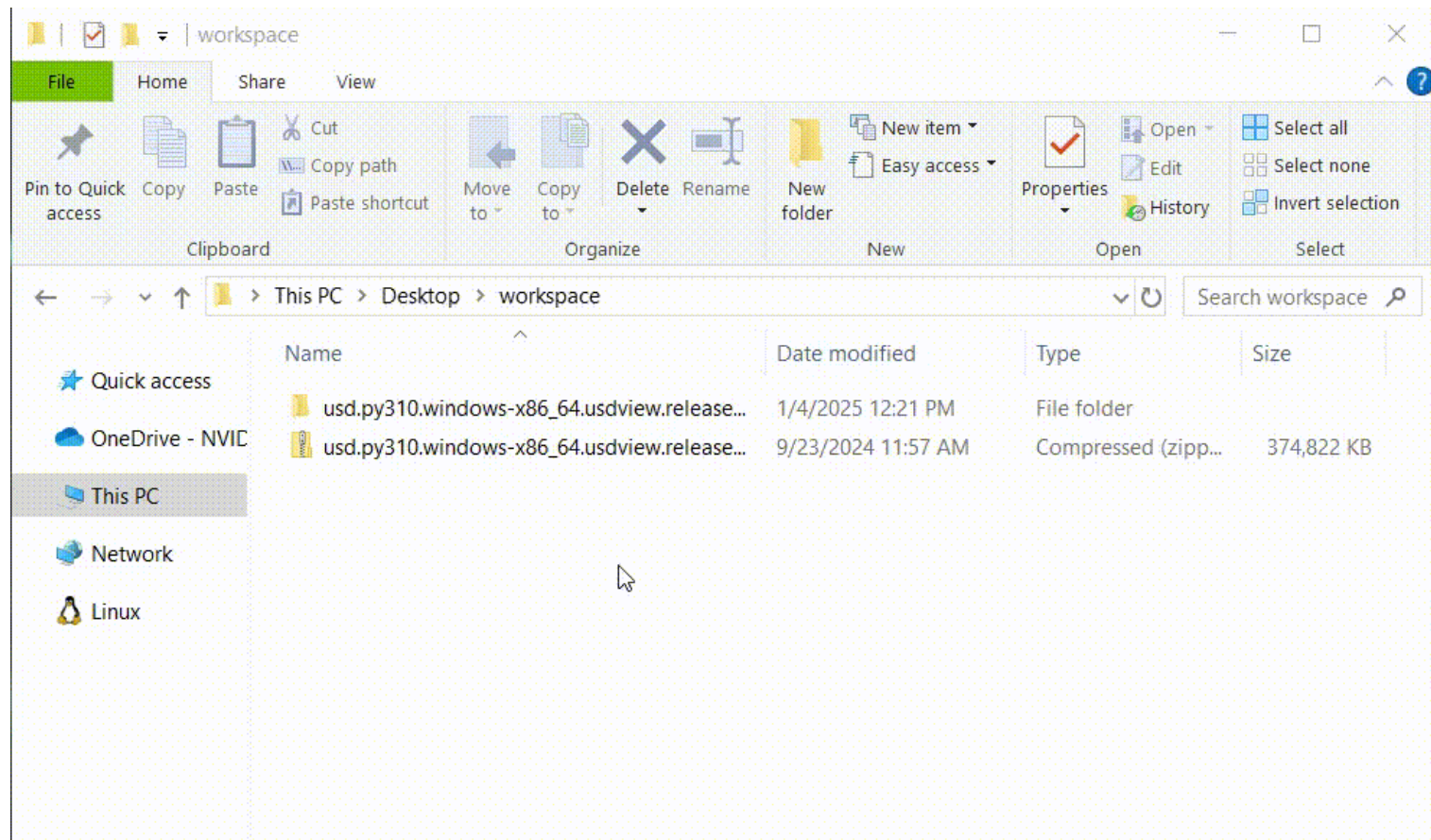
The usd-core package is also available if you want to experiment with the USD Python API or if you only need the core USD features for reading and writing USD stages and layers. USD comes pre-built, and it's just a "pip install" away.

