

# Satrlar bilan ishlash

String (matn) - bu char(belgi) lar ketma-ketligidir. String bu belgilar ketma-ketligidan iborat object hisoblanadi.



```
string str1 = "Hello";  
string str2 = "Personal Development Process";
```

# Satrlarni yaratish



```
// null qiymat berish
string message1 = null;

// bo'sh string hosil qilish literal bilan
string message2 = "";

// bo'sh string hosil qilish "" literalisiz
string message3 = System.String.Empty;

// new kalit so'zi orqali yaratish
string alphabet = new string("ABCDEFGG");
```

# String immutable toifa

String toifasi **immutable** toifa hisoblanadi. Ya'ni ular yaratilgandan keyin o'zgartirib bo'lmaydi. Barcha string toifasidagi qiymatni o'zgartirishga xizmat qiladigan string metodlari va C# operatorlari bajarilganda, hosil bo'lgan qiymat yangi string obyektida qaytariladi.



```
string s1 = "1-dars ";  
string s2 = "string bilan ishlash";  
  
// += operatori qo'llanilgan paytda yana bir string  
// hosil qilinadi va u  
// qayta s1 objectiga o'zlashtiradi  
s1 += s2;  
  
System.Console.WriteLine(s1);  
  
// Output: "1-dars string bilan ishlash"
```

# String intern pool

Xotiradan unumli foydalanish maqsadida, bir xil qiymatga ega bo'lgan constant literals(o'zgarmas literallar)ni heap xotirada saqlashda **intern pool**dan foydalaniladi. Aynan bu imkoniyat har bir unique(yagona) string literali bitta reference(xotira manzili) da saqlanishiga xizmat qiladi.



```
string s1 = "Hello";  
string s2 = "Hello";  
string s3 = new string("Hello");  
  
Console.WriteLine(object.ReferenceEquals(s1, s2));  
Console.WriteLine(object.ReferenceEquals(s1, s3));  
  
// Output:  
// True  
// False
```

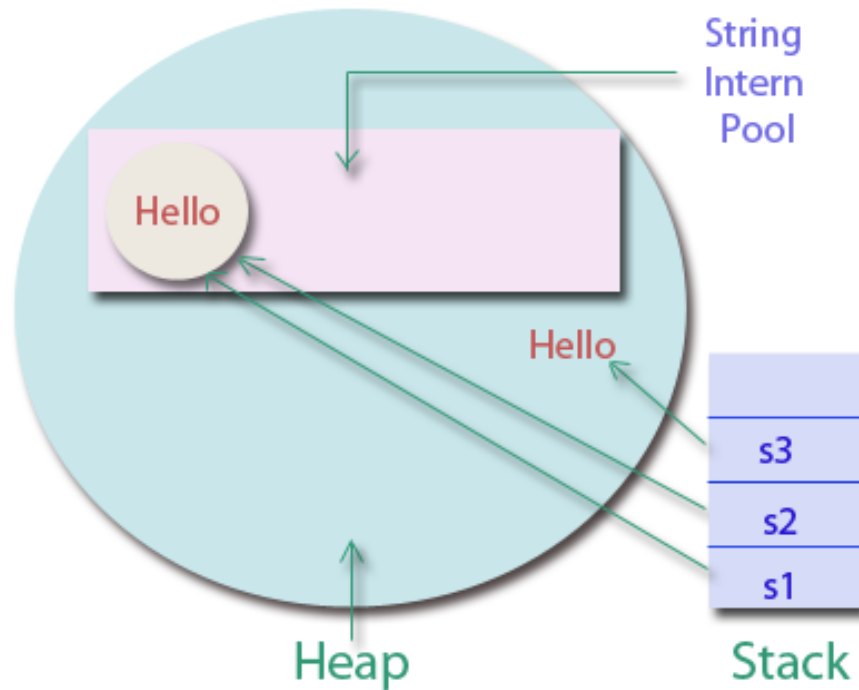


Figure 1



# Verbatim string literali

**Verbatim** string **@** belgisi orqali hosil qilinadi. Aynan ushbe belgi string literali oldidan ishlatilinganda, belgilangan satrda **escape** belgilarini ignore(e'tiborga olmaslik) qilib ketadi.



```
string quote = @"Her name was ""Sara.""";
```

```
//Output: Her name was "Sara."
```

```
string filePath = @"C:\Users\scoleridge\Documents\";
```

```
//Output: C:\Users\scoleridge\Documents\
```

\'	Single quote
\"	Double quote
\\	Backslash
\0	Null
\a	Alert
\b	Backspace
\f	Form feed
\n	New line
\r	Carriage return
\t	Horizontal tab
\v	Vertical tab

# Satrlarni formatlash

String formatlash usullari dastur ishga tushgandan keyin aniqlanadi. `{}`(scope) ichida berilgan har qanday o'zgaruvchidagi qiymat, berilgan satr orasiga qo'shib yuboriladi. Asosan 2 xil usul foydalaniladi: **string interpolation** va **composite** formatlash



```
// string interpolation
string helloWorld = "Salom ";

Console.WriteLine($"{helloWorld}dunyo!");

// composite formatlash
Console.WriteLine("{0}dunyo!", helloWorld);
```

# StringBuilder sinfi

Agarda string toifasi ustida yuzlab, minglab operatsiyalar bajariladigan bo'lsa dasturni ishlashiga biroz ta'sir o'tkazadi. Sababi bitta string toifasi 100 marta o'zgarsa 100 ta yangi string objecti hosil qilinadi. Shu kabi holatlarda **StringBuilder** sinfidan foydalanish maslahat beriladi. StringBuilder toifasidagi qiymat o'zgaradigan bo'lsa, o'zgarish aynan ushbu object ustida amalga oshiriladi.



```
StringBuilder sb = new StringBuilder("Salom");  
sb[0] = 'C';
```

```
Console.WriteLine(sb.ToString());
```