

Python基础语法文档以及配套练习（一）

Author: Limzh

Section 1. 序言以及如何运行Python文件

这篇文档是快速上手Python基础语法的入门级资料，我结合自己的经验写出来的总结。不涉及深层次的东西，读完这篇文档之后，最终目的是**运行一个python文件、理解缩进和注释的重要性**。

1. 运行Python文件

当我们拿到后缀名字为 `.py` 的Python文件之后，如何在 **终端** 运行该文件是迈出学习Python的第一步。最基本的方法如下：

- 打开终端，进入对应文件的目录(使用cd等命令)
- 假设我们要运行的文件名字为 `test.py`
- 输入 `python test.py`
- 完成运行

补充：关于如何打开终端，在windows下使用win+R键调出运行框。在框内输入 `cmd`，回车进入终端。



或者我推荐使用VScode，使用快捷键 `ctrl+k+o` 进入打开文件夹，用鼠标选中打开文件所在文件夹.之后使用快捷键 `ctrl+shift+~` (`~` 键位于数字1键的左边)，直接打开终端。之所以这更推荐，是因为它直接避免使用 `cd` 命令。

练习：

1. 请尝试运行 `basic\section1\run.py` 并跟随指令完成程序.
2. 用Vscode打开相应python文件，使用快捷键 `F5`，直接运行程序.
3. 依据，File - Reference - keyboard shortcuts （Ctrl + K + Ctrl + S）-- 尝试修改run Code的快捷键

Section 2. 注释和缩进

1. 注释

注释(comment)：注释是程序开发者写在代码文件任何位置的，增强代码可读性的自然语言解释。它是一段永远不会被编译执行的无效代码。它的规定如下：

- `#` 字符之后的位于该行的所有代码都为注释内容
- 注释既可以单独另起一行，也可以写在一行有效代码的尾部。

`#` 这是单独一行的注释，可以写在文档的任何位置。

`x = 10 #` 这是一行写在有效代码尾部注释。

tips: 我习惯注释时候尽量另起一行，并写在所要注释代码的前端。并且 `#` 字符之后要空一格空格。这不是必须的，但是是美观的。美观性和整洁性是写代码的重要素养。

2. 缩进

缩进(indentation)：缩进是python所独有的决定代码 作用域 范围的语法。关于作用域(scope)是什么，暂时不重要。传统的C/C++使用花括号({...})决定一个函数、模块或者循环等结构体的范围。与之类似的，python使用 **缩进(indentation)** 的概念去规定：谁和谁属于同一个作用域。可以简单地理解为，

处于相同缩进级别的代码属于同一个“花括号”

- **缩进级别** 被定义为，每行代码开头的空格数目。
- 缩进既可以用四个空格也可以用tab（制表符，位于Q键的左边）来形成，但是你要么都用空格，要么都用制表符，不可以混用。
- 每一个引发缩进级别的代码之后使用 `:` 来引导，也就是说，在一个函数或者一个for 循环之后，需要添加冒号来告诉程序我们进入了更第一级的 作用域。

```
# 第一缩进级别
def main():
    # 第二缩进级别
    x = 10
    y = 20
    for i in range(10):
        # 第三缩进级别
        print(x+y)
```

分析上面一个实例，在上面的实例中存在三个缩进级别。代码相当于C++中的，

```
int main(void){
    x = 10;
    y = 10;
    for(int i = 0; i < 10; i++){
        printf("%d", x+y);
    }
}
```

练习：

- 为 basic\section2\Fix_indentation_problem.py 添加 头部注释。
- 头部注释的要求如下(第一行第二行不需要填充)：

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# @Time    : ${DATE} ${TIME}
# @Author  : Yanru Yang
# @Site    : ${SITE}
# @File    : ${NAME}.py
# @Software: ${PRODUCT_NAME}
```

在每一份属于自己的文件写下这些注释有利于保持原创性，并且给自己以成就感。

- 噢，天哪！`Fix_indentation_problem.py` 存在运行时候的 `error`，尝试一次性修复它直到它能正确运行(我们或许并不能看懂程序写了什么，但看完这个section之后，你理应知道在哪些地方需要缩进.)。

当你发觉修复困难的时候，从以下几个方面思考：

1. 报错给了什么有价值的信息吗？
 2. `:` 冒号是不是遗漏了？
 3. 空格和制表符是不是混用了？
 4. 缩进的格数正确吗？
- 请根据这个网址的信息，实现[头部注释自动生成](#)。

请在完成上面的小练习之后，点击图片。

