

Python 程序设计

Week1: Introduction

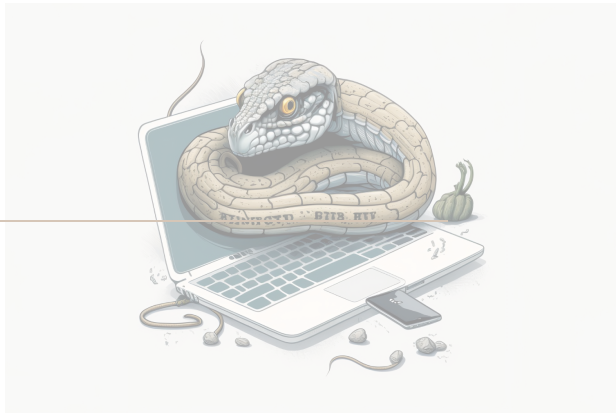
CHEN Zhongpu

SWUFE, Fall 2024

School of Computing and Artificial Intelligence

zpchen@swufe.edu.cn

1. 课程介绍



关于我：陈中普

研究方向

时空数据库，大数据分析，向量数据库，大模型应用

办公室

格致楼 J-310

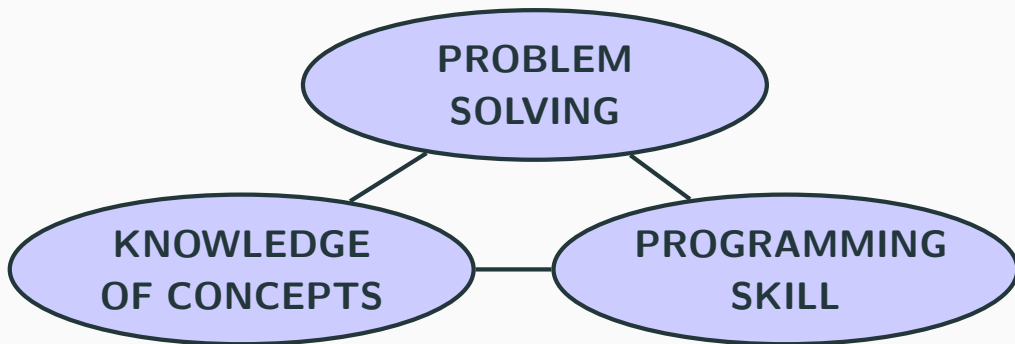
主页

<https://zhongpu.info>

关于课程

Not Only Python

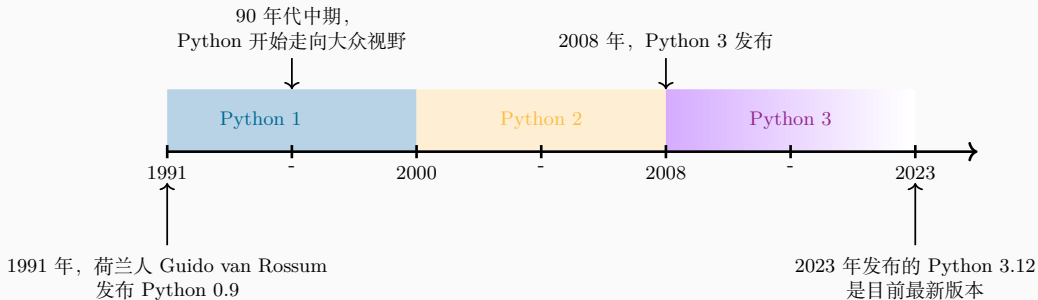
本课程面向没有编程基础的学生，将围绕着 Python 语言，讨论程序设计的**术与道**。



关于 Python

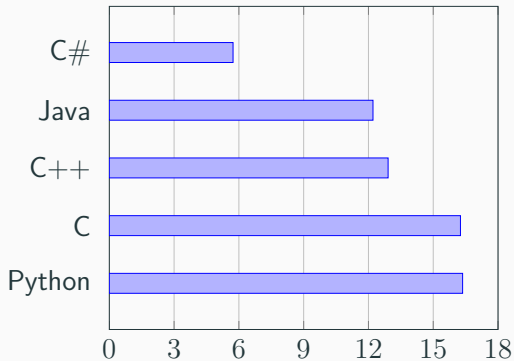
Python 官网

Python is a programming language that lets you work quickly and integrate systems more effectively.

