

# 초보자도 할 수 있는 C# Programming

Fast Campus  
Online

한창민 강사.

# 11강 파일 처리

- 스트림(Stream)
  - 파일, 네트워크 등에서 사용
  - File & Directory 클래스

```
FileStream fs = File.Create(path);
```

```
File.Exists(path)
```

```
File.GetCreationTime(path)
```

```
FileInfo fileInfo = new FileInfo("b.txt");  
FileStream ff = fileInfo.Create();
```

```
ff.Close();  
File.Copy("a.txt", "c.txt");  
FileInfo dst = fileInfo.CopyTo("d.txt");
```

- **System.IO**
  - 파일과 데이터 스트림에 읽고 쓸 수 있게 하는 형식
  - 기본 파일과 디렉터리 지원을 제공하는 형식
  - msdn참조

<https://docs.microsoft.com/ko-kr/dotnet/api/system.io?redirectedfrom=MSDN&view=netframework-4.8>

- 바이트 입출력

- FileStream, BitConverter
- 데이터 형식을 byte 배열로 변환(BitConverter)
- 사용빈도 낮음

```
long lValue= 1234567890123456789;  
Stream outputStream = new FileStream(fileName, FileMode.Create);  
byte[] wBytes = BitConverter.GetBytes(lValue);  
outputStream.Write(wBytes, 0, wBytes.Length);
```

```
FileStream inputStream = new FileStream(fileName, FileMode.Open);  
byte[] rBytes = new byte[sizeof(long)];  
inputStream.Read(rBytes, 0, rBytes.Length);  
long readValue = BitConverter.ToInt64(rBytes, 0);
```

- 텍스트 입출력
  - StreamWriter, StreamReader

```
FileStream fsWrite = new FileStream("a.txt", FileMode.Create);  
StreamWriter sw = new StreamWriter(fsWrite);  
  
sw.Write("Hello World");  
  
sw.Close();
```

```
FileStream fsRead = File.Open("a.txt", FileMode.Open);  
StreamReader sr = new StreamReader(fsRead);  
  
Console.WriteLine("sr.BaseStream.Length:" + sr.BaseStream.Length);  
  
while(false == sr.EndOfStream) {  
    Console.WriteLine(sr.ReadLine());  
}  
  
sr.Close();
```

## • 사용자 자료 입출력

- [Serializable]
- BinaryFormatter, Serialize, Deserialize
- 사용빈도 높음

```
[Serializable]
struct Player
{
    public string _Name;
    public int _Level;
    public double _Exp;
}
```

```
FileStream fsW = new FileStream("savePlayer.txt", FileMode.Create);
BinaryFormatter bf = new BinaryFormatter();
bf.Serialize(fsW, player);
fsW.Close();
```

```
FileStream fsR = new FileStream("savePlayer.txt", FileMode.Open);
BinaryFormatter bf2= new BinaryFormatter();
Player[] readPlayer = (Player[])bf2.Deserialize(fsR);
fsR.Close();
```

➡ 소스코드 (\_135\_File\_BinaryFormatter)

➡ 소스코드 (\_136\_File\_BinaryFormatter02)

## • 이진 입출력

- BinaryWriter, BinaryReader
- 모든 기본 데이터 형식에 읽고 쓰기 오버로딩

```
FileStream fs = new FileStream(fileName, FileMode.Create);  
BinaryWriter bw = new BinaryWriter(fs);
```

```
FileStream fs = new FileStream(fileName, FileMode.Open);  
BinaryReader br = new BinaryReader(fs);
```

```
bw.Write(100);  
bw.Write(100.001f);
```

```
int num = br.ReadInt32();  
float fNum = br.ReadSingle();
```



- CSV데이터 활용
  - 게임 데이터 협업
  - String.Split 활용

	A	B	C	D	E	F
1	stage	min	max	finish	gold	
2	1	10	20	25	100	
3	2	20	40	35	200	
4	3	10	60	100	330	
5	4	10	100	225	460	
6						

```
string str = "0, 1, 2, 3, 4, 5";  
string[] splitRead = str.Split(',');  
for(int i = 0; i < splitRead.Length; i++) {  
    Console.WriteLine(" {0} ", splitRead[i]);  
}
```



```
C:\>  
0 1 2 3 4 5
```