C# - WINDOWS FILE SYSTEM

https://www.tutorialspoint.com/csharp/csharp_windows_file_system.htm

C# allows you to work with the directories and files using various directory and file related classes such as the **DirectoryInfo** class and the **FileInfo** class.

The DirectoryInfo Class

The **DirectoryInfo** class is derived from the **FileSystemInfo** class. It has various methods for creating, moving, and browsing through directories and subdirectories. This class cannot be inherited.

Following are some commonly used **properties** of the **DirectoryInfo** class –

Sr.No.	Property & Description
1	**Attributes** Gets the attributes for the current file or directory.
2	**CreationTime** Gets the creation time of the current file or directory.
3	**Exists** Gets a Boolean value indicating whether the directory exists.
4	**Extension** Gets the string representing the file extension.
5	**FullName** Gets the full path of the directory or file.
6	**LastAccessTime**

	Gets the time the current file or directory was last accessed.
7	**Name** Gets the name of this DirectoryInfo instance.

Following are some commonly used **methods** of the **DirectoryInfo** class –

Sr.No.	Method & Description
1	**public void Create** Creates a directory.
2	**public DirectoryInfo CreateSubdirectory***stringpath* Creates a subdirectory or subdirectories on the specified path. The specified path can be relative to this instance of the DirectoryInfo class.
3	**public override void Delete** Deletes this DirectoryInfo if it is empty.
4	**public DirectoryInfo[] GetDirectories** Returns the subdirectories of the current directory.
5	**public FileInfo[] GetFiles** Returns a file list from the current directory.

For a complete list of properties and methods, please visit Microsoft's C# documentation.

The FileInfo Class

The **FileInfo** class is derived from the **FileSystemInfo** class. It has properties and instance methods for creating, copying, deleting, moving, and opening of files, and helps in the creation of FileStream objects. This class cannot be inherited.

Following are some commonly used **properties** of the **FileInfo** class –

Sr.No.	Property & Description
1	**Attributes** Gets the attributes for the current file.
2	**CreationTime** Gets the creation time of the current file.
3	**Directory** Gets an instance of the directory which the file belongs to.
4	**Exists** Gets a Boolean value indicating whether the file exists.
5	**Extension** Gets the string representing the file extension.
6	**FullName** Gets the full path of the file.
7	**LastAccessTime** Gets the time the current file was last accessed.
8	**LastWriteTime**

	Gets the time of the last written activity of the file.
9	**Length** Gets the size, in bytes, of the current file.
10	**Name** Gets the name of the file.

Following are some commonly used **methods** of the **FileInfo** class −

Sr.No.	Method & Description
1	**public StreamWriter AppendText** Creates a StreamWriter that appends text to the file represented by this instance of the FileInfo.
2	**public FileStream Create** Creates a file.
3	**public override void Delete** Deletes a file permanently.
4	**public void MoveTo**$stringdestFileName$ Moves a specified file to a new location, providing the option to specify a new file name.
5	**public FileStream Open**$FileModemode$ Opens a file in the specified mode.
6	

	public FileStream Open$FileModemode, FileAccessaccess$ Opens a file in the specified mode with read, write, or read/write access.
7	**public FileStream Open**$FileModemode, FileAccessaccess, FileShareshare$ Opens a file in the specified mode with read, write, or read/write access and the specified sharing option.
8	**public FileStream OpenRead** Creates a read-only FileStream
9	**public FileStream OpenWrite** Creates a write-only FileStream.

For complete list of properties and methods, please visit Microsoft's C# documentation.

Example

The following example demonstrates the use of the above-mentioned classes –

```
using System;
using System.IO;

namespace WindowsFileApplication {
   class Program {
      static void Main(string[] args) {
         //creating a DirectoryInfo object
         DirectoryInfo mydir = new DirectoryInfo(@"c:\Windows");

         // getting the files in the directory, their names and size
         FileInfo [] f = mydir.GetFiles();
         foreach (FileInfo file in f) {
            Console.WriteLine("File Name: {0} Size: {1}", file.Name, file.Length);
         }

         Console.ReadKey();
      }
   }
}
```

When you compile and run the program, it displays the names of files and their respective sizes in the Windows directory.