

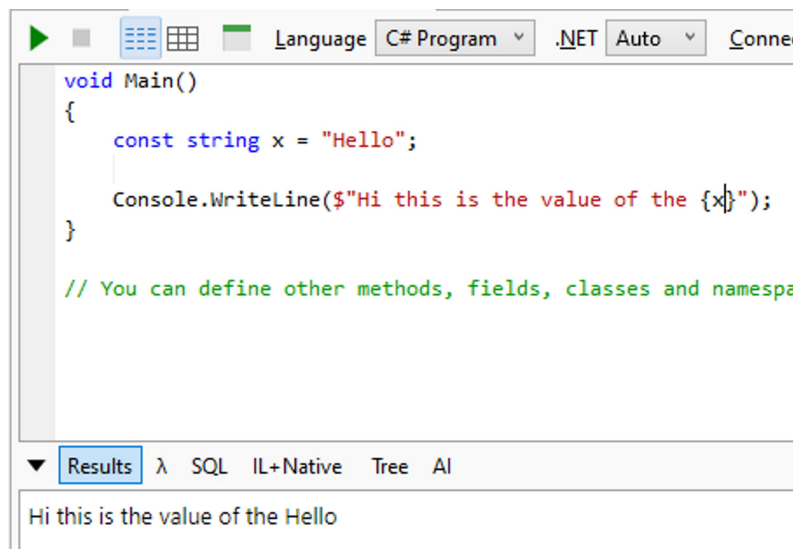
# Readonly vs Const Keywords

Thursday, May 23, 2024 10:07 AM

- ★ So both are constant variables, the main difference is when can we initialize them  
**Const** can be initialized only at compile time where **Readonly** can be initialized both at compile or run time, but for both once initialized can not be changed
- ★ **Readonly** can be initialized inside the Constructor

## Const

- Keyword that creates a constant
- Must have a value at compile time
- Static by default



```
void Main()
{
    const string x = "Hello";
    Console.WriteLine($"Hi this is the value of the {x}");
}

// You can define other methods, fields, classes and namespaces
```

Results | SQL | IL+Native | Tree | AI

Hi this is the value of the Hello

## Readonly

- Ensures that an instance variable or property cannot be modified after initialization
- Cannot be used on local variables
- Can be initialized during compile time or runtime

```

void Main()
{
    const string x = "Hello";
    Console.WriteLine($"Hi this is the value of the {x}");

    InitializeReadOnlyVar obj = new InitializeReadOnlyVar("ROI");
    Console.WriteLine($"Hi this is the value of the {obj.x}");
}

public class InitializeReadOnlyVar{

    public readonly string x;

    public InitializeReadOnlyVar(string value){

        x = value;

    }

}

// You can define other methods, fields, classes and namespaces here

```

▼ Results λ SQL IL+Native Tree AI

Hi this is the value of the Hello  
Hi this is the value of the ROI

## Const

- Variables must be initialized by compile time
- Variables are on a class level

## ReadOnly

- Variables can be initialized at compile time or at runtime
- Variables are on the instance level

## Reference

<https://www.linkedin.com/learning/nail-your-c-sharp-interview/readonly-versus-const?autoSkip=true&resume=false>