Download From PyPI >

After Downloading Zipped Folder

- 下載壓縮資料夾後，解壓縮所有內容。**這可能需要一段時間(1.3GB左右)**。
- 解壓縮完成後，將解壓縮後的資料夾重新命名為 `usd_root`
- 為確保一切正常，接著將在 Windows 中使用 Visual Studio Code 進行測試
- ◆ **注意! 對於 Linux :**
需要按照以下步驟安裝 Ubuntu 中缺少的 X11 依賴項。請先執行此操作，然後再繼續下一步:

```
sudo apt-get install libxkbcommon-x11-0 libxcb-xinerama0  
libxcb-image0 libxcb-shape0 libxcb-render-util0 libxcb-icccm4  
libxcb-keysyms1
```



Usdview

- 開啟 Visual Studio Code
- 在「檔案」→「開啟資料夾」下，找到你的 `usd_root/` 資料夾
- 透過前往 Terminal → New Terminal 開啟一個新終端。
- 在此我們使用 Windows 上的 PowerShell 作為預設終端，在終端機中，輸入以下指令：

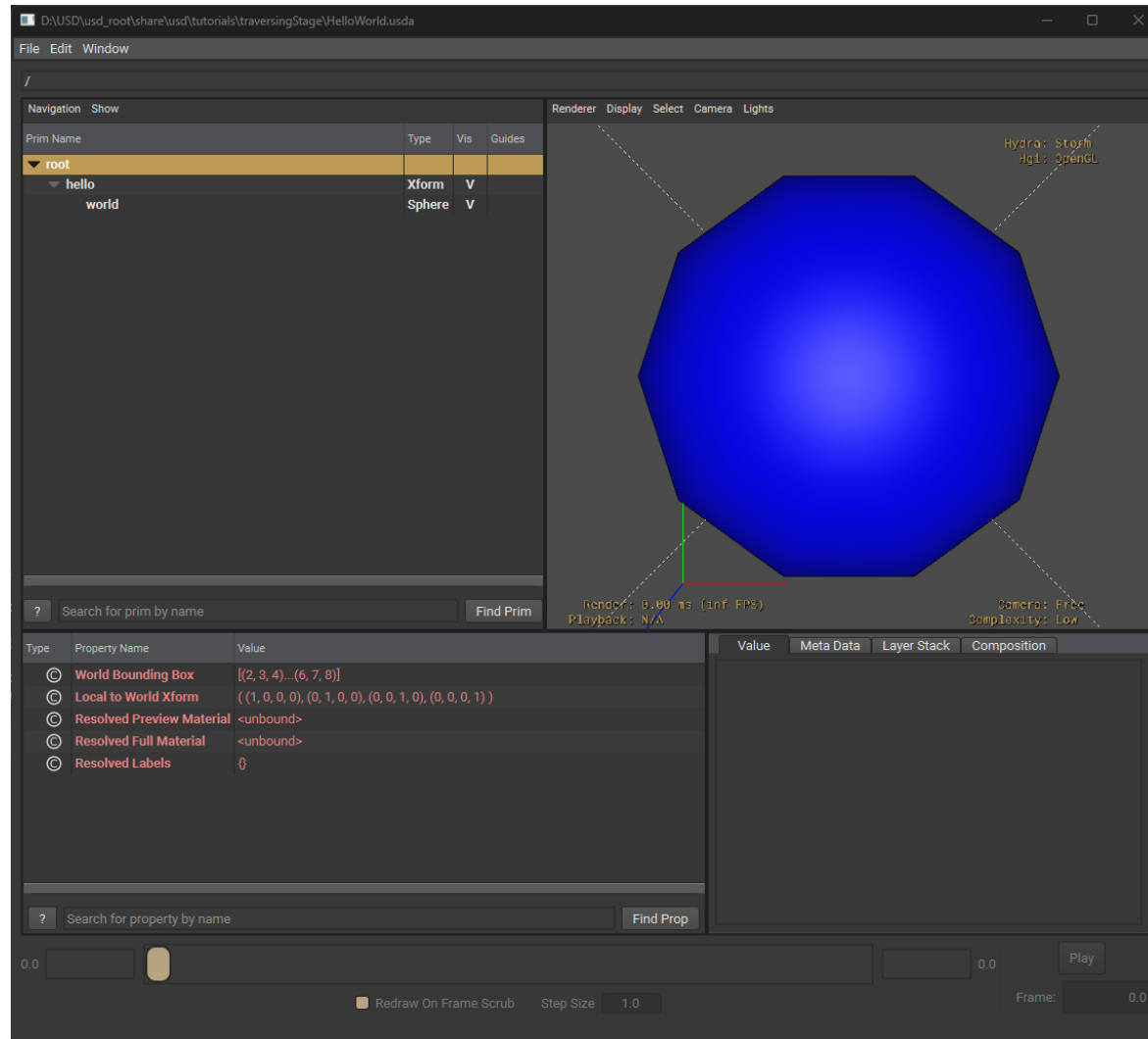
➤ **Windows:**

```
.\scripts\usdview_gui.bat
```

