

代码

周潇翔

摘要. 这里总结自己学过的代码供查阅。为啥不用英文？英文的参考文献浩如烟海，也不差我一个啊。对数学系的同学而言，代码的逻辑并不难，大家不会的只是格式而已。第一节横向介绍我们需要啥，之后纵向对每一种语言给出对应的代码。

在这份文档的编译中第一次学 python 和 Github, 可以说是紧跟潮流
23333

1. 代码需求

大部分的语言都需要：

- 安装 + 初始代码 (Halloworld)
- 基本逻辑
- 调试
- 参考文档
- (想保留的) 例子

以下是具体需求：

1.1. 安装 + 初始代码.

- 简要说明该语言的目的
- 说明自己使用何种编译器
- 解释该语言的结构 (基本框架)
- 使用该语言在屏幕中打出”Halloworld”
- 必要时给出英文注释

1.2. 基本逻辑.

- 数据结构类型 (数字、字符串、其他结构)
- 基本四则运算 + mod (若数据结构中包含矩阵，则需要矩阵的各类运算；)
- 条件语句
- 循环语句

- 函数

1.3. 调试.

- 快捷键
 - 运行代码
 - 注释方式及快捷键 (单行注释 + 多行注释)
 - 自动补全功能
 - 自动对齐功能
 - 其他快捷键
- 如何获得帮助
- 控制输出
- 如何设置断点
- 控制输入

1.4. 参考文档.

- 官方文档
- 民间优秀文档

超出科大 C 语言的知识: 编程范式 (Programming paradigm)、方法 (method)

2. PYTHON

2.1. 安装 + 初始代码. Python 是一门高级的编程语言。他有许多的标准模块 (standard module).

2.2. 基本逻辑. 与 C 语言不同, Python 不需要声明变量。当有赋值时不输出结果。

数据类型在这里看到。¹

对于数字, Python 不仅有 int 和 float, 还有 Decimal, Fraction and complex numbers 这些奇葩的变量。Task: 学会 Decimal, Fraction.

```
1 >>> complex('1+2j')*complex('1+3j')
```

四则计算像自然计算一样自然, 不过带余除法用//, 余数用%, 幂次用**.(好符号)²

```
>>> a,b=8,13 # a++ is not allowed in python
>>> a ** (b-1) % b # verify the Fermat's little theorem
```

¹你需要知道那些类型是可改变的 (mutable);

²请小心使用负数的带余除法。

计算器上的 Ans 记为_, `round(0.142857,1)`给出 0.1

条件语句和循环语句的书写规范详见[这里](#)。以下是计算素数的例子。

```
>>> for n in range(2, 100):
...     for x in range(2, n):
...         if n % x == 0:
...             break
5 ...     else:
...         # loop fell through without finding a factor
...         print(n, end=",")
```

2.3. 调试. 控制输入使用函数()。

2.4. 参考文档. 官方文档

2.5. (想保留的) 例子. 这个例子计算 \mathfrak{gl}_n 幂零轨道的维数.

```
# this computes dimension of orbits of nilpotent elements
young_diagram = [2,1,1,1,1] # Here is the input
n=len(young_diagram)
a,b=0,0
5 for i in range(n):
    a=young_diagram[i]
    for j in range(n):
        if young_diagram[i]<young_diagram[j]:
            b=young_diagram[i]
10    else:
        b=young_diagram[j]
c=a**2 -b
print(c)
```

3. SAGE

3.1. 安装 + 初始代码. Sage 是专为数学家设计的程序, 基于 python, 通过集成大量数据库, 使用其中的数学函数来简化编程难度。我使用的是网上在线的编辑器CoCalc, 在 New 中生成 Sage worksheet. 你可以在每一行前添加sage: 或者不加。在最开始输入需要的宏包, 然后直接进行计算即可。

```
sage: print("Halloworld!233") #The code is the same as
python
```

3.2. 基本逻辑. 本节的基本内容参见[这里](#)。

3.3. 参考文档. 官方文档

Sage V9.1 中文文档

3.4. (想保留的) 例子. 这个例子能帮助我计算 Dynkin quiver 的所有不可约表示.

```
sage: Q = DiGraph({1:{2:['a1']},2:{3:['a2']},4:{3:['a3']},5:{4:['a4']},6:{3:['a5']}}).path_semigroup()
sage: M = Q.I(GF(11),3)
sage: M
4 sage: tauM = M.AR_translate()
sage: tauM
```

4. LATEX

4.1. 本文参考文献. listings 的具体设置

L^AT_EX 中如何自动补全代码? 不容易。

本文 python 代码格式来源

4.2. 参考文档. 刘海洋的书: L^AT_EX 入门

5. 电脑快捷键 (中级)

部分参见Chrome 快捷键。

- ctrl+win+left: 切换桌面
- alt+(shift)+tab: 切换任务栏上的程序
- ctrl+(shift)+tab: 切换标签页

REFERENCES

SCHOOL OF MATHEMATICAL SCIENCES, UNIVERSITY OF SCIENCE AND TECHNOLOGY
OF CHINA, HEFEI, 230026, P.R. CHINA,

Email address: email