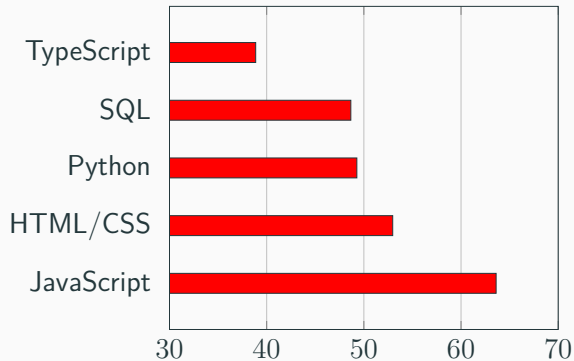


Pros

Python 被广泛应用于科学计算、Web 开发、系统运维、人工智能等领域，是目前最流行的编程语言之一。



Tiobe 2023 年编程语言排名



StackOverFlow 2023 年语言排名

Computer Programming for Everybody

1999 年，Guido van Rossum 在Computer Programming for Everybody中写道：

- 一门简单直观的语言并与主要竞争者一样强大。
- 开源，以便任何人都可以为它做贡献。
- 代码像纯英语那样容易理解。
- 适用于短期开发的日常任务。

Cons

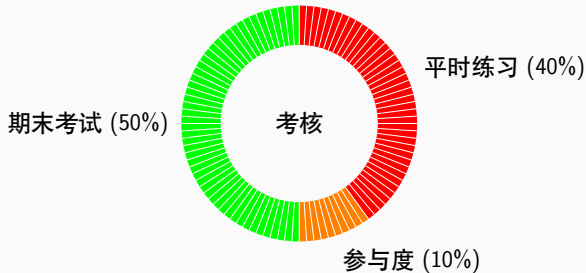
但是trade-off总是存在的，Python 也有很明显的缺点。

Language	Time
C	1.00
Rust	1.04
C++	1.56
Ada	1.85
Java	1.89
Go	2.83
Python	71.90
Lua	82.91

论文[Energy Efficiency across Programming Languages](#)表明，在运行时间方面，Python 在主流编程语言排名倒数第二。

预备知识、考核标准

原则上几乎不需要预备知识，掌握**阅读文档**（尤其是英文文档）和**安装软件**的能力即可。<https://docs.python.org/3/index.html>



严禁抄袭！多练习！

课程资源

参考资料:

- Eric Matthes. **Python Crash Course**. Third Edition. 2022.
- **课程代码及课件**



搜索工具

- StackOverFlow、Google
- ChatGPT, Claude

获取知识的能力比知识本身更有价值。

编程工具

本课程推荐使用Python 3.10+。

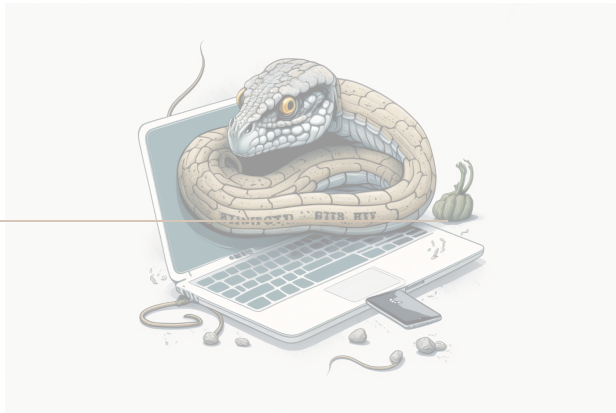
IDE 或编辑器（要求掌握）：

- PyCharm（适合专业、大型项目）
- VSCode（适合轻量级、小型项目）
- Jupyter（适合演示、学习）

代码 AI 助手（不要求掌握）：

- <https://codeium.com>
- GitHub CoPilot

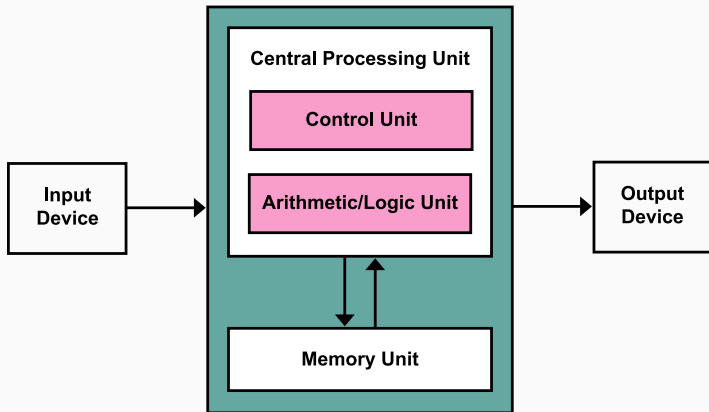
2. 什么是计算？



计算、知识与编程

思考

计算机最重要的两个功能是什么？



尽管现代计算机的功能强大（TB 级别的存储和 GHz 级别的 CPU），但是计算机只能执行程序员告诉它的任务。换言之，我们需要将**知识注入计算机**，而知识的载体就是编程语言。

尽管现代计算机的功能强大（TB 级别的存储和 GHz 级别的 CPU），但是计算机只能执行程序员告诉它的任务。换言之，我们需要将**知识注入计算机**，而知识的载体就是编程语言。

知识的分类：

- **声明式知识** (declarative knowledge)：WHAT
- **过程式知识** (imperative knowledge)：HOW

编程语言与自然语言

Each programming language has a set of primitive constructs, a syntax, and a semantics.

	自然语言	编程语言
基本构造	词汇 (cat)	基本数据 ("cat"、1) 和 简单操作符 (+)
语法	I love cats. cats love.	1 + 2, 1 2 +
语义	I loves cats.	1 + "cat"

从 Hello World 开始

```
print("Hello World!")
```

即使本地没有安装 Python，也可以通过浏览器在<https://replit.com>上运行 Python 代码。

- <https://www.online-ide.com/>
- <https://www.programiz.com/python-programming/>
- <https://www.jdoodle.com/>

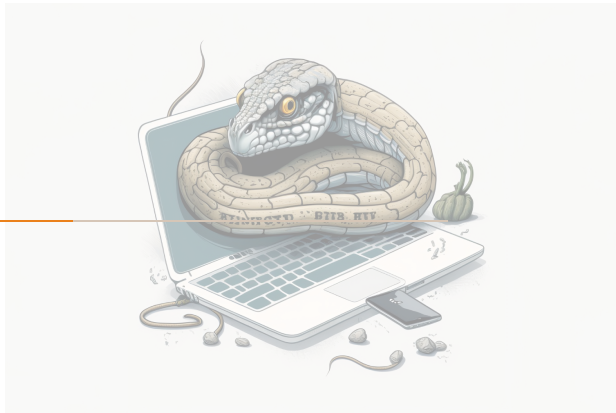
```
#include <stdio.h>
#include <stdlib.h>
int main(void) {
    printf("Hello World!\n");
    return EXIT_SUCCESS;
}
```

```
public class Test {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

几个案例

- 获取股票价格
- 生成二维码
- 生成词云
- 聊天机器人

3. Python 程序



基本概念

程序

程序 (program) 由一系列定义 (definition) 和命令 (command) 组成。

```
print("Hello, world!")
```

程序所保存的文件被称为**源代码** (source code), 在 Python 中也称**脚本** (script), 最后交给 Python**解释器** (interpreter) 执行, 也可以通过 Shell 执行。

对象

对象

程序是用来操作数据**对象** (data objects)。对象由**类型** (type) 和值 (value) 构成。不同类型的对象能够执行的操作是不同的。

对象

对象

程序是用来操作数据**对象** (data objects)。对象由**类型** (type) 和值 (value) 构成。不同类型的对象能够执行的操作是不同的。

Python 中一切皆对象！换言之，均属于某个类 (class)。

对象可大致分为两类：

- **标量** (scalar)：不可再分的对象，如整数、浮点数、布尔值，属于**基本数据类型**。
- **非标量**：可再分的对象，如字符串、列表、字典。

基本数据类型

<code>int</code>	表示整数 (integer), 比如 42
<code>float</code>	表示浮点数 (floating-point number), 比如 3.14
<code>bool</code>	表示布尔值 (boolean), 取值为 <code>True</code> 和 <code>False</code>
<code>NoneType</code>	表示空值 (None), 只有一个取值 <code>None</code>

`str` 是最简单的**非标量**类型之一, 表示字符串 (string), 比如 "Hello, world!"。

类型相关操作

查看类型:

```
>>> type(42)
>>> type(3.14)
```

检查类型:

```
>>> isinstance(42, int)
>>> isinstance(3.14, int)
>>> isinstance(True, int)
```

转换类型:

```
>>> int(3.14)
>>> float(42)
>>> float("3.14")
>>> str(3.14)
```

表达式

表达式

对象和操作符（operator）的组合称为**表达式**（expression），表达式的值也是对象。

算术操作符有：+、-、*、/、%、**；算术表达式的基本格式是：

< 对象 > < 操作符 > < 对象 >

其中，/的结果总是浮点数（整除需要使用//）；%的计算不同编程语言不一样；** 在其他编程语言中很少见。

除法和取余

与其他编程语言相比，Python 中的整除（又称 floor division）和取余操作符有点特殊。

```
>>> 5 / 2
```

```
>>> 5 // 2
```

```
>>> 5 / -2
```

```
>>> 5 // -2
```

```
>>> 5 % 2
```

```
>>> 5 % -2
```

```
>>> -5 % 2
```

思考

下面表达式的结果是什么？

```
>>> 1 + 2 * 3
```

```
>>> (1 + 2) * 3
```

```
>>> +1
```

```
>>> ++++1
```

```
>>> -1
```

```
>>> ----1
```

```
>>> 1 ++++++ 1
```

变量

变量

值可以通过赋值 (assignment) 绑定到一个变量 (variable); 变量是一个标识符 (identifier), 用来引用值。

In Python, a variable is just a name, nothing more.

```
>>> a = 1
>>> a
>>> a = 2
>>> a = a + 1
>>> b = a
>>> c = d = 42
```

前 4 个是强制，后 2 个是建议。

1. Variable names must start with a letter or underscore not a number.
2. Variable names can only contain letters, numbers, and underscores. No other characters are allowed.
3. Variable names are case-sensitive.
4. Avoid using **reserved keywords** as variable names.
5. Variable names should be descriptive and meaningful.
6. Use lowercase letters for variable names, separate words with underscores.

练习与思考

考虑下面的 Python 代码，请定义一个变量来表示圆的面积：

```
pi = 3.14159  
radius = 2.2  
# compute the area of an circle
```


分支

分支 (branching) 是程序中最基本的结构之一，也称为选择结构 (selection)。

```
age = 45
if age > 40:
    salary = 20000
else:
    salary = 18000
print(salary)
```

在上面程序的基础上，进一步掌握比较和逻辑操作符号。

Tab 还是 Space

通过上面程序，可以看到**缩进**（indentation）在 Python 中的重要性。

In Python, the choice between using tabs or spaces for indentation is a matter of personal preference. However, it is highly recommended to be consistent in your choice throughout your codebase. **PEP 8**, the official style guide for Python code, recommends using spaces for indentation and suggests using 4 spaces per level of indentation. This is the widely accepted convention in the Python community.

三元操作符

请搜索资料，使用 Python 的三元（ternary）操作符实现下面的代码：

```
age = 45
if age > 40:
    salary = 20000
else:
    salary = 18000
```

Conclusion

1. 什么是计算
2. Python 的特点
3. 基本数据类型，表达式，变量，分支

本周任务

- 安装 Python 3.10+
- 安装 Visual Studio Code, 并安装 Python 插件
- 通过 Python Shell 或 VSCode 练习 Python
- 在 Python Shell 中输入 `import this`, 并翻译显示的内容

安装 JupyterLab

建议创建并启动[虚拟环境](#)后再安装第三方包。`python -m venv ThePath`

第 0 步

配置国内的 PIP 镜像，如[清华大学](#)、[上海交通大学](#)。

第 1 步：安装

命令行输入：`pip install jupyterlab`

第 2 步：启动

命令行输入：`jupyterlab`