

00.NOTE BOOK 环境安装

一.为何选择notebook ?

Jupyter notebook (<http://jupyter.org/>) 是一种 Web 应用，能让用户将说明文本、数学方程、代码和可视化内容全部组合到一个易于共享的文档中。

可选择语言: 支持超过40种编程语言，包括Python、R、Julia、Scala等。

分享笔记本: 可以使用电子邮件、Dropbox、GitHub和Jupyter Notebook Viewer与他人共享。

交互式输出: 代码可以生成丰富的交互式输出，包括HTML、图像、视频、LaTeX等等。

大数据整合: 通过Python、R、Scala编程语言使用Apache Spark等大数据框架工具。支持使用pandas、scikit-learn、ggplot2、TensorFlow来探索同一份数据。

二.环境安装

1. Python环境安装

jupyter notebook 基于Python， 机器必须有Python运行环境，建议安装3.8.x的环境

Linux/macOS/WSL2 安装

请参考docs文件夹下的00_Linux_WSL_MacOS_Install_Python.pdf

Windows安装

请参考docs文件夹下的00_Windows_Install_Python.pdf

一些补充：

Windows 下建议使用WSL2 来设置，关于WSL2 的文档(<https://docs.microsoft.com/zh-cn/learn/modules/get-started-with-windows-subsystem-for-linux/> 搭配Ubuntu 20.04 来安装)

更具体的深度学习环境安装可以参考 <https://blog.csdn.net/kinfey/article/details/117635067>

2. 安装Jupyter Notebook

通过pip3 安装Jupyter 运行环境

```
pip3 install jupyter
```

运行jupyter

```
jupyter notebook --no-browser
```

```
% jupyter notebook --no-browser
[I 11:49:16.367 NotebookApp] Serving notebooks from local directory:
[I 11:49:16.367 NotebookApp] Jupyter Notebook 6.4.0 is running at:
[I 11:49:16.367 NotebookApp] http://localhost:8888/
[I 11:49:16.367 NotebookApp] Use Control-C to stop this server and sh
```

关于**notebook** 的更多使用 <https://jupyter-notebook.readthedocs.io/en/stable/>

第一次运行**notebook** 会要求输入密码，这个时候你需要做的(https://jupyter-notebook.readthedocs.io/en/stable/public_server.html)

3. 关于.NETCore 版本

建议使用5.0.x 版本 <https://dotnet.microsoft.com/download>

三.关于.NET Interactive

官方的话 .NET Interactive takes the power of .NET and embeds it into your interactive experiences. Share code, explore data, write, and learn across your apps in ways you couldn't before. 就是一个更灵活的代码交互方式，专注于数据浏览与整理 (注意：如果你没有安装好Python和Jupyter环境参参考本文档的第二点完成)

1. 通过dotnet tools 安装.NET Interactive

```
dotnet tool install --global Microsoft.dotnet-interactive
```

一些提示 dotnet tool 需要配置全局PATH，否则会影响dotnet-interactive的运行 (关于 dotnet tool path 的文档可以参考 <https://docs.microsoft.com/en-us/dotnet/core/tools/dotnet-tool-install>)

2. 把.NET Interactive 绑定到Jupyter Notebook

```
dotnet-interactive jupyter install
```