➤ **Linux/macOS:**

```
./scripts/usdview_gui.sh
```

- 這會以 `usdview` 開啟 `HelloWorld.usda` 檔
- 藉此可熟悉 `usdview` 介面
- 為了方便使用，你可以將 `scripts/` 資料夾加入到 `PATH` 環境變數中，但不做不影響後面操作



Setting Up the Python Environment

我們將使用 **Visual Studio Code** 來建立虛擬環境。

1. 開啟 **Visual Studio Code**。
2. 進入 **檔案 (File) → 開啟資料夾 (Open Folder)**。
3. 透過 **終端機 (Terminal) → 新終端機 (New Terminal)** 打開一個終端機視窗。
4. 在終端機中執行以下指令：

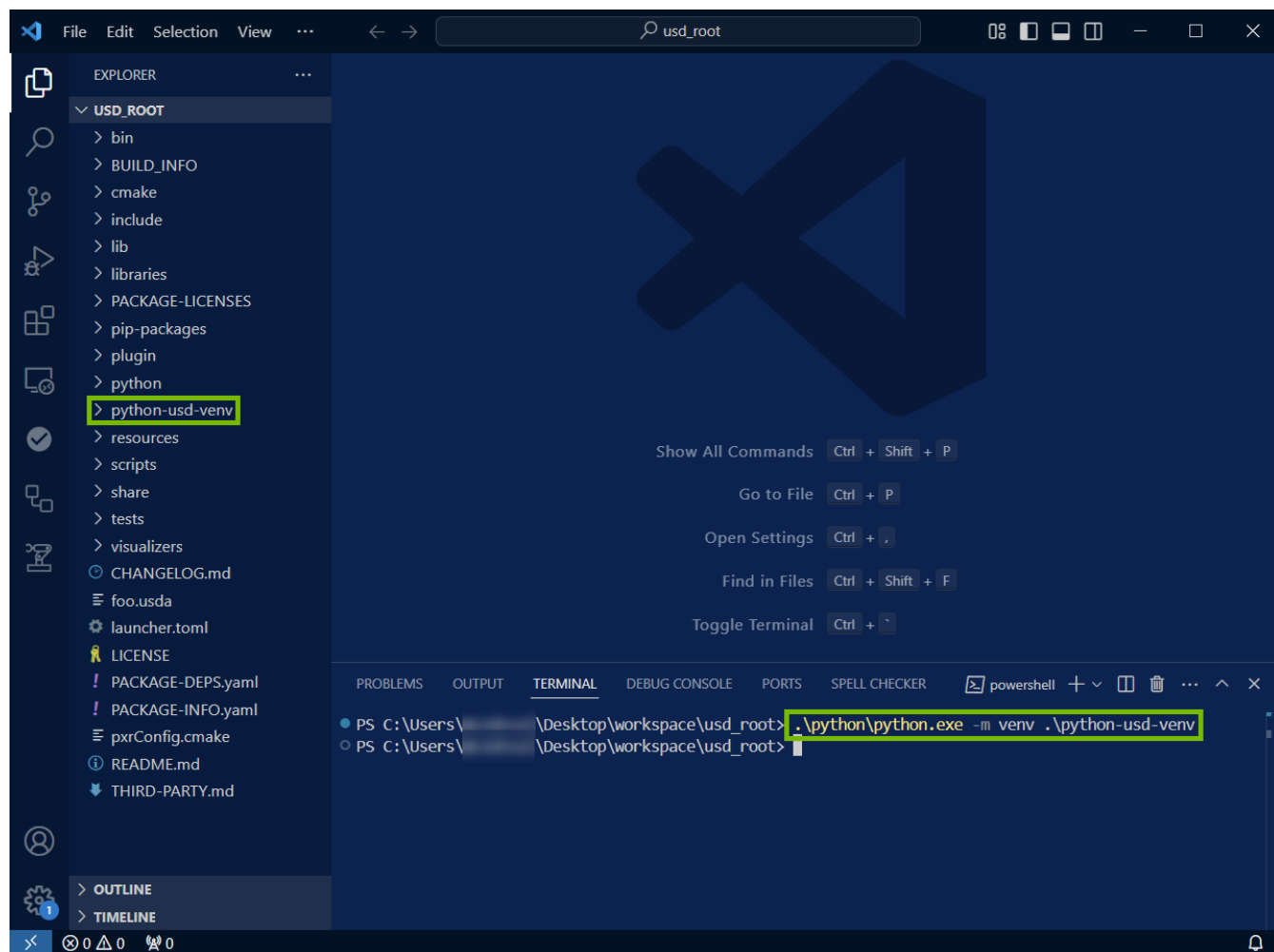
➤ **Windows:**

```
.\python\python.exe -m venv .\python-usd-venv
```

