2) 프로그래밍 기본 요소

매 강의 강의자료 시작에 PDF파일을 올려두었어요!

▼ PDF 파일

[수업 목표]

- “Hello World” 구조 확인 해본다.
- Console.WriteLine 메서드를 이용한 출력을 확인한다
- 주석을 작성해 본다.
- 자동 완성 기능으로 효율성 높이며 코드를 작성해본다.

[목차]

01. Hello World 로 기본 코드 구조 확인

02. 출력

03. 주석

04. 자동 완성 기능과 보조 기능을 사용하여 프로그래밍 효율성 높이기

모든 토글을 열고 닫는 단축키
Windows : `Ctrl` + `alt` + `t`
Mac : `⌘` + `⌥` + `t`

---

01. Hello World 로 기본 코드 구조 확인

✔️ 우리의 첫 코드 였던 “Hello World”를 확인해보자

▼ 1) “Hello World” 코드 구조

```
// Hello World.cs
using System;

namespace HelloWorld
{
  class program
  {
    static void Main(string[] args)
    {
      Console.WriteLine("Hello World!");
    }
  }
}
```

- `using System;`은 C#에서 기본적으로 제공하는 네임스페이스(System 네임스페이스)를 사용하기 위한 코드입니다. Console 클래스를 사용하기 위해 필요합니다.
- `namespace`는 코드를 구성하는 데 사용되며 클래스 및 기타 네임스페이스의 컨테이너입니다.
- `class Program`는 C# 클래스를 정의하는 키워드입니다. 클래스 이름은 Program로 지정합니다.
- `static void Main()`은 C#의 진입점(entry point)입니다. Main 메서드는 프로그램이 시작할 때 자동으로 호출되는 메서드입니다. Main 메서드는 프로그램 실행에 필수적입니다.
- `Console.WriteLine("Hello World");`은 콘솔에 출력할 내용을 지정하는 코드입니다. WriteLine 메서드는 새 줄을 시작하고 출력할 문자열을 인자로 받습니다.
- `{}`는 코드 블록의 시작과 끝을 나타내는 중괄호입니다.
- 모든 C# 문은 세미콜론으로 끝납니다 `;`.

02. 출력

✔️ 프로그램 결과를 출력해서 확인해보자

▼ 1) Console.WriteLine

C#에서 콘솔 출력을 할 때는 `Console.WriteLine` 메소드를 사용합니다. `WriteLine` 메소드는 인수로 전달된 값을 출력하고 줄 바꿈(new line) 문자열을 추가합니다. 즉, 출력한 후에는 다음 줄로 커서가 이동합니다.

`Console.WriteLine` 메소드는 다음과 같은 형식으로 사용할 수 있습니다.

```
Console.WriteLine(value);
```

`value`는 출력할 값입니다. `value`는 문자열, 숫자, 변수, 연산식 등 어떤 값이든 사용할 수 있습니다.

▼ [코드스니펫] 출력 기초

```
Console.WriteLine("Hello World!");
Console.WriteLine("My Name is Kero");

Console.WriteLine(10);
Console.WriteLine(3.141592);
Console.WriteLine(3 + 3);
```

다음은 예시 입니다.

```
Console.WriteLine("Hello World!");
Console.WriteLine("My Name is Kero");

[출력]
Hello World!
My Name is Kero
```

```
Console.WriteLine(10);
Console.WriteLine(3.141592);
Console.WriteLine(3 + 3);

[출력]
10
```

```
3.141592
6
```

▼ 2) Console.Write

`Console.Write` 메소드는 `Console.WriteLine` 메소드와 유사하지만, 줄 바꿈 문자열을 추가하지 않습니다. 즉, 출력한 후에는 다음 출력이 이어서 출력됩니다.

예를 들어, 다음과 같이 `Console.Write` 메소드와 `Console.WriteLine` 메소드를 사용하여 값을 출력할 수 있습니다.

▼ [코드스니펫] 한줄 출력 기초

```
Console.Write("Hello! ");
Console.Write("We are Learning ");
Console.WriteLine("at TeamSparta");
```

다음은 예시 입니다.

```
Console.Write("Hello! ");
Console.Write("We are Learning ");
Console.WriteLine("at TeamSparta");

[출력]
Hello! We are Learning at TeamSparta
```

▼ 3) 이스케이프 시퀀스(Escape Sequence)

문자열 내에 특수한 문자를 포함시키기 위해 사용되는 특별한 문자 조합입니다.

다음은 일부 흔한 이스케이프 시퀀스의 예시입니다.

