

《Python气象应用》

周洋、张涛涛、李文锐、黄艳艳 编著

气象出版社

Markdown by 周洋

yangzhou@nuist.edu.cn



课程内容

第一章 绪论

1. Why Python及一个简单的Python代码
2. Python之禅
3. Python主要特点
4. Python与气象应用
5. Conda环境下使用Python

第二章 Python语法基础

1. 语法特点
2. Python的基础对象
3. Python的语句
4. 自定义函数

第三章 利用Python读写数据

1. 多维数据结构
2. 文本数据读写
3. 二进制数据读写

第四章 Python气象绘图

1. 图形基本要素
2. 一维图形
3. 二维图形
4. 地理信息

第五章 数据处理与分析

1. 时间处理
2. 数组变形
3. 数据插值
4. 统计分析函数
5. 动力分析函数

第六章 机器学习简介

1. 分类问题示例
2. 回归问题示例

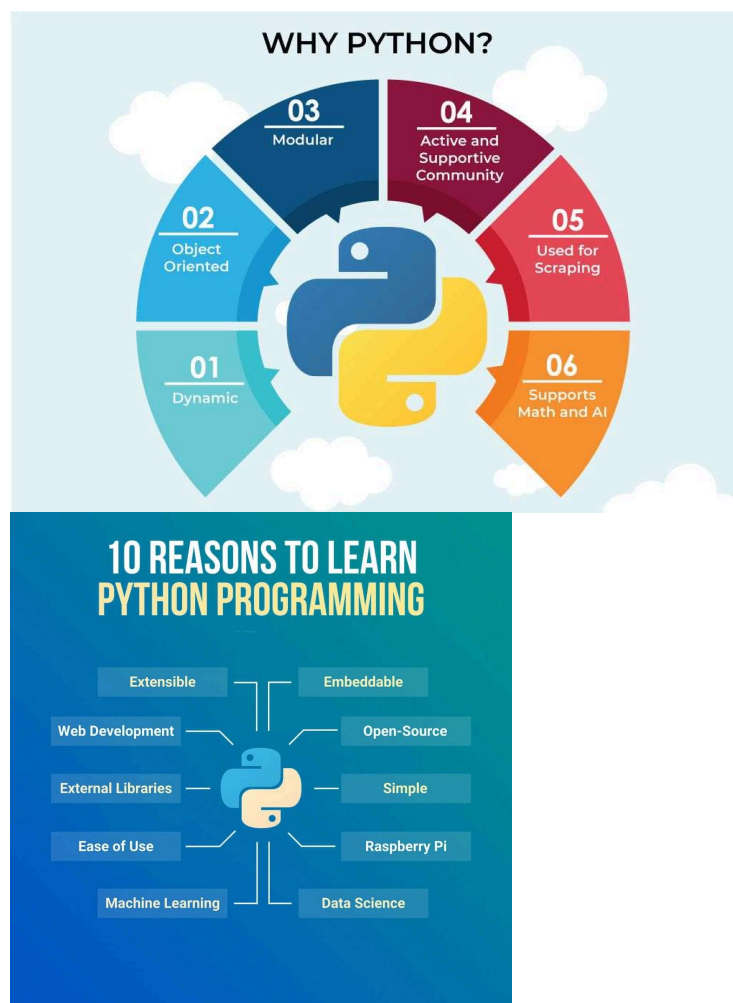
考核方式

期末考试：上机、开卷考试；

成绩占比：平时成绩50%，期末考试50%

第一章 绪论

1989年荷兰程序员Guido van Rossum发明了Python (v0.9.0) ， 随后它成为全球第二大流行语言。

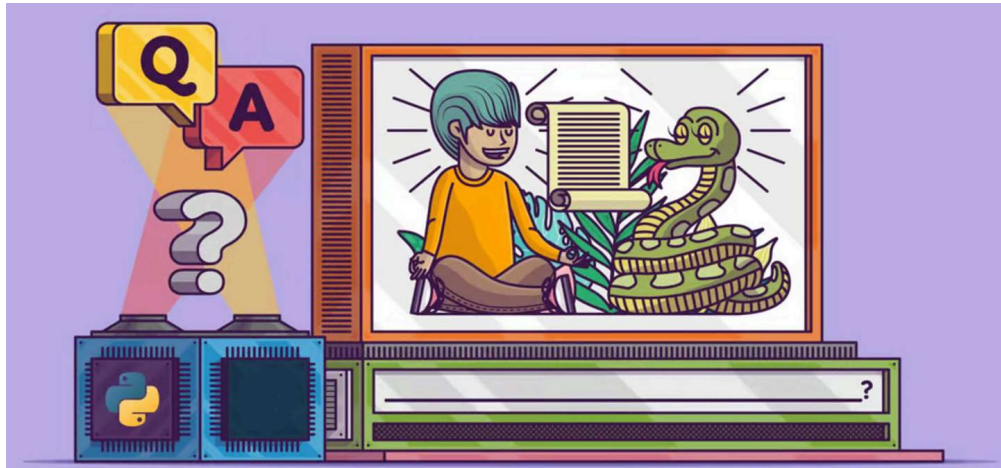


1. 程序happy.py

```
In [1]: import time
if __name__ == '__main__':
    print('Be happy on '+time.strftime('%Y-%m-%d'))
```

Be happy on 2024-12-17

2. Python之禅



作者Tim Peters 是一位资深的Python 用户;

除了“Python 之禅”,他以开发出世界上最快的排序算法 (Timsort) 闻名于计算机界。

```
In [2]: import this
```

The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!

