

Basic Data Structure and Variable in Python ——Python的基础数据结构以及变量

学习任何语言，尝试用这门语言构建逻辑，都离不开最基本的素材——基础数据结构以及变量

TIP:如果你不知道什么是变量请先阅读[这里](#)，反之您可以跳到下一个小标题

什么是变量？你还记得你初中的时候学过的直线方程吗？

like this:

$$y = 3x + 5, x \in R$$

这是一个函数，函数中的 x ， y 就是一个变量。

他是可以任意改变的，并且有自己的取值范围

x 可以等于任何一个实数，可以任意的**变化**，所以是一个**变量**。

在程序中我们可能需要用一些可以随意更改的值，来帮助我们实现逻辑

比如

```
1 int add (int a, intb)
2 {
3     return a + b;
4 }
```

你不需要关心这个是什么意思，你只需要知道这个功能可以帮你完成两个整数的加减法

变量就是这样使用的

一个程序中可能需要使用多个变量。

如果你学过一些其他的语言or阅读完了上面的部分

如果你有其他的静态语言的基础，你可能会熟悉

```
1 int x = 0;
2 double y = 3.01;
```

这样的写法，使用这种C风格的变量声明/初始化，你需要先书写变量的类型，然后是变量名。

Python中使用变量，不需要显式地写出变量的类型，即，你不需要写 `int`，`double` 之类的

直接使用 `x = 0` 即可声明并给`x`赋值

C风格的书写中你可以声明一个不使用且不初始化的变量：`int x;`

`x` 可能被声明以后就再也不被使用

但是Python是不允许这种风格的

在Python中你当然可以写 `x = 0`，然后不去使用这个 `x`，但是Python不支持单独书写 `x`

即：

C允许

```
1 | #include "stdio.h"
2 | int main()
3 | {
4 |     int x;
5 |     return 0;
6 | }
```

但是Python不允许

```
1 | x
```

如果你直接这样写，Python的解释器会给你一个 `NameError: name 'x' is not defined`

这样的错误

Python允许你这样写

```
1 | x = 0
```

直接赋值，但是如果 `x` 不被使用的话建议不要写出来

另外一个值得注意的是Python的命名风格

如果你学过Java，C#这一类的语言，应该听过大名鼎鼎的驼峰命名法

即

```
ThisIsAVar
```

这种写法在Python中并不推荐，Python建议你使用蛇形命名法

即

```
this_is_a_var
```

虽然变量名更长了，但是阅读性有着显著的提升