

ASP.NET Web Programming

CODE CONVENTIONS

version - 1.0

MỤC LỤC

GIỚI THIỆU	3
CSS	3
JavaScript	4
Visual Studio .NET (C#)	5
Qui định về việc đặt tên	5
Qui định về việc viết code	8

GIỚI THIỆU

Tài liệu này được sử dụng cho lập trình viên để qui định các qui tắc cần phải thống nhất và tuân thủ.

Nội dung	Biên soạn
CSS	Trần Nguyên Phong
Javascript	Trần Nguyên Phong
VS.NET C#	Trần Nguyên Phong

CSS

Phân loại	Qui định
Name	<p>lower-case</p> <p>Tất cả tên các id, lớp, đối tượng trong css phải được viết dưới dạng chữ thường và cách nhau bằng dấu - (dấu gạch ngang).</p> <p>Tên phải rõ ràng, thể hiện rõ mục đích sử dụng.</p> <p>Ví dụ:</p> <pre>#header-menu, .news-content { font-family: Verdana, Arial; font-size: 1em; }</pre>
Code Indentation	<p>Giữa các khối code nhên cách nhau 4 khoảng trắng (không nên dùng tab)</p> <p>Ví dụ:</p> <pre>.guide-format { color: #000; background-color: rgba(0, 0, 0, .5); border: 1px solid #0f0; }</pre>

CSS3	<p>Khi sử dụng CSS3 lưu ý rằng luôn đặt các thuộc tính của phiên bản mới hơn sau các thuộc tính của phiên bản cũ, các thuộc tính chuẩn được đặt ở cuối. Tham khảo thêm (http://css-tricks.com/ordering-css3-properties/)</p> <p>Ví dụ:</p> <pre> /* Wrong */ .bar { border-radius: 30px 10px; -webkit-border-radius: 30px 10px; } /* Right */ .bar { -webkit-border-radius: 30px 10px; border-radius: 30px 10px; } </pre>
------	---

JavaScript

Phân loại	Qui định
Tên biến, tên hàm	<p>camelCase</p> <p>Ví dụ:</p> <pre> var firstName = "John"; var price = 19.90; function ajaxOptionStudent() { } </pre>
Biến toàn cục/hằng	<p>UPPERCASE</p> <p>Tên tất cả các biến toàn cục hoặc hằng số được đặt theo quy tắc UPPERCASE</p> <p><i>Lưu ý: Nên hạn chế sử dụng biến toàn cục</i></p> <p>Ví dụ:</p> <pre> var ADMINID = 1; var DBPI = PI *2; </pre>
Khai báo biến	<p>Tất cả các biến được khai báo trước khi sử dụng.</p> <p>Nếu sử dụng tên biến bằng tiếng anh không nên đặt tên biến ở dạng số nhiều (thêm s)</p> <p>Ví dụ:</p> <pre> var level, // indentation level value, // instead of values size; // size of table </pre>

Sử dụng khoảng trắng giữa các toán tử	<p>Giữa các toán tử = + / * nên có khoảng trắng</p> <p>Sau dấu , cũng nên có khoảng trắng</p> <p>Ví dụ:</p> <pre>var x = y + z; var value = ["Volvo", "Saab", "Fiat"];</pre>
Toán tử === và !==	<p>Trong các phép toán so sánh nên sử dụng === và !== thay cho == và !=</p> <p>Ví dụ:</p> <pre>if (abc === 1){ return TRUE; }</pre>
Code Indentation	<p>Giữa các khối code nên cách nhau 4 khoảng trắng (không nên dùng tab)</p> <p>Ví dụ:</p> <pre>function abc(){ return 123; }</pre>
Comments	<p>Nên sử dụng // để comment các đoạn mã lệnh</p> <p>Ví dụ:</p> <pre>var i = 0; // Set i to zero. // Lấy danh sách option sinh viên của một lớp function ajaxOptionStudent(idLop) { }</pre>

Visual Studio .NET (C#)

Qui định về việc đặt tên

Phân loại	Qui định
Solution Name	<p>PascalCase</p> <p>Ví dụ:</p> <pre>MySolution</pre>

Project Name	<p>PascalCase</p> <p>Tên project, tên assembly và tên namespace phải đồng nhất</p> <p>Ví dụ:</p> <p>Project: <code>Hitec.StudentManager.csproj</code></p> <p>⇒ Assembly: <code>Hitec.StudentManager.dll</code></p> <p>⇒ Namespace: <code>Hitec.StudentManager</code></p> <p>Đối với một ứng dụng bao gồm nhiều project, tên của các project trong ứng dụng cần có sự thống nhất chung về mặt khuôn dạng và tạo ra sự phân cấp trong tên thông qua dấu chấm (.)</p> <p>Ví dụ:</p> <p><code>Hitec.UniHRM.Web</code></p> <p><code>Hitec.UniHRM.DomainModel</code></p> <p><code>Hitec.UniHRM.DataLayer</code></p> <p><code>Hitec.UniHRM.BussinessLayer</code></p> <p><code>Hitec.UniHRM.Service</code></p> <p>...</p>
Namespace	<p>PascalCase</p> <p>Ví dụ:</p> <p><code>namespace Hitec.StudentManager</code></p>
Class/Struct	<p>PascalCase</p> <p>Ví dụ:</p> <pre>public class DatabaseContext { } public class HomeController : BaseController { }</pre>
Interface	<p>PascalCase</p> <p>Tên của Interface phải bắt đầu bởi ký tự I</p> <p>Ví dụ:</p> <pre>interface IPerson { }</pre>

Function	PascalCase Ví dụ: <pre>public bool AddNewStudent() { } public DataTable Student_List() { }</pre>
Property	PascalCase Ví dụ: <pre>public class Person { public string Name { get; set; } public string PlaceOfBirth { get; set; } }</pre>
Public Variable	Không sử dụng biến public trong lớp, thay thế bằng thuộc tính (Property)
Protected Variable	_camelCase Ví dụ: <pre>protected string _connectionString;</pre>
Private Variable	camelCase Ví dụ: <pre>private int countOfStudents; private string[] availableLanguages;</pre>
Constant	UPPER_CASE Ví dụ: <pre>public const string DENIED_MSG = "Denied!";</pre>
Enum	PascalCase Ví dụ: <pre>enum ActionTypes { AddNew, Update, Delete }</pre>
Delegate	PascalCase Ví Dụ: <pre>public delegate void DataChangedHandler(object o, EventArgs e);</pre>

Event	PascalCase Ví dụ: <code>public event DataChangedHandler DataAdded;</code>
Parameter	camelCase Ví dụ: <code>public void Add(string personId, string name) { }</code>

Qui định về việc viết code

- Sử dụng `//` để comment các đoạn mã lệnh (tránh sử dụng `/* */`)
- Đối với các field, nên sử dụng cách chú thích `///`
Ví dụ:

```
/// <summary>
/// Có cho phép bổ sung dữ liệu hay không?
/// </summary>
private bool _allowAddNew;
```

- Đối với các class, hàm, thuộc tính, public/protected/internal field, phải giải thích rõ chức năng/ý nghĩa, sử dụng chú thích có dạng

```
/// <summary>
///
/// </summary>
```

Ví dụ:

```
/// <summary>
/// Giáo viên
/// </summary>
public class Teacher
{
    /// <summary>
    /// Mã giáo viên
    /// </summary>
    public string TeacherId { get; set; }

    /// <summary>
    /// Họ tên giáo viên
    /// </summary>
    public string Name { get; set; }

    /// <summary>
    /// Lấy danh sