

استانداردهای کدنویسی:

- کامنت گذاری:

- در تعریف کلاس‌ها، متدها حتماً بشکل زیر توضیحات اضافه شوند، جهت دسترسی کافی است قبل از تعریف متد از /// و کلید tab استفاده شود و سپس موارد مورد نیاز تکمیل شود.

```
/// <summary>
/// Service responsible for Handling Database Connection
/// </summary>
public class DataService
{
    /// <summary>
    /// Check if Database Connection is Valid
    /// </summary>
    public bool IsValid;

    /// <summary>
    /// set Connection String of Database
    /// </summary>
    /// <param name="connectionString"> Connection String should contain
    /// Database server and username and password </param>
    public void SetDatabaseConnection(string connectionString)
    {
    }
}
```

- در کامنت گذاری برای Api ها حتماً، نمونه Sample هم جزو کامنت طبق استاندارد قرار بگیرد.

<https://docs.microsoft.com/en-us/aspnet/core/tutorials/getting-started-with-swashbuckle?view=aspnetcore-5.0&tabs=visual-studio>

- نام گذاری:

- نام متغیرها، متدها، اینترفیس‌ها، کلاس‌ها و ... باید بدرستی انتخاب شود. نام انتخابی باید معنی کاری که قرار است انجام شود را داشته باشد.
- استفاده از اختصار در نام گذاری‌ها تا حد امکان پرهیز شود.

- نام گذاری Class و Struct بشکل PascalCasing باشد.

```
public class DataService
{
```

```
}
```

- نام‌گذاری Interface بشکل PascalCasing با اضافه کردن I در ابتدای نام باشد.

```
public interface IDataService
{
}
```

- نام‌گذاری متدها و property هایی که بشکل public هستند بشکل PascalCasing باشند.

```
public class DataService
{
    public bool IsValid;
    public void GetDatabaseConnection()
    {
    }
}
```

- نام‌گذاری property هایی که بشکل private هستند بشکل camelCasing باشد.
برای استفاده از Interface ها یا Context ابتدای نام _ استفاده شود.

```
public class DataService
{
    private IWorkerQueue _workerQueue;
}
```

- نام‌گذاری پارامترها بشکل camelCasing باشد.

```
public T SomeMethod<T>(int someNumber, bool isValid)
{
}
```

• Layout

۱. در هر خط از کد یک دستور نوشته شود.
۲. در هر خط یک تعریف انجام شود.
۳. بلوک‌های کد که در یک بلوک دیگر تعریف می‌شود از ابتدای بلوک به اندازه یک tab فاصله داشته باشد
۴. برای دستورات If یا For که یک دستور را انجام می‌دهند هم از {} برای مشخص کردن بلوک کد استفاده شود.

• Language Specific

- در استفاده از نوع‌های ابتدایی مثل `int`, `string` از معادل `net.Int`, `net.String` استفاده نشود.
- برای ساخت `string` از متغیرها یا از تابع `format` استفاده شود یا از `$` استفاده گردد.

```
string displayName = $"{nameList[n].LastName}, {nameList[n].FirstName}";
```

```
string displayName = string.Format("{0} , {1}", nameList[n].LastName, nameList[n].FirstName);
```

- استفاده از `var` برای نوع متغیر فقط در صورتی استفاده شود که از سمت راست نوع مشخص باشد یا نوع اهمیت نداشته باشد.
- در حلقه‌های `foreach` بهتر است از `Var` استفاده نشود.

```
foreach (char ch in laugh)
{
    if (ch == 'h')
        Console.Write("H");
    else
        Console.Write(ch);
}
Console.WriteLine();
```

- در دستورات مقایسه‌ای به منظور افزایش کارایی کد، از `&&` و `||` بجای `&` و `|` استفاده شود.
- در هنگام نمونه سازی از یک کلاس بشکل زیر عمل شود.

```
var instance1 = new ExampleClass();
```

```
var instance3 = new ExampleClass
{
    Name = "Desktop",
    ID = 37414,
    Location = "Redmond",
    Age = 2.3 };

```

جهت مطالعه بیشتر به لینک های زیر مراجعه کنید.

<https://docs.microsoft.com/en-us/dotnet/csharp/fundamentals/coding-style/coding-conventions>

<https://github.com/ktaranov/naming-convention/blob/master/C%23%20Coding%20Standards%20and%20Naming%20Conventions.md>