

## ◆ String Methods, Examples & Outputs

Operation	Example	Description	Code Snippet	Output
Length	<code>str.Length</code>	Gets number of characters	<pre>string str = "hello"; Console.WriteLine(str.Length);</pre>	5
Access character	<code>str[0]</code>	Get character at index	<pre>string str = "hello"; Console.WriteLine(str[0]);</pre>	h
ToUpper / ToLower	<code>str.ToUpper()</code>	Convert case	<pre>string str = "Hello"; Console.WriteLine(str.ToUpper()) ;</pre>	HELLO
	<code>str.ToLower()</code>		<pre>Console.WriteLine(str.ToLower()) ;</pre>	hello
Substring	<code>str.Substring(1, 3)</code>	Extract part of string	<pre>string str = "hello"; Console.WriteLine(str.Substring(1 , 3));</pre>	ell
IndexOf / LastIndexOf	<code>str.IndexOf("l")</code>	Index of first match	<pre>string str = "hello"; Console.WriteLine(str.IndexOf('l' ));</pre>	2

	<code>str.LastIndexOf("l")</code>	Index of last match Saved memory full ⓘ	<code>Console.WriteLine(str.LastIndexO f('l'));</code>	3
Contains	<code>str.Contains("ell")</code>	Checks if contains substring	<code>string str = "hello"; Console.WriteLine(str.Contains("e ll"));</code>	True
Replace	<code>str.Replace("l", "x")</code>	Replace characters	<code>string str = "hello"; Console.WriteLine(str.Replace("l" , "x"));</code>	hexxo
Split	<code>str.Split(',')</code>	Split into array	<code>string str = "a,b,c"; string[] parts = str.Split(','); Console.WriteLine(parts[1]);</code>	b
Trim	<code>str.Trim()</code>	Remove whitespace	<code>string str = " hello "; Console.WriteLine(str.Trim());</code>	hello
Startswith / Endswith	<code>str.StartsWith("he")</code>	Check prefix/suffix	<code>string str = "hello"; Console.WriteLine(str.StartsWith( "he"));</code>	True
	<code>str.EndsWith("lo")</code>		<code>Console.WriteLine(str.EndsWith(" lo"));</code>	True
Equals	<code>str.Equals("hello")</code>	Compare equality	<code>string str = "hello"; Console.WriteLine(str.Equals("hel lo"));</code>	True



```
int[] numbers = new int[5];           // declare array of size 5
int[] scores = new int[] { 10, 20, 30 }; // initialized array
string[] names = { "Alice", "Bob" };   // shorthand initialization
```

## ◆ 1.2 Common Methods & Properties of Arrays

Operation	Syntax / Example	Description
Length	<code>arr.Length</code>	Total number of elements
Indexing	<code>arr[0]</code>	Access element at index
Update	<code>arr[2] = 99;</code>	Change value at index
Looping	<code>foreach (int x in arr)</code>	Iterate over array
<code>Array.Sort</code>	<code>Array.Sort(arr)</code>	Sort in ascending order
<code>Array.Reverse</code>	<code>Array.Reverse(arr)</code>	Reverse elements

## ◆ 3.2 Common Methods & Properties of List


Operation	Example	Description	Code Snippet	Output	📄
Add	<code>list.Add(5)</code>	Add element to end	<pre>var list = new List&lt;int&gt;(); list.Add(5); Console.WriteLine(list[0]);</pre>	5	
AddRange	<code>list.AddRange(new[] {1,2})</code>	Add multiple elements	<code>list.AddRange(new[] {1, 2});</code>	[1, 2]	
Insert	<code>list.Insert(1, 100)</code>	Insert at index	<code>list.Insert(1, 100);</code>	[5, 100, 1, 2]	
Remove	<code>list.Remove(100)</code>	Remove first occurrence	<code>list.Remove(100);</code>	[5, 1, 2]	
RemoveAt	<code>list.RemoveAt(0)</code>	Remove at index	<code>list.RemoveAt(0);</code>	[1, 2]	
Contains	<code>list.Contains(2)</code>	Check if list contains element	<code>Console.WriteLine(list.Contains(2));</code>	True	
IndexOf	<code>list.IndexOf(2)</code>	Get index of element	<code>Console.WriteLine(list.IndexOf(2));</code>	1	
Count	<code>list.Count</code>	Get number of elements	<code>Console.WriteLine(list.Count);</code>	2	
Sort	<code>list.Sort()</code>	Sort list ascending	<code>list.Sort();</code>	[1, 2, 5]	
Reverse	<code>list.Reverse()</code>	Reverse the list	<code>list.Reverse();</code>	[5, 2, 1]	
Clear	<code>list.Clear()</code>	Remove all elements	<code>list.Clear();</code>	[] (empty list)	
ToArray	<code>list.ToArray()</code>	Convert to array	<code>int[] arr = list.ToArray();</code>	Same elements as list	

