* The Zen of Python is a collection of 19 "guiding principles" for writing computer programs that influence the design of the Python programming language.
* Python code that aligns with these principles is often referred to as "Pythonic".
* Software engineer “Tim Peters” wrote this set of principles in 1999.
* Tim Peter’s list left open a 20th principle "for Guido to fill in", referring to “Guido van Rossum”, the original author of the Python language. The vacancy for a 20th principle has not been filled.
* The principles are listed as follows:

1. Beautiful is better than ugly.
2. Explicit is better than implicit.
3. Simple is better than complex.
4. Complex is better than complicated.
5. Flat is better than nested.
6. Sparse is better than dense.
7. Readability counts.
8. Special cases aren't special enough to break the rules.
9. Although practicality beats purity.
10. Errors should never pass silently.
11. Unless explicitly silenced.
12. In the face of ambiguity, refuse the temptation to guess.
13. There should be one-- and preferably only one --obvious way to do it.
14. Although that way may not be obvious at first unless you're Dutch.
15. Now is better than never.
16. Although never is often better than right now.[d]
17. If the implementation is hard to explain, it's a bad idea.
18. If the implementation is easy to explain, it may be a good idea.
19. Namespaces are one honking great idea – let's do more of those!