《Python之禅》

编者译

美丽比丑陋好，清晰比晦涩强，简单胜过复杂，复杂强于烦琐；

扁平比嵌套好，稀疏比浓密强，好的代码易读；

别让特例打破规则，尽管实用战胜纯粹；

绝不容忍错误，除非你默许；
不知所措，不要臆测，去找寻最优方案。除了Python 之父，没人一蹴而就；
做比不做强，但不能不假思索；
解释不清的方案，不是个好方案。简单易懂的方案，可能是个好方案；
多用命名空间这个好想法！

3. Python主要特点

a. 开源免费 open source initiative

代码公开
免费下载、安装和使用
可修改
PyPI (The Python Package Index;
<https://pypi.org/>)



b. 移植性好

基于标准C 语言
主流计算机操作系统
出色的跨平台特性
各类平台无差别运行

c. 简洁易学

将缩进引入语法结构
格式规范、可读性强
不需编译链接
集成优点，更加灵活

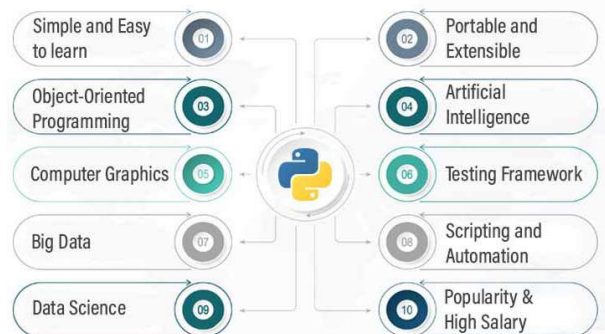
d. 功能强大

面向对象，万物皆对象
丰富的工具和函数库
多语言交互
不同工具交叉

e. 存在缺点

人无完人，金无足赤
较汇编语言慢
庞大的第三方函数库

Advantages of Python



4. Python 与气象应用

a. 常规数据读写、处理和可视化

多种数据格式
数据分析处理
绘图



NumPy



SciPy

Pandas



xarray



PyTorch

b. 预报、服务的智能化

大数据处理
人工智能
学科交叉

matplotlib



5. Conda环境下使用Python

a. 安装Conda环境

推荐在Conda 环境下安装和使用
Python
Conda虚拟环境下安装Python及其函数库
<https://www.anaconda.com/>



ANACONDA

b. Python 的安装

```
进入 (base) PS C:\Users\XXX>
安装 > conda create -n mete python 或 conda create -n mete
python=3.12
查看 > conda info --env 或 > conda info --envs
删除 > conda remove -n mete --all
激活 > conda activate mete
```

c. Python 函数库的安装

```
安装Jupyter Lab > conda install -c conda-forge jupyterlab
查看 > conda list
常用channel "-c conda-forge"
```

d. 交互式命令行

```
> python
Python 3.12.2 | packaged by Anaconda, Inc. | (main, Feb 27
2024,17:28:07)
[MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> import this
>>> exit()
```


e. 运行“.py”脚本

取消隐藏已知文件名



进入文件路径

```
> cd D:\learn_python\chp01
```

运行代码

```
> python happy.py
```

f. 在Jupyter Lab 中使用Jupyter Notebook

```
> jupyter lab
```

下面任意一个粘入浏览器

```
file:///C:/Users/...../jupyter/runtime/jpserver-6788-open.html
```

```
http://localhost:8888/lab?token=.....
```

```
http://127.0.0.1:8888/lab?token=.....
```

```
In [3]: %run happy.py
```

Be happy on 2024-11-30

```
In [4]: %load happy.py
```

```
In [4]: # %Load happy.py
import time
print('Be happy on '+time.strftime('%Y-%m-%d'))
```

Be happy on 2024-11-30

本章小结

遵循“Python 之禅”写代码可让编程事半功倍
用3种方式运行代码

命令	用途
conda create -n	建立Conda 环境
conda info --env	查看所有Conda 环境
conda activate	激活Conda 环境，退出当前环境用“deactivate”
conda install -c	在当前Conda 环境下安装各类函数库
conda remove -n (--all)	删除所有Conda 环境