

String Methods:-

Contains(): Checks if a string contains a specific substring.
string.Contains(string): checks if a string contains a substring.
IndexOf(): Returns the index of the first occurrence of a substring.
Substring(): Extracts a portion of a string.
Replace(): Replaces all occurrences of a substring with another string.
ToUpper(): Converts a string to uppercase.
ToLower(): Converts a string to lowercase.
Split(): Splits a string into an array based on a delimiter.
Trim(): Removes leading and trailing whitespaces.
StartsWith(): Checks if a string starts with a specific substring.
EndsWith(): Checks if a string ends with a specific substring.
Join(): Concatenates an array of strings with a delimiter.
Format(): Inserts values into a formatted string.
PadLeft(): Pads a string with a specified character on the left.
PadRight(): Pads a string with a specified character on the right.
Equals(): Compares two strings for equality.
Compare(): Compares two strings lexicographically.
Concat(): Concatenates multiple strings into one.
Insert(): Inserts a string at a specified index.
Remove(): Removes a specified portion of a string.
IsNullOrEmpty(): Checks if a string is null or empty.
IsNullOrWhiteSpace(): Checks if a string is null, empty, or consists only of whitespaces.
CompareOrdinal(): Compares two strings based on their binary value.

Numeric Methods (Math and Convert):-

Math.Abs(): Returns the absolute value of a number.
Math.Round(): Rounds a number to the nearest integer.
Math.Ceiling(): Rounds a number up to the nearest integer.
Math.Floor(): Rounds a number down to the nearest integer.
Math.Max(): Returns the larger of two numbers.
Math.Min(): Returns the smaller of two numbers.
Math.Pow(): Raises a number to a specified power.
Math.Sqrt(): Returns the square root of a number.
Math.Sin(): Returns the sine of an angle.
Math.Cos(): Returns the cosine of an angle.
Math.Tan(): Returns the tangent of an angle.
Convert.ToInt32(): Converts a value to an integer.
Convert.ToDouble(): Converts a value to a double.

Collection Methods (Arrays, Lists, Dictionaries):-

Add(): Adds an item to a collection (e.g., List, Dictionary).

Remove(): Removes an item from a collection.

Sort(): Sorts a collection (array or list).

Insert(): Inserts an item at a specific index.

Clear(): Removes all items from a collection.

Contains(): Checks if a collection contains a specific item.

IndexOf(): Returns the index of an item in a collection.

Find(): Finds the first item matching a condition.

FindAll(): Finds all items matching a condition.

RemoveAt(): Removes an item at a specific index.

Count(): Returns the number of items in a collection.

ToArray(): Converts a collection to an array.

Array-Specific Methods:-

Array.Sort(): Sorts the elements of an array.

Array.Reverse(): Reverses the order of elements in an array.

Array.FindIndex(): Finds the index of the first element that matches a condition.

Array.BinarySearch(): Searches a sorted array for an element using binary search.

Array.Resize(): Changes the size of an array.

Array.ForEach(): Performs an action on each array element.

Array.Copy(): Copies elements from one array to another.

Array.Clone(): Creates a shallow copy of an array.

Array.Contains(): Can be used with arrays by importing System.Linq.

List-Specific Methods:-

AddRange(): Adds a range of items to a list.

InsertRange(): Inserts a range of items at a specified index.

RemoveRange(): Removes a range of items.

Exists(): Checks if any items in the list match a condition.

FindLast(): Finds the last item matching a condition.

FindLastIndex(): Finds the index of the last item matching a condition.

GetRange(): Gets a range of elements from a list.

TrueForAll(): Checks if all items in the list match a condition.

Dictionary-Specific Methods:-

Dictionary.Add(): Adds a key-value pair to a dictionary.

Dictionary.Remove(): Removes a key-value pair by its key.

Dictionary.TryGetValue(): Tries to get the value associated with a key.

ContainsKey(): Checks if the dictionary contains a specific key.

ContainsValue(): Checks if the dictionary contains a specific value.

Keys(): Returns all the keys in the dictionary.

Values(): Returns all the values in the dictionary.

GetEnumerator(): Returns an enumerator to iterate through the dictionary.

LINQ Methods:-

Where(): Filters a collection based on a condition.

Select(): Projects each element into a new form.

First(): Returns the first element in a collection.

FirstOrDefault(): Returns the first element or a default value if none is found.

Any(): Checks if any elements match a condition.

All(): Checks if all elements match a condition.

OrderBy(): Sorts a collection in ascending order.

ThenBy(): Adds a secondary sort order.

GroupBy(): Groups elements by a specified key.

Join(): Joins two collections based on a key.

Count(): Counts the number of elements.

Sum(): Sums up numeric values.

Average(): Returns the average of numeric values.

Max(): Returns the maximum value.

Min(): Returns the minimum value.

Task/Async Methods:-

Task.Run(): Runs a task asynchronously.

