

译

1 Beautiful is better than ugly.

代码应该是优雅而非丑陋的。这强调代码的美观和可读性。

2 Explicit is better than implicit.

直接明确的代码比含糊不清的代码更好。避免使用难以理解的技巧或隐式的行为。

3 Simple is better than complex.

简单的解决方案比复杂的更好，除非简单的方法无法满足需求。

4 Complex is better than complicated.

如果需要复杂性，它应该是必要的，而不是因为做事的方法本身就很复杂。

5 Flat is better than nested.

尽量避免多层嵌套的代码块，扁平的代码结构更容易理解。

6 Sparse is better than dense.

代码应该有适当的间隔和组织，不应该过于紧凑。

7 Readability counts.

代码的可读性非常重要。

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

即使遇到特殊情况，也不应违反已建立的编程规范。

9 Although practicality beats purity.

虽然理想很重要，但实用性更为关键。

10 Errors should never pass silently.

错误不应被忽略。

11 Unless explicitly silenced.

除非你有明确的理由要忽略某个错误。

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

当代码的意图不明确时，不要去猜测它的行为。

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

有一个明确的方法来做某事总是好的。

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

这句话是一个玩笑，意思是：“那种显而易见的方法在开始时对大多数人来说可能不是那么明显，但如果你是荷兰人（像 Python 的创始人 Guido van Rossum 那样）你可能会很容易看出来。”

为什么会这样说呢？因为 Python 的某些设计决策可能在初看时并不直观或显而易见，但一旦你深入了解，就会发现它们的实际价值和智慧。这句话用一种轻松幽默的方式表达了这一点，并特意提到“荷兰人”是为了向 Python 的荷兰创始

人 Guido van Rossum 致敬。

15Now is better than never.

采取行动总比不做任何事情要好。

16Although never is often better than right now.

但有时等待直到有更好的解决方案是更明智的。

17If the implementation is hard to explain, it's a bad idea.

如果某个实现方案很难解释，那么它可能不是一个好主意。

18If the implementation is easy to explain, it may be a good idea.

如果实现方案容易解释，那么它可能是个好主意。

19Namespaces are one honking great idea -- let's do more of those!

命名空间是一个非常好的概念，我们应该更多地使用它。

Python 是为了解决什么问题而出现的？

将 Python 称为 "胶水语言" 是因为它具有将不同的系统或组件连接在一起的能力

1 提供一种简洁、明确的编程方式：与某些编程语言相比，Python 旨在提供一种简洁、不冗长的代码风格。这是通过强制缩进、有限的语法结构和明确的编程约定来实现的。

2 创建一种易于阅读和编写的语言：Python 的设计哲学是“只有一种方法来做某事”，尽管实际上可能存在多种方法。但这种哲学鼓励编程者编写易于他人理解的代码。

3 提供一种广泛用途的语言：Python 是一种通用编程语言，可以用于 Web 开发、数据科学、人工智能、桌面应用程序、硬件编程等多种应用。

4 让编程变得有趣并为初学者降低学习难度：Python 的语法简单，使得初学者更容易上手。

5 作为脚本语言或粘合语言：Python 的初衷之一是作为一种脚本语言，可以用来自动化日常任务，或作为不同系统和应用之间的“粘合”代码。

6 模块和包的支持：Python 支持模块和包，这使得代码的重用和组织变得更加容易。

7 交互性：Python 提供了交互式解释器，允许开发者实时执行代码并看到结果，这对于学习和原型设计非常有用。

Guido van Rossum 创建 Python 时并没有想到它会变得如此流行和广泛使用。

他只是想解决某些编程问题，提供一种比当时

若有收获，就点个赞吧