

无效值

- nan,inf,极大值, nonvalue

无效值的出现

- 各无效值间运算的结果
- NAN的传播性/NAN的不可比较性

无效值的处理

- 产品的无效值处理
 - tif
 - nc/hdf
- 运算中的无效值处理

设计模式

设计模式简介

Python中的设计模式

单例模式

所谓单例模式，也就是说不管什么时候我们要确保只有一个对象实例存在。很多情况下，整个系统中只需要存在一个对象，所有的信息都从这个对象获取，比如系统的配置对象，或者是线程池。这些场景下，就非常适合使用单例模式。总结起来，就是说不管我们初始化一个对象多少次，真正干活的对象只会生成一次并且在首次生成。

单例模式应用场景

- Python的logger就是一个单例模式，用以日志记录
- Windows的资源管理器是一个单例模式
- 线程池，数据库连接池等资源池一般也用单例模式
- 网站计数器

单例模式概念及一般实现

- 模块
- 装饰器
- 使用类
- `__new__`

使用单例模式实现日志