3. 运行检查

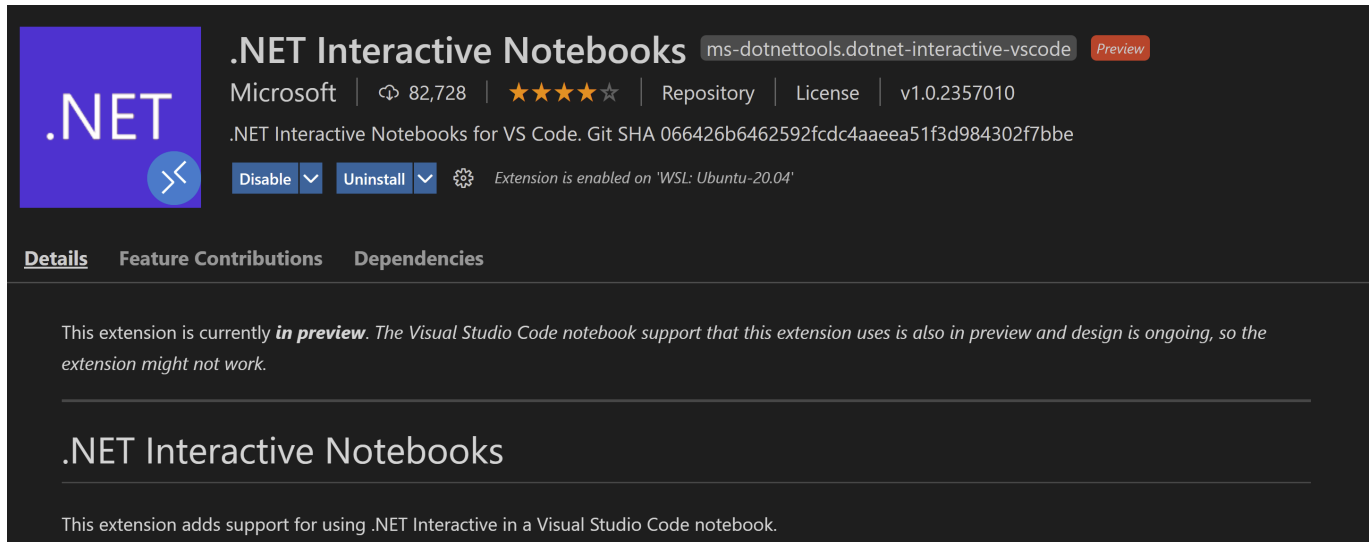
```
jupyter kernelspec list
```

```
% jupyter kernelspec list
Available kernels:
.net-csharp      /home/lokinfey/.local/share/jupyter/kernels/.net-csharp
.net-fsharp      /home/lokinfey/.local/share/jupyter/kernels/.net-fsharp
.net-powershell  /home/lokinfey/.local/share/jupyter/kernels/.net-powershell
python3          /home/lokinfey/.pyenv/versions/3.8.8/envs/pydev/share/jupyter/kernels/python3
```

四.Visual Studio Code 的.NET Interactive插件

1.安装Visual Studio Code (<https://code.visualstudio.com/>)

2.安装.NET Interactive 插件



The screenshot shows the extension page for '.NET Interactive Notebooks' in the Visual Studio Code marketplace. The extension is by Microsoft, has 82,728 downloads, and a 4.5-star rating. It is currently in 'Preview' mode. The page includes tabs for 'Details', 'Feature Contributions', and 'Dependencies'. A warning message states: 'This extension is currently **in preview**. The Visual Studio Code notebook support that this extension uses is also in preview and design is ongoing, so the extension might not work.' Below this, the extension name is repeated, followed by a description: 'This extension adds support for using .NET Interactive in a Visual Studio Code notebook.'

五. 例子

1. 基本CSharp运行

```
Console.WriteLine("Hello World , Reactor")
```

```
using System;  
using System.Text.Json;  
using System.Text.Json.Serialization;
```

```
public class ReactorInfo  
{  
    public string Name { get; set; }  
    public string Title { get; set; }  
}
```

```
var info = new ReactorInfo{ Name = "KinfeY", Title = "Reactor Guest" };
```

```
string infoString = JsonSerializer.Serialize(info);
```

```
Console.WriteLine(infoString);
```

2. Nuget引入

Blazor 引入到Notebook (<https://github.com/plbonneville/RazorInteractive>)

```
#r "nuget: RazorInteractive, 1.0.5"
```

```
#!razor
```

```
@{  
    var colors = new [] { "red", "green", "blue" };  
}  
  
<ol>  
@foreach(var color in colors)  
{  
    <li style="color: @color;">@color</li>  
}  
</ol>
```

```
var firstname = "John";  
var lastname = "Doe";  
var colors = new [] { "red", "green", "blue" };
```

```
#!razor
```

```
<p>Hello <b>@Model.firstname @Model.lastname</b>, what is you favorite a color?  
</p>  
  
<ol>  
@foreach(var color in Model.colors)  
{  
    <li style="color: @color;">@color</li>  
}  
</ol>
```