



Properties





# Properties

- Properties are a way of getting values without exposing the member variables of a class.
- Properties have accessors through which the private fields can be read or written.





# Properties

- Properties prevent the methods and attributes of a class from being accessed from outside of the class.
- It is important to prevent people from inserting illegal values or change the value of a field in a way you (the developer) did not expect.
- You, as the developer, decide how your data can be accessed.





# Accessors

- The accessor of a property contains the ability to get (read or compute) or set (write) the property.
- A property can have a get accessor, a set accessor, or both.





# Get Accessor

- Think of the get accessor as similar to that of a method. It will return the property type, just like a method will return the return type.
- The get accessor will return the value of a private field.
- Only using the get accessor is a read-only property, which means a value can't be assigned to it.





# Set Accessor

- The set accessor uses the value keyword to store values in a private variable.
- Only using the set accessor is a write-only property.





# Property Example

```
private int health = 100;  
7 references  
public int Health  
{  
    get  
    {  
        return health;  
    }  
    set  
    {  
        health = value;  
    }  
}
```

```
static void Main(string[] args)  
{  
    Player player = new Player();  
    Console.WriteLine(player.Health);  
    player.Health -= 50;  
    Console.WriteLine(player.Health);  
}
```

```
//Output  
//100  
//50
```

