

# Strings

- String is a collection of characters whose memory will allocate on the heap.

String s="abc";



0 1 2

string s1 = @"D:/Sample/ConsoelApp"

**System. String** is a predefined namespace for strings.

- In c#.net strings are two types
  - ▣ **Immutable string**
  - ▣ **Mutable string**
- **Immutable strings:**
  - ▣ Immutable strings are fixed length in size.
  - ▣ Immutable string memory will not change dynamically.
  - ▣ Immutable strings are declared using string data type

# Strings

- ▣ **Methods of System.String class**

- ▣ **copy()**-it copy one string into another

  - String.Copy(value)

- ▣ **concat()**-it concats two strings

  - String.Concat(s1,s2)

- ▣ **compare()**-it compare two strings if strings are equal it returns 0 otherwise it returns +ve or -ve number

**compare(string s1,string s2,bool ignorecase)**

if ignorecase is false it compare strings with case sensitive if it is true compare strings with out case sensitive

- ▣ **Equal(String s)**- it compare two strings and return boolean value

- ▣ **split()**-split string into substrings based on separator

**string []s=s1.split('separator',' separator',....)**

- ▣ **join()**-it will join the array of strings into one string with separator

**string s=string.join('separator', array name)**

# Strings

- ❑ `indexOf("char")`-find the index of given char
- ❑ `indexOf("string")`-find the index of given string
- ❑ `contains("substring")`-it returns Boolean values the substring exist in the string
- ❑ `toLowerCase()`-return string in lowercase
- ❑ `toUpperCase()`-return string in uppercase
- ❑ `ToCharArray()`-it converts string to char array
- ❑ `replace("old string", "new string")`-it replace old string with new string
- ❑ `replace('old char', 'new char')`-it replace old char with new char
- ❑ `Substring(int index)`-it return the substring starting from index to total string.
- ❑ `Substring(int index,int count)`-it return the substring starting from index to given counted chars
- ❑ `String.Format(format, "string")`-it format the string  
ex:`string.format("{0:C}", "1234")`-\$1234

# Strings

- **Formatting strings:** used for formatting the output.
  - {0:F $n$ }-To display fixed decimal points  $n=1,2,\dots$
  - {0:E}-To display output in exponential format
  - {0:C}-To display in currency format-\$123
  - {0:P}-To display in percentage format-123%
  - {0:D}-To display date in long format-March 5 2011
  - {0:d}-To display date in small format-02/5/2011
  - {0:T}-To display time in long format-12:23:12
  - {0:t}-To display time in small format-12:23

# Strings

**Mutable strings:** The strings which are dynamic in length and which automatically increases its length.

- Mutable string are recommended in developing projects
- It increases the performance of the application
- We can declare mutable strings using **StringBuilder** class comes with **System. Text** namespace

# String Builder

- StringBuilder is a predefined class under System. Text namespace
- StringBuilder useful for appending, removing, replacing, or inserting characters.
- **Methods of stringBuilder:**
- **Append()** -Appends string to the end of the current StringBuilder.
- **Insert()**- Inserts a string or object into the specified index of the current StringBuilder.
  - ▣ **Insert(int index, string value)**
  - ▣ **Insert(int index, char value)**
- **Remove()** -Removes a specified number of characters from the current StringBuilder.
  - ▣ **Remove(int startindex,int length)**
- **Replace()**- Replaces a specified character at a specified index.
  - ▣ **Replace('oldchar','newchar')**
  - ▣ **Replace('oldstring','newstring');**