### 출처: https://zenn.dev/monry/articles/438f2f78867dec2a59ab

#### 시작하기

C#에 있어서 "문자열 연결 방법"에는 여러 가지가 있다.

- 문자열 연결 연산자 +( String.Concat() 메소드의 신텍스 슈거)
- String.Format() 메소드
- StringBuilder.Append() 메소드
- StringBuilder.AppendFormat() 메소드
- string interpolation (문자열 보간)

그리고, 사람들이 말하는 것은 " + 연결은 성능적으로 좋지 않으므로 StringBuilder.Append()를 사용해라」라고 한다.

실제 + 연결은 위에서 말한 것처럼 컴파일러에 의해 String.Concat()로 변환 되기 때문에 두 String 인스턴스에서 새로운 String인스턴스를 생성하는 코트가 되고 처리 속도적으로도 메모리 적으로도 상당히 불리하기 하다.

# TL; DR

# StringBuilder.Append() 압승!!!

### 상세

벤치마킹 방법

BenchmarkDotNet 를 사용하여 간단한 벤치 마크를 뽑아 보았다.

#### 시나리오는

- 1,000 번째 루프에서 반복 횟수를 문자열로 연결
- 연결시 Prefix, Suffix를 붙인다

실제 코드는 GitHub에 공개 하고 있다.

### 결과

Method	Mean	Error	StdDev	Allocated
String.Format()	971.36 μs	12.731 µs	11.285 µs	14324.43 KB
문자열 보간	405.63 μs	5.367 µs	4.758 μs	4761.95 KB

String.Concat()	365.94 µs	6.989 µs	6.196 µs	4871.04 KB
문자열 연결 연산자	313.47 µs	3.249 µs	2.880 µs	4738.23 KB
StringBuilder.Append() (문자열 보간)	131.18 µs	2.069 µs	1.935 µs	80.97 KB
StringBuilder.AppendFormat()	95.23 µs	1.021 µs	0.955 µs	49.94 KB
StringBuilder.Append()	33.49 µs	0.383 µs	0.340 µs	26.5 KB

무려 StringBuffer.Append()는 String.Format()의 약 30 배 빠르다! 힙 할당에 이르러서는 약 1/550!!!