## ◆ 4.2 Common Methods & Properties of Dictionary

Dictionary<string, int> ageMap = new Dictionary<string, int>();

Operation	Example	Description	Code Snippet	Output
Add	<code>dict.Add("Alice", 25)</code>	Add key-value pair	<pre>var dict = new Dictionary&lt;string, int&gt;(); dict.Add("Alice", 25);</pre>	<code>{"Alice": 25}</code>
[ ] indexer	<code>dict["Bob"] = 30</code>	Add/update value by key	<code>dict["Bob"] = 30;</code>	<code>{"Alice":25, "Bob":30}</code>
Remove	<code>dict.Remove("Alice")</code>	Remove key-value by key	<code>dict.Remove("Alice");</code>	<code>{"Bob":30}</code>
ContainsKey	<code>dict.ContainsKey("Alice")</code>	Check if key exists	<pre>Console.WriteLine(dict.ContainsK ey("Alice"));</pre>	<code>False</code>
ContainsValue	<code>dict.ContainsValue(30)</code>	Check if value exists	<pre>Console.WriteLine(dict.ContainsV alue(30));</pre>	<code>True</code>
Count	<code>dict.Count</code>	Number of key-value pairs	<code>Console.WriteLine(dict.Count);</code>	<code>1</code>
Keys	<code>dict.Keys</code>	All keys	<pre>foreach (var key in dict.Keys) Console.WriteLine(key);</pre>	<code>Bob</code>
Values	<code>dict.Values</code>	All values	<pre>foreach (var val in dict.Values) Console.WriteLine(val);</pre>	<code>30</code>
TryGetValue	<code>dict.TryGetValue("Alice", out var age)</code>	Safe get without exception	<pre>bool found = dict.TryGetValue("Alice", out var age); Console.WriteLine(found);</pre>	<code>False</code>
Clear	<code>dict.Clear()</code>	Remove all key-value pairs	<code>dict.Clear();</code>	<code>{}</code> (empty dictionary)

## ◆ 7.2 Common HashSet Methods & Properties of HashSet

Operation	Example	Description	Code Snippet	Output	
Add	<code>set.Add(10)</code>	Adds a value if not present	<pre>set.Add(10); set.Add(10);</pre> <code>HashSet&lt;int&gt; set = new HashSet&lt;int&gt;();</code>	Set: {10} (no duplicates)	
Contains	<code>set.Contains(10)</code>	Checks if value exists	<pre>Console.WriteLine(set.C ontains(10));</pre>	True	
Remove	<code>set.Remove(10)</code>	Removes a value if exists	<code>set.Remove(10);</code>	Removes 10 from set	
Count	<code>set.Count</code>	Number of unique items in set	<pre>Console.WriteLine(set.C ount);</pre>	0 or more	
Clear	<code>set.Clear()</code>	Removes all elements	<code>set.Clear();</code>	Empty set	
SetEquals	<code>set1.SetEquals(set2)</code>	Check if two sets have same values	<pre>Console.WriteLine(set1. SetEquals(set2));</pre>	True or False	
UnionWith	<code>set1.UnionWith(set2)</code>	Merge sets (no duplicates)	<code>set1.UnionWith(set2);</code>	Combined unique set	
IntersectWith	<code>set1.IntersectWith(s2)</code>	Keep only common elements	<code>set1.IntersectWith(set2);</code>	Set with common values	
ExceptWith	<code>set1.ExceptWith(set2)</code>	Remove values that exist in another set	<code>set1.ExceptWith(set2);</code>	Only unique to set1	
ToArray	<code>set.ToArray()</code>	Convert set to array	<pre>var arr = set.ToArray();</pre>	Array from set	