

# A Humble Attempt: How Concise Can Python Code Be?

Jianlin Su

October 7, 2014

This article is very basic; experts may skip it...

Python is renowned for its development efficiency. High development efficiency naturally means it can achieve more functions with less code. So, for the same problem, how concise can Python code be? And how do we balance development efficiency and execution efficiency? Having studied Python for a few months and knowing a little, I would like to make a humble attempt to demonstrate this here.

In this example, let's program to calculate:

$$\sum_{n=0}^{10} n^2$$

This is, of course, a very simple exercise. Following the general approach, the most natural code would be:

```
s=0
for i in range(11):
    s = s + i**2

print(s)
```

However, if we pursue code conciseness, we only need to write:

```
print(sum([i**2 for i in range(11)]))
```

It's done in just one line! This is one of the strategies for streamlining code and improving efficiency in Python and other scripting languages: utilize arrays and built-in functions, and try to avoid writing loops yourself.

However, although the latter code is concise, its efficiency is not particularly high. The reason is that lists in Python are very flexible objects; their elements do not necessarily have to be of the same type. For example, code like `a=[1,[2,3]]` is valid. In this case, the first element `a[0]=1` is an integer, and the second element `a[1]=[2,3]` is a list. The downside of this flexibility is that every time an element in the list is accessed, the interpreter must check its data type, which leads to a decrease in efficiency.

The solution is to use actual arrays from `numpy` to replace lists:

```
import numpy as np
print((np.arange(0,11)**2).sum())
```

This way, both execution efficiency and development efficiency are taken into account.

Of course, more techniques need to be summarized continuously through practice. I welcome any advice from readers.

**When reposting, please include the original link:** <https://kexue.fm/archives/2971>

**For more detailed reposting matters, please refer to:** Scientific Space FAQ