Python 的设计哲学

2014年03月02日

在 python shell 中输入：import this

The Zen of Python, **by** Tim Peters

Python 的设计哲学，作者：Tim Peters

Beautiful **is** better than ugly.

优雅胜于丑陋。

Explicit **is** better than **implicit**.

明确胜于含糊。

Simple **is** better than complex.

简单胜于复杂。

Complex **is** better than complicated.

复杂胜于繁琐。

Flat **is** better than nested.

扁平胜于嵌套。

Sparse **is** better than dense.

间隔胜于紧凑。

Readability counts.

可读性很重要。

Special cases aren't special enough to break the rules.

即使假借特殊之名，也不应打破这些原则。

Although practicality beats purity.

尽管实践大于理论。

Errors should never pass silently.

错误不可置之不理。

Unless explicitly silenced.

除非另有明确要求。

In the face of ambiguity, refuse the temptation to guess.

面对模棱两可，拒绝猜测。

There should be one-- and preferably only one --obvious way to do it.

用一种方法，最好是只有一种方法来做一件事。

Although that way may not be obvious at first unless you're Dutch.

虽然这种方式开始时并不容易，除非你是 Python 之父。

Now **is** better than never.

但从现在就开始这么做，总比永远都不做好。

Although never **is** often better than \*right\* now.

尽管经常有时 “没有做” 反倒比 “现在立马做“ 结果要好。

If the implementation **is** hard to explain, it's a bad idea.

如果一个实现不容易解释，那么它肯定是个坏主意。

If the implementation is easy to explain, it may be a good idea.

如果一个实现很容易解释，那么它也许是个好主意。

Namespaces are one honking great idea -- let's **do** more of those!

就像命名空间就是一个绝妙的想法，应当多加利用。