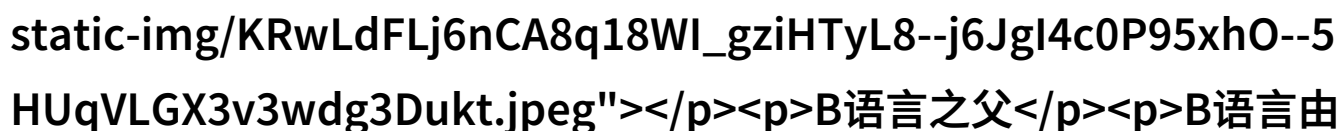
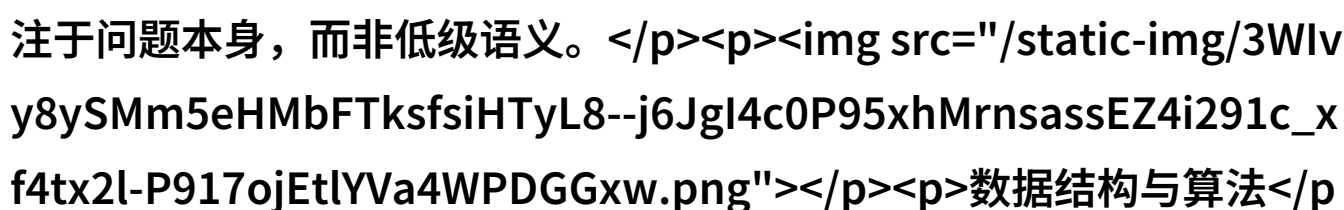


B系列探索从基础到深度的全面回顾

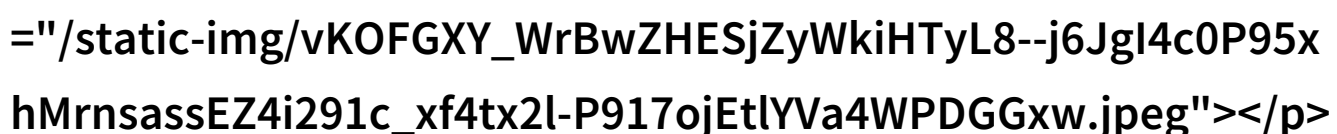
在我们追求知识和理解的道路上，B系列不仅是编程语言中的一个重要组成部分，更是一种思想、方法论和工具集。今天，我们将对30个与B相关的主题进行探讨，揭开它们背后的奥秘。

B语言之父

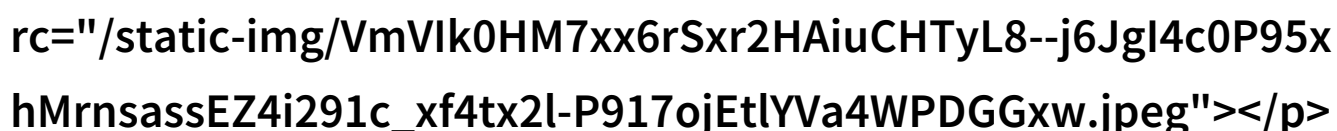
B语言由Donald E. Knuth设计，是一种用于编写算法描述书籍《计算机程序设计艺术》的高级编程语言。它以其简洁明了、表达能力强著称，为后来的C和Pascal等语言奠定了基础。Knuth教授通过一本接一本精心撰写的书籍，将复杂的算法转化为易于理解的小段代码，使得程序员能够专注于问题本身，而非低级语义。

数据结构与算法

B系列中，对数据结构（如栈、队列、树等）的研究，以及各种排序和查找算法（如快速排序、二分搜索）的实现，是B语言特色的体现。这类数据结构和算法对于解决实际问题至关重要，它们使得软件开发者可以更有效地处理大规模数据，并提高系统性能。

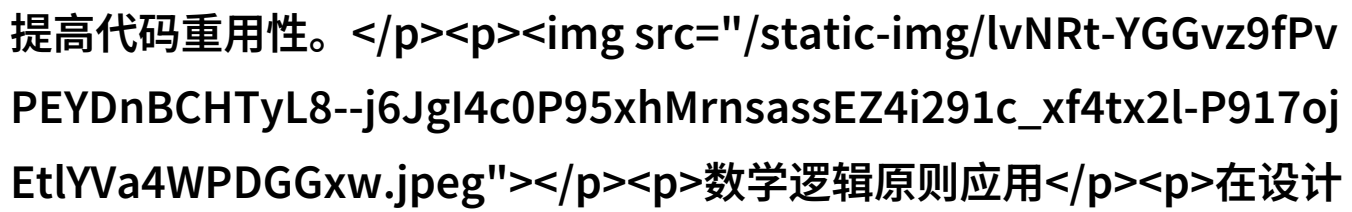
类型系统与内存管理

B语言具有独特的类型系统，这种类型安全性有助于避免运行时错误并促进代码可靠性。在内存管理方面，尽管没有直接支持垃圾收集，但B提供了一套清晰且直观的指针操作规则，使得用户可以自行控制内存使用，从而保证效率。

函数式编程理念

虽然不是传统意义上的函数式编程范式，但B中包含了一些函数式元素，如高阶函数调用以及对递归运用的推崇。这为程序员提供了一种更加抽象化思维方式，有助于解决复杂问题并

提高代码重用性。



数学逻辑原则应用

在设计计算机科学理论模型时，数学逻辑原则是不可或缺的一部分。例如，在图灵机模型中所采用的状态转移规则，即便是在人工智能领域也常见其影响。此外，还包括了形式逻辑在证明理论中的应用，它为验证计算机系统正确性的工作奠定了基础。

教育价值与实践应用

由于其独特性质，许多学术机构选择采用B作为教学工具来教授学生如何思考如何解决问题。而在工业界，其简洁、高效且直观的地方让它成为一些项目或小型团队成员之间沟通技术细节的一个好方法。

[下载本文pdf文件](/pdf/847174-B系列探索从基础到深度的全面回顾.pdf)