Task.Delay(): Delays task execution for a specified amount of time.

Task.WhenAll(): Waits for multiple tasks to complete.

Task.WhenAny(): Waits for any one of several tasks to complete.

await: Waits for an asynchronous task to complete.

async: Marks a method as asynchronous.

File I/O Methods:-

File.ReadAllText(): Reads all text from a file.

File.WriteAllText(): Writes text to a file.

File.ReadAllLines(): Reads all lines from a file into an array.

File.WriteAllLines(): Writes an array of strings to a file.

File.Exists(): Checks if a file exists.

File.Delete(): Deletes a file.

File.AppendAllText(): Appends text to a file.

Exception Handling:-

throw: Throws an exception.

try: Defines a block of code to test for exceptions.

catch: Catches and handles exceptions.

finally: Defines a block of code that will always execute.

throw new Exception(): Creates and throws a new exception.

throw new InvalidOperationException(): Throws an **InvalidOperationException**.

DateTime Methods:-

DateTime.Now: Gets the current local date and time.

DateTime.UtcNow: Gets the current UTC date and time.

DateTime.Parse(): Converts a string to a **DateTime**.

DateTime.TryParse(): Tries to parse a string as a **DateTime**.

DateTime.AddDays(): Adds a specified number of days to a date.

DateTime.AddMonths(): Adds a specified number of months to a date.

DateTime.AddYears(): Adds a specified number of years to a date.

DateTime.ToString(): Converts a date to a string.

Reflection Methods:-

Type.GetType(): Gets a **Type** object based on a type's name.

type.GetMethod(): Retrieves a **MethodInfo** object for a specific method of a type.

type.GetProperty(): Retrieves a **PropertyInfo** object for a specific property of a type.

type.GetField(): Retrieves a **FieldInfo** object for a specific field of a type.

Activator.CreateInstance(): Creates an instance of a type dynamically.

MethodInfo.Invoke(): Invokes a method on an object dynamically.

Serialization Methods:-

JsonConvert.SerializeObject(): Serializes an object to a JSON string (commonly used in .NET with the Newtonsoft.Json library).

JsonConvert.DeserializeObject(): Deserializes a JSON string to an object.

XmlSerializer.Serialize(): Serializes an object to XML.

XmlSerializer.Deserialize(): Deserializes an XML string into an object.

Utility Methods:-

Guid.NewGuid(): Generates a new globally unique identifier (GUID).

Random.Next(): Returns a random integer or a random number within a specified range.

Console.WriteLine(): Writes a line of text to the console.

Console.ReadLine(): Reads a line of input from the console.

Thread.Sleep(): Pauses the current thread for a specified time.

Stopwatch.Start(): Starts a stopwatch to measure elapsed time.

Stopwatch.Stop(): Stops the stopwatch.

Queue Methods:-

Enqueue(): Adds an item to the end of the queue.

Dequeue(): Removes and returns the item at the front of the queue.

Peek(): Returns the item at the front of the queue without removing it.

Clear(): Removes all items from the queue.

Contains(): Checks if the queue contains a specific item.

ToArray(): Copies the elements of the queue to a new array.

Stack Methods:-

Push(): Adds an item to the top of the stack.

Pop(): Removes and returns the item at the top of the stack.

Peek(): Returns the item at the top of the stack without removing it.

Contains(): Checks if the stack contains a specific item.

Clear(): Removes all items from the stack.

ToArray(): Copies the elements of the stack to a new array.

Enumerable Methods (LINQ Extended):-

Take(): Returns a specified number of elements from the start of a collection.

Skip(): Skips a specified number of elements and returns the remaining elements.

TakeWhile(): Returns elements from a collection as long as a specified condition is true.

SkipWhile(): Skips elements as long as a specified condition is true, and then returns the remaining elements.

Distinct(): Returns distinct elements from a collection (removes duplicates).

Union(): Returns the union of two collections (distinct elements).

Intersect(): Returns the common elements between two collections.

Except(): Returns the elements from the first collection that are not in the second collection.

Aggregate(): Applies an accumulator function to the collection (e.g., to compute a sum, product, etc.).

DefaultIfEmpty(): Returns the collection or a default value if the collection is empty.

Zip(): Merges two collections by combining their elements into pairs.

ToList(): Converts an enumerable collection to a List.

ToDictionary(): Converts an enumerable collection to a Dictionary.

FileStream Methods:-

Open(): Opens a file for reading or writing.

Read(): Reads bytes from a file into a buffer.

Write(): Writes bytes from a buffer to a file.

Flush(): Flushes the stream, ensuring that all data is written to the file.

Seek(): Sets the position within the file.

SetLength(): Sets the length of the file.

StreamReader/StreamWriter Methods:-

ReadLine(): Reads a line of characters from the stream.

ReadToEnd(): Reads all characters from the current position to the end of the stream.