➤ **Linux/macOS:**

```
./python/python -m venv ./python-usd-venv
```

- 這將在目前根目錄中建立一個虛擬 Python 環境。



Virtual Environment

- 如何使用虛擬環境取決於您使用的平台和 Shell。請參閱下表：

Platform	Shell	Command to activate virtual environment
POSIX	bash/zsh	<code>\$ source python-usd-venv/bin/activate</code>
	fish	<code>\$ source python-usd-venv/bin/activate.fish</code>
	csh/tcsh	<code>\$ source python-usd-venv/bin/activate.csh</code>
Windows	cmd.exe	<code>C:\> python-usd-venv\Scripts\activate.bat</code>
	PowerShell	<code>PS C:\> python-usd-venv\Scripts\Activate.ps1</code>

- 注意! 對於 Windows/PowerShell，可能還需要先設定執行原則以允許腳本執行。對此可以使用以下指令：

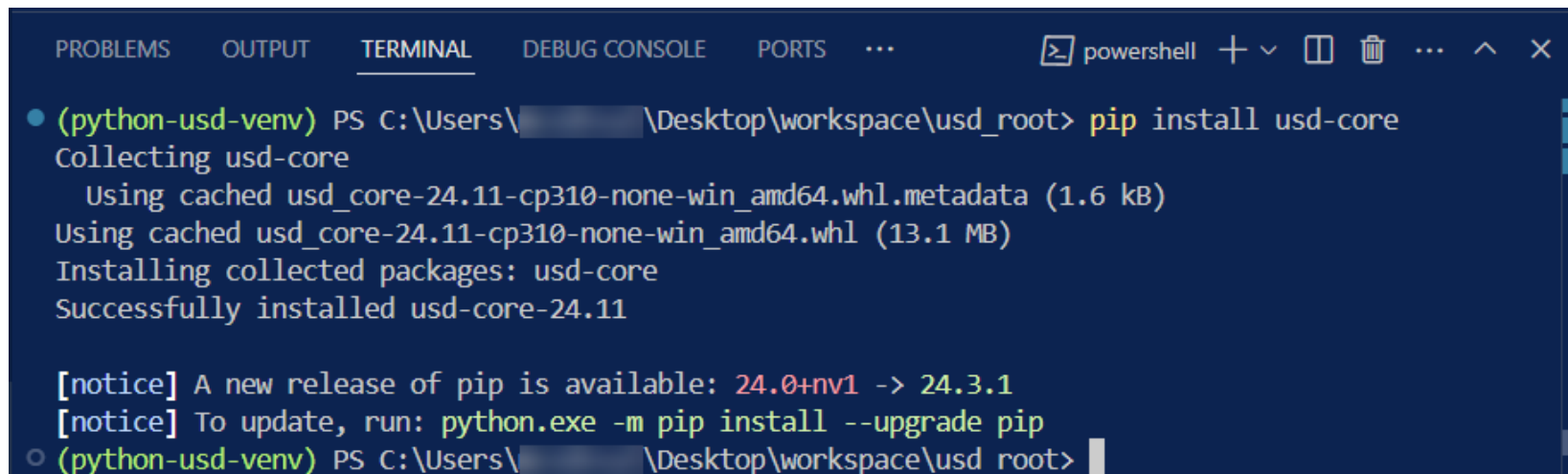
```
Set-ExecutionPolicy RemoteSigned -Scope CurrentUser
```

Installing **usd-core**

- 現在我們要安裝 usd-core，它是一個必要套件，能讓我們使用 USD API。
- 請先確認已啟用並執行先前步驟建立的 Python 虛擬環境。在環境啟用後，執行以下指令來安裝 usd-core:

➤ Windows/Linux/macOS:

```
pip install usd-core
```



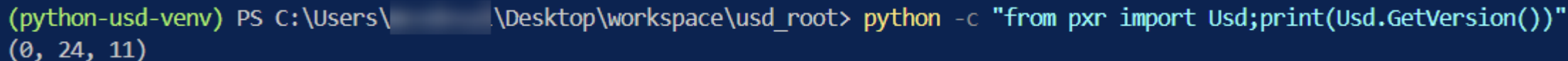
```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS ... powershell + - v [ ] [X] ... ^ X
• (python-usd-venv) PS C:\Users\... \Desktop\workspace\usd_root> pip install usd-core
Collecting usd-core
  Using cached usd_core-24.11-cp310-none-win_amd64.whl.metadata (1.6 kB)
  Using cached usd_core-24.11-cp310-none-win_amd64.whl (13.1 MB)
Installing collected packages: usd-core
Successfully installed usd-core-24.11

[notice] A new release of pip is available: 24.0+nv1 -> 24.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip
○ (python-usd-venv) PS C:\Users\... \Desktop\workspace\usd_root>
```

- 要檢查 usd-core 是否已正確安裝，我們可以在虛擬環境中執行以下指令:

➤ Windows/Linux/macOS:

```
python -c "from pxr import Usd;print(Usd.GetVersion())"
```



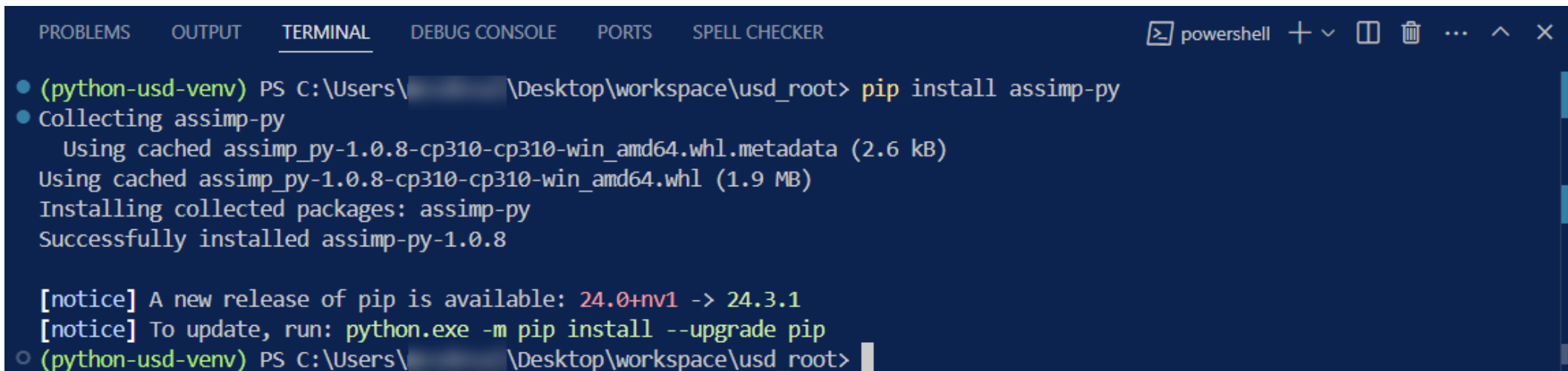
```
(python-usd-venv) PS C:\Users\... \Desktop\workspace\usd_root> python -c "from pxr import Usd;print(Usd.GetVersion())"
(0, 24, 11)
```

Installing Assimp

- 請確保已啟動並運行 Python 虛擬環境。如果你是接續上一節操作，這一步應該已經完成。
- 在虛擬環境運行中，執行以下指令來安裝 **assimp**。

➤ Windows/Linux/macOS:

```
pip install assimp-py
```



```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE  PORTS  SPELL CHECKER
(python-usd-venv) PS C:\Users\...\Desktop\workspace\usd_root> pip install assimp-py
Collecting assimp-py
  Using cached assimp_py-1.0.8-cp310-cp310-win_amd64.whl.metadata (2.6 kB)
  Using cached assimp_py-1.0.8-cp310-cp310-win_amd64.whl (1.9 MB)
Installing collected packages: assimp-py
Successfully installed assimp-py-1.0.8

[notice] A new release of pip is available: 24.0+nv1 -> 24.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip
(python-usd-venv) PS C:\Users\...\Desktop\workspace\usd_root>
```

- 要檢查 Assimp 是否已正確安裝，我們可以在虛擬環境中執行以下指令:

➤ Windows/Linux/macOS:

```
python -c "import assimp_py"
```

- 如未出現以下錯誤：ModuleNotFoundError: No module named 'assimp_py'，表示 Assimp 安裝成功。

恭喜，你已安裝完成使用 OpenUSD 初步所需的資源，並可使用 `usdview`