

Jupyter Notebook

交互计算的应用程序，允许你创建和共享包含实时代码、方程、可视化和解释性文本的文档。它广泛用于数据清洗和转换、数值模拟、统计建模、机器学习等领域。

优点：

- 1、助教检查实验作业方便，可以保留运行结果
- 2、同学们实验方便

交互式编程：代码块及时运行，及时查看结果

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
In [2]: arr = np.array([1,2,3])
```

单变量只需在一个单元内执行即可查看结果，无需print等函数辅助，快速检测结果

```
In [3]: arr
```

```
Out[3]: array([1, 2, 3])
```

支持 pandas 的内联显示，打印DataFrame时，会被格式化并以表格形式呈现，这使得查看数据更加直观

```
In [4]: pd.DataFrame([[1,2,3],[4,5,6],[7,8,9]],columns=['A','B','C'])
```

```
Out[4]:
```

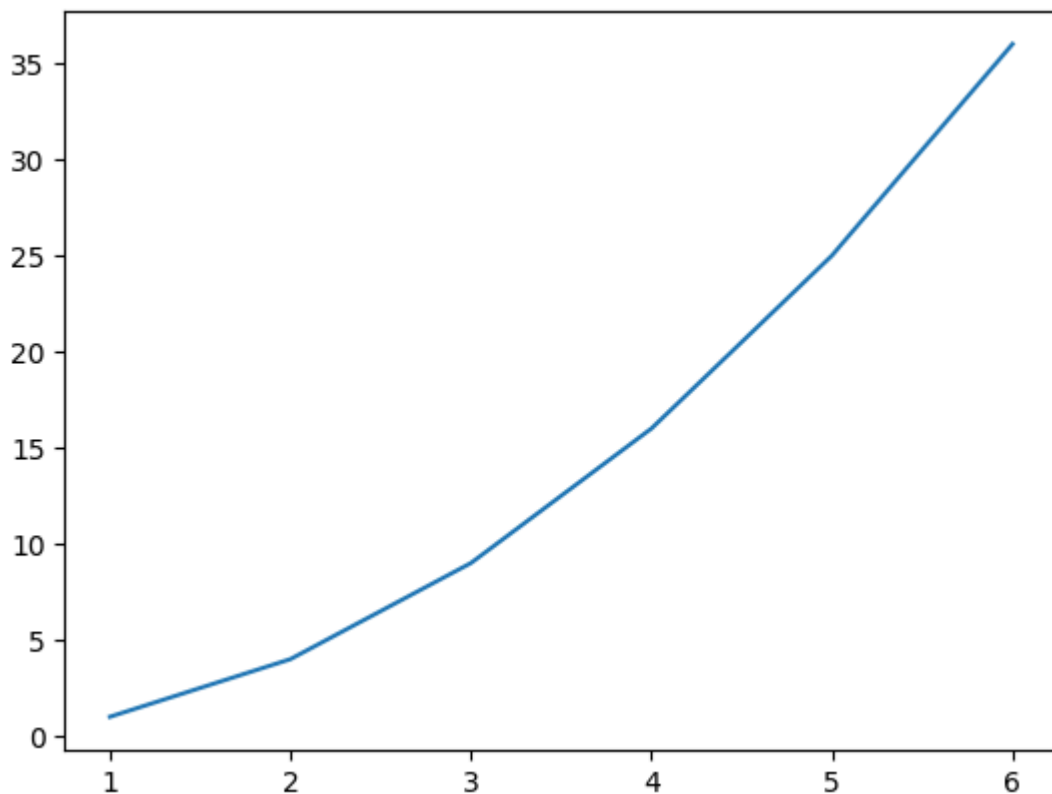
	A	B	C
0	1	2	3
1	4	5	6
2	7	8	9

Jupyter Notebook 与 matplotlib 集成度高，在 Notebook 中进行数据可视化变得非常方便。

```
In [5]: X = [1,2,3,4,5,6]
Y = [1,4,9,16,25,36]
```

```
In [6]: plt.plot(X,Y)
```

```
Out[6]: [<matplotlib.lines.Line2D at 0x27deb0e85f0>]
```



文档和代码的融合

可以在代码单元之间插入 Markdown 单元，用于添加文本、方程、图像和 HTML 内容

多语言支持，上手极其简单，轻量化适合机器学习领域的研究学习

.py 文件和 **.ipynb** 文件内容相互迁移时几乎一致，基本不需要修改代码

注：Jupyter Notebook在执行**并行计算**时会遇到限制，一段并行计算的代码在py文件下可运行，但在ipynb中是不可运行的

可扩展性、插件多

通过安装不同的扩展，可以增加 Jupyter Notebook 的功能，如代码补全、自动保存等。

Jupyter Notebook使用方式

1、Anaconda下集成的Jupyter Notebook程序

1.1 打开anaconda navigator

1.2 找到Jupyter Notebook, 点击launch

默认是安装的, 如果没有安装可以通过 pip 或者 conda 安装

pip install jupyter notebook

conda install jupyter

2、VScode 拓展 jupyter notebook

2.1 在工作路径下新建一个 **ipynb** 文件, 并打开

2.2 随便在一个单元内输入一段代码, 并执行它。

2.3 联网状态下, 根据提示安装jupyter插件 (过程可能会较慢)

补充:

2.1 在vscode拓展中搜索jupyter notebook

2.2 分别下载: Jupyter、Jupyter Cell Tags、Jupyter Keymap、Jupyter Notebook Renderers和Jupyter Slide Show

3、Pycharm 配置 Jupyter Notebook