Read(): Reads characters or bytes from the stream.

Write(): Writes characters or bytes to the stream.

WriteLine(): Writes a line of characters to the stream.

Flush(): Clears buffers for the stream and causes any buffered data to be written to the underlying device.

Close(): Closes the stream.

Path Methods:-

Combine(): Combines strings into a path.

GetDirectoryName(): Returns the directory information for a specified path.

GetFileName(): Returns the file name and extension of the specified path.

GetExtension(): Returns the extension of the specified path.

GetFileNameWithoutExtension(): Returns the file name without the extension.

GetFullPath(): Returns the full path of a specified file or directory.

GetTempPath(): Returns the path of the temporary directory.

ChangeExtension(): Changes the extension of a file.

Directory Methods:-

CreateDirectory(): Creates a new directory.

Delete(): Deletes a directory.

Exists(): Checks if a directory exists.

GetFiles(): Returns the names of files in a directory.

GetDirectories(): Returns the names of subdirectories.

GetParent(): Returns the parent directory of a specified path.

Move(): Moves a directory to a new location.

Network Methods:-

WebClient.DownloadString(): Downloads a string from a URL.

WebClient.UploadString(): Uploads a string to a URL.

HttpClient.GetAsync(): Sends an asynchronous GET request to a URL.

HttpClient.PostAsync(): Sends an asynchronous POST request to a URL.

HttpRequestMessage(): Represents a request message to be sent by **HttpClient**.

HttpResponseMessage(): Represents a response message received from **HttpClient**.

Socket.Connect(): Establishes a connection to a remote endpoint.

Socket.Send(): Sends data over a connected socket.

Socket.Receive(): Receives data from a connected socket.

TcpListener.Start(): Starts listening for incoming TCP connections.

TcpClient.Connect(): Connects to a remote TCP endpoint.

XmlDocument Methods:-

Load(): Loads an XML document from a file or stream.

Save(): Saves the XML document to a file or stream.

SelectSingleNode(): Selects a single node that matches an XPath query.

SelectNodes(): Selects a list of nodes that match an XPath query.

CreateElement(): Creates a new XML element.

AppendChild(): Appends a child node to the XML document.

RemoveChild(): Removes a child node from the XML document.

Json Methods (Extended):-

JsonObject.Parse(): Parses a JSON string into a JsonObject.

JsonObject.Add(): Adds a new key-value pair to a JsonObject.

JsonObject.Remove(): Removes a key-value pair from a JsonObject.

JArray.Add(): Adds an element to a JArray.

JArray.Remove(): Removes an element from a JArray.

JToken.SelectToken(): Selects a token from a JToken using a JSONPath query.

Event Methods:-

AddHandler(): Adds an event handler to an event.

RemoveHandler(): Removes an event handler from an event.

Invoke(): Invokes the event, calling all registered event handlers.

RaiseEvent(): Raises an event to signal that something has occurred.

Reflection (Extended):-

type.GetConstructor(): Retrieves a constructor for a type.

type.GetEvent(): Retrieves information about an event.

type.GetInterface(): Retrieves information about an interface.

FieldInfo.SetValue(): Sets the value of a field on an object.

FieldInfo.GetValue(): Gets the value of a field from an object.

PropertyInfo.SetValue(): Sets the value of a property on an object.

PropertyInfo.GetValue(): Gets the value of a property from an object.

Regular Expression Methods:-

Regex.Match(): Searches a string for a match to a regular expression.

Regex.Matches(): Returns all matches for a regular expression in a string.

Regex.Replace(): Replaces all occurrences of a regular expression pattern in a string.

Regex.Split(): Splits a string into an array based on a regular expression pattern.

Regex.IsMatch(): Checks if a string matches a regular expression pattern.

Threading Methods:-

Thread.Start(): Starts a thread to execute a specified method.

Thread.Join(): Blocks the calling thread until the thread terminates.

Thread.Sleep(): Suspends the current thread for a specified period.

Thread.Abort(): Attempts to terminate a thread (not recommended in modern .NET versions).

Monitor.Enter(): Acquires an exclusive lock on an object, used in thread synchronization.

Monitor.Exit(): Releases a lock on an object.

Monitor.Wait(): Releases the lock and waits for a notification.

Monitor.Pulse(): Notifies one thread waiting on the lock that it can proceed.

Task.Wait(): Waits for a task to complete.

Process Methods:-

Process.Start(): Starts a new process (e.g., running a program or opening a file).

Process.Kill(): Terminates a process.

Process.WaitForExit(): Waits for the associated process to exit.

Process.GetProcesses(): Retrieves the list of running processes on the system.

Process.GetCurrentProcess(): Gets the process associated with the current program.

Timers:-

Timer.Start(): Starts a timer.

Timer.Stop(): Stops a running timer.

Timer.Dispose(): Releases the resources used by the timer.

Timer.Change(): Changes the start time and interval for a timer.