Programming Guidelines

C#

Naming

Symbol	Naming Convention	Example
Class	PascalCase	MotorVehicle
Method	PascalCase	TurnKey()
Event	PascalCase	Click
NameSpace	PascalCase	System.ComponentModel
Property	PascalCase	NumberOfDoors
Private Field	camelCase	listItem
Local Variable	camelCase	arrayOfValues
Parameter	camelCase	inputParameter

- Use US English
- Name Types using nouns
- Do not pluralise Type names
- Do not abbreviate Type names
- Name Methods using verbs
- Name Events using verbs
- Use descriptive names for Parameters
- Do not abbreviate Property names

Layout

- Do not exceed 120 characters per line

- Indent using tabs not spaces

- Add a space between `if` and the first `(` but not between the brackets and the expression: `if (Expression)`

- Curly brackets should be on their own line:

  ```
  if (Expression)
  {

  }
  ```

- Always add curly brackets after: `if`, `else`,`do`,`while`,`for`,`foreach`,`case`

- Add a space around operators: `a + b - c = d`

- Add empty lines between unrelated code blocks

Class Layout

Classes should follow the guide below for layout of the various Members

Order	Member
1	Private Fields
2	Static ReadOnly fields
3	Public properties
4	Constructor
5	Events
6	Methods

Use `region` sections in long code files to group code by intended function using the following guide:

Region name	Contents
Setup	Constructor + Private/Public Fields
Events	Custom + page events
Methods	Business logic
Helpers	Private methods used to perform small tasks
Endpoints	Get, Put, Post, Delete methods returning Http or View Responses

Documentation

- Write documentation in English
- Begin comments with an uppercase character
- End comments with a period
- use a single space between the comment delimiter `//` and the comment text: `// This is a comment.`
- Avoid inline comments
- Document all non-generated public methods with XML comments
- Write comments to explain complex Algorithms or Business Logic

Javascript

Naming

- Javascript files must have the `.js` file extension
- use camelCasing for methods and variables
- global variables should be defined in all capital letters

Layout

- All variables should be declared before they are used
- All variables should be declared at the top of a function
- prefer the `let` keyword over `var` to maintain block level scope
- Do not exceed 120 characters per line
- Indent using tabs not spaces
- All statements should be terminated with a semicolon `;`
- Add a space between `if` and the first `(` but not between the brackets and the expression: `if (Expression)`
- The opening curly bracket for a function should be on the same line as the function declaration:

  ```
  if (Expression){

  }
  ```

Documentation

- Write documentation in English
- Begin comments with an uppercase character
- End comments with a period
- use a single space between the comment delimiter `//` and the comment text: `// This is a comment.`
- Avoid inline comments
- Write comments to explain complex Algorithms or Business Logic