이스케이프 시퀀스	설명
`\'`	작은따옴표(') 삽입
`\"`	큰따옴표(") 삽입
`\\`	역슬래시() 삽입

`\n`	새 줄(줄바꿈) 삽입
`\r`	현재 줄 맨 앞으로 이동
`\t`	탭 삽입
`\b`	백스페이스 삽입

다음은 이스케이프 시퀀스를 사용한 예시입니다.

```
Console.WriteLine("Hello\nWorld");
// 출력결과
// Hello
// World

Console.WriteLine("Name\tAge");
Console.WriteLine("Kero\t30");
Console.WriteLine("Young\t25");
// 출력결과
// Name    Age
// Kero    30
// Young   25

Console.WriteLine("We learn \"C# Programming\"");
// 출력결과
// The book is called "C# Programming"

Console.WriteLine("He said, \'Hello\' to me.");
// 출력결과
// He said, 'Hello' to me.

Console.WriteLine("C:\\MyDocuments\\Project\\");
// 출력결과
// C:\MyDocuments\Project\
```

03. 주석

✔️ 주석을 이용해 코드에 메모를 해두자

▼ 1) 주석 (Comments) 이란?

주석은 코드의 설명이나 개발자간의 의사소통을 위해 사용됩니다.

두 가지 종류의 주석이 있습니다.

- //: 한 줄 주석. 해당 줄 끝까지 주석 처리됩니다.
- /* */: 여러 줄 주석. 시작과 끝을 명시하여 주석 처리됩니다.

주석은 코드를 작성하는 과정에서 중요한 역할을 하며, 코드를 작성하기 전에 설계한 내용을 주석으로 작성하면 나중에 코드를 수정할 때 도움이 됩니다.

예를 들어, 다음과 같이 주석을 사용하여 코드를 설명할 수 있습니다.

```
// 변수 a를 선언합니다.
int a;

/*
여러 줄 주석을 사용하여
다음과 같이 코드를 설명할 수 있습니다.
*/

// 변수 a에 10을 할당합니다.
a = 10;
```

▼ 2) 주의할 점

1. **주석은 코드를 대체하지 않는다**
   주석은 코드를 대신하는 것이 아니라, 코드를 설명하거나 보충하는 역할을 합니다. 주석으로 코드의 복잡한 부분이나 로직을 대체하지 않아야 합니다.
2. **주석의 내용은 정확하고 명확해야 한다**
   주석의 내용은 코드의 작동 방식이나 의도를 명확하게 설명해야 합니다. 주석이 혼란스러우거나 모호하다면 오히려 코드를 이해하기 어려워질 수 있습니다. 주석에 틀린 정보가 포함되지 않도록 주의해야 합니다.
3. **주석은 업데이트 되어야 한다**
   코드가 변경되면 주석도 변경되어야 합니다. 주석이 오래된 버전의 코드를 설명하는 것일 경우, 코드와 맞지 않는 정보를 제공할 수 있습니다. 주석은 코드 변경 사항에 맞춰서 업데이트되어야 합니다.
4. **주석은 필요한 경우에만 사용해야 한다**
   코드가 명확하고 의도가 분명하다면 주석은 필요하지 않을 수도 있습니다. 불필요한 주석은 코드를 복잡하게 만들 수 있습니다. 주석은 코드를 이해하는 데 도움이 되는 경우에만 사용하는 것이 좋습니다.

04. 자동 완성 기능과 보조 기능을 사용하여 프로그래밍 효율성 높이기

익숙해 질수록 편리해지는 기능!

▼ 자동 완성 기능을 사용하여 코드 작성 시간을 단축하는 방법

1. 클래스, 메서드, 변수 등의 이름을 입력할 때 일부를 입력하고, Tab 키를 눌러 나머지를 자동 완성합니다.
    - Console.WriteLine을 작성할 때, Console.까지 입력하고 Tab 키를 누르면 자동으로 WriteLine이 완성됩니다.
    - 메서드나 변수를 입력하는 도중에 Ctrl + Space를 눌러 IntelliSense를 호출하면, 해당 메서드나 변수에 대한 정보와 예제를 볼 수 있습니다.
2. 코드 템플릿을 사용하여 코드를 더 빠르게 작성합니다.
    - 예를 들어, for문을 작성할 때, for 키워드를 입력하고, 두 번 Tab 키를 누르면 for문의 기본적인 코드 템플릿이 자동으로 생성됩니다.

이전 강의

📕 1) C# 소개, 개발 환경 설정

다음 강의

3강 변수와 자료형