

DIFFERENCE BETWEEN CONSTANT AND READONLY AND STATIC

Constant and ReadOnly keyword are used to make a field constant which value cannot be modified. Static keyword is used to make members static that can be shared by all the class objects. In this article, I am going to explain the difference among these three.

Constant

Constant fields or local variables must be assigned a value at the time of declaration and after that they cannot be modified. By default constant are static, hence you cannot define a constant type as static.

```
1. public const int X = 10;
```

A const field is a compile-time constant. A constant field or local variable can be initialized with a constant expression which must be fully evaluated at compile time.

```
1. void Calculate(int Z)
2. {
3.     const int X = 10, X1 = 50;
4.     const int Y = X + X1; //no error, since its evaluated a compile time
5.     const int Y1 = X + Z; //gives error, since its evaluated at run time
6. }
```

You can apply const keyword to built-in value types (byte, short, int, long, char, float, double, decimal, bool), enum, a string literal, or a reference type which can be assigned with a value null.

```
1. const MyClass obj1 = null;//no error, since its evaluated a compile time
2. const MyClass obj2 = new MyClass();//gives error, since its evaluated at run time
```

Constants can be marked as public, private, protected, internal, or protected internal access modifiers. Use the const modifier when you sure that the value a field or local variable would not be changed.

ReadOnly

A readonly field can be initialized either at the time of declaration or with in the constructor of same class. Therefore, readonly fields can be used for run-time constants.

```
1. class MyClass
2. {
3.     readonly int X = 10; // initialized at the time of declaration
4.     readonly int X1;
5.
6.     public MyClass(int x1)
7.     {
8.         X1 = x1; // initialized at run time
9.     }
10. }
```

Explicitly, you can specify a readonly field as static since, like constant by default it is not static. Readonly keyword can be apply to value type and reference type (which initialized by using the new keyword) both. Also, delegate and event could not be readonly.

Use the readonly modifier when you want to make a field constant at run time.

Static

The static keyword is used to specify a static member, which means static members are common to all the objects and they do not tied to a specific object. This keyword can be used with classes, fields, methods, properties, operators, events, and constructors, but it cannot be used with indexers, destructors, or types other than classes.

```
1. class MyClass
2. {
3.     static int X = 10;
4.     int Y = 20;
5.     public static void Show()
6.     {
7.         Console.WriteLine(X);
8.         Console.WriteLine(Y); //error, since you can access only static members
9.     }
10. }
```

Key points about Static keyword

1. If the static keyword is applied to a class, all the members of the class must be static.
2. Static methods can only access static members of same class. Static properties are used to get or set the value of static fields of a class.
3. Static constructor can't be parameterized. Access modifiers can not be applied on Static constructor, it is always a public default constructor which is used to initialize static fields of the class.

Source : <http://www.dotnet-tricks.com/Tutorial/csharp/FU4N141113-Difference-Between-Constant-and-ReadOnly-and-Static.html>