编程通用技能

这个文档里记录的知识，学校不一定会教，但是作为开发者来说一定要会用。

目录

[1. 版本管理工具——Git 与 Github 1](#_Toc156423063)

[1.1. 工作原理 2](#_Toc156423064)

[1.2. 使用方法 2](#_Toc156423065)

[1.3. 常用命令 2](#_Toc156423066)

[1.4. 文件忽略列表 2](#_Toc156423067)

[2. VScode 2](#_Toc156423068)

[3. 基本算法 2](#_Toc156423069)

[3.1. 排序算法 2](#_Toc156423070)

[1.1.1. Python自带排序函数 2](#_Toc156423071)

[1.1.2. 插入排序 3](#_Toc156423072)

[1.1.3. 选择排序 3](#_Toc156423073)

[1.1.4. 冒泡排序 3](#_Toc156423074)

[4. 变量名常用单词缩写 4](#_Toc156423075)

# 版本管理工具——Git 与 Github

## 工作原理

## 使用方法

|  |
| --- |
| echo "# REPOSITORY NAME" >> README.md  git init  git add README.md  git commit -m "first commit"  git branch -M main  git remote add origin git@github.com:USERNAME/ REPOSITORY NAME.git  git push -u origin main |

## 常用命令

## 文件忽略列表

# VScode

# 基本算法

## 排序算法

在编程领域，排序算法是非常重要的基础算法，常用于优化、数据处理以及作为其他算法的步骤之一。在Python中，列表（list）等数据结构有自带的排序成员方法（method）。

下面的具体代码示例全部以数字从小到大排序为例。

### Python自带排序函数

在Python中有两种自带的排序函数，若想直接改变原列表的排序，可以用列表附属方法list.sort()，若要创建一个新排列好的数组，用内置函数sorted()；所有可迭代数据类型（iterable）都可以使用。

两个函数的用法类似，且都有两个可选输入参数key和reverse，key接受一个映射函数，将根据元素的映射值（比如长度）进行排序，reverse规定是否倒序。

若不指定key：

·数值类型按从小到大排序

·字符串按Unicode排序

·同时包含不同数据类型会报错

### 插入排序

每一轮从左到右依次选择一个未排序元素插入到已排序序列中次序恰当的位置。

·Python代码：

|  |
| --- |
| def insertion\_sort(list):      for i\_selected, selected in enumerate(list): # Select the inserting item          for i\_moved in range(i\_selected - 1, -1, -1): # Comparison              if selected >= list[i\_moved]:                  list.pop(i\_selected)                  list.insert(i\_moved + 1, selected)                  break              if i\_moved == 0: # The selected item is the smallest                  list.pop(i\_selected)                  list.insert(0, selected)      return list |

### 选择排序

每一轮从左到右找出最小的未排序元素，这就是该轮中已排序部分的最大值，把它放在已排序部分的最右侧。

·Python代码：

|  |
| --- |
| def selection\_sort(list):      for i\_pos, pos in enumerate(list): # Mark the rightmost of sorted          least = pos          i\_least = i\_pos          for i\_selected in range(i\_pos, len(list)): # Find the least              selected = list[i\_selected]              if selected < least:                  least = selected                  i\_least = i\_selected          temp = pos          list[i\_pos] = list[i\_least]          list[i\_least] = temp      return list |

### 冒泡排序

从左到右，相邻的元素之间每次把较大的对调到右边，这样每一轮就会把最大的元素移到序列的右侧。序列完成排列的顺序是从右到左，从大到小。

|  |
| --- |
| def bubble\_sort(list):      for i\_complete\_pos in range(len(list) - 1, -1, -1): # Mark the leftmost of sorted          for i\_shift in range(0, i\_complete\_pos): # Shift one by one              if list[i\_shift + 1] < list[i\_shift]:                  temp = list[i\_shift]                  list[i\_shift] = list[i\_shift + 1]                  list[i\_shift + 1] = temp      return list |

# 变量名常用单词缩写

|  |  |
| --- | --- |
| 原单词 | 缩写 |
| source | src |
| destination | dst |
| index | idx (i) |
| iterator | i |
| previous | prev |
| current | curr |