

C# Base

In C#, base keyword is used to access fields, constructors and methods of base class.

You can use base keyword within instance method, constructor or instance property accessor only. You can't use it inside the static method.

C# base keyword: accessing base class field

We can use the base keyword to access the fields of the base class within derived class. It is useful if base and derived classes have the same fields. If derived class doesn't define same field, there is no need to use base keyword. Base class field can be directly accessed by the derived class.

Let's see the simple example of base keyword in C# which accesses the field of base class.

```
1. using System;
2. public class Animal{
3.     public string color = "white";
4. }
5. public class Dog: Animal
6. {
7.     string color = "black";
8.     public void showColor()
9.     {
10.         Console.WriteLine(base.color);
11.         Console.WriteLine(color);
12.     }
13.
14. }
15. public class TestBase
16. {
17.     public static void Main()
18.     {
19.         Dog d = new Dog();
20.         d.showColor();
21.     }
22. }
```

Output:

```
white
black
```

C# base keyword example: calling base class method

By the help of base keyword, we can call the base class method also. It is useful if base and derived classes defines same method. In other words, if method is overridden. If derived class doesn't define same method, there is no need to use base keyword. Base class method can be directly called by the derived class method.

Let's see the simple example of base keyword which calls the method of base class.

```
1. using System;
2. public class Animal{
3.     public virtual void eat(){
4.         Console.WriteLine("eating...");
5.     }
6. }
7. public class Dog: Animal
8. {
9.     public override void eat()
10.    {
11.        base.eat();
12.        Console.WriteLine("eating bread...");
13.    }
14.
15.}
16. public class TestBase
17. {
18.     public static void Main()
19.     {
20.         Dog d = new Dog();
21.         d.eat();
22.     }
23. }
```

Output:

```
eating...
eating bread...
```

C# inheritance: calling base class constructor internally

Whenever you inherit the base class, base class constructor is internally invoked. Let's see the example of calling base constructor.

```
1. using System;
2. public class Animal{
3.     public Animal(){
4.         Console.WriteLine("animal...");
5.     }
6. }
7. public class Dog: Animal
8. {
9.     public Dog()
10.    {
11.        Console.WriteLine("dog...");
```

```
12. }  
13.  
14. }  
15. public class TestOverriding  
16. {  
17.     public static void Main()  
18.     {  
19.         Dog d = new Dog();  
20.  
21.     }  
22. }
```

Output:

```
animal...  
dog...
```