**字符串和文本**

对字符串的任何操作均会导致什么？

由于 N 个字符串连接需要分配多少个中间字符串，串行连接也可能成为托管内存压力的主要原因？

如果必须在紧凑循环中或每帧期间对字符串进行连接，请使用什么执行实际连接操作？

参考文档：Microsoft 整理了一份处理 C# 中的字符串的最佳做法清单，msdn.microsoft.com。

目前已知某些 C# String API 的效率极低，其中包括哪些？

示例：使用自己的代码代替String.StartsWith 和 String.EndsWith

尽管正则表达式是匹配和操作字符串的强大方法，但它们可能怎么样？

如果必须使用正则表达式，强烈建议不要使用什么？

尽管反射在开发过程中是绝佳选择（因为它能让解析器快速适应不断变化的数据布局），但众所周知，怎么样？

Unity引入了采用其内置JSONUtility API的部分解决方案，该解决方案提供了读取/发出JSON的Unity序列化系统接口。在大多数基准测试中，它比纯 C# JSON 解析器怎么样？

当遇到文本数据解析所引起的性能问题时，请考虑三种替代解决方案。

避免文本解析成本的最佳方法是完全取消运行时文本解析。通常，这意味着通过某种构建步骤将文本数据“烘焙”成二进制格式。大多数选择使用该方法的开发者会将其数据移动到某种什么衍生的类层级视图中，然后通过AssetBundle分配数据？

选择实现线程的项目通常使用内置的C#什么类（请参阅 msdn.microsoft.com）来管理其工作线程以及标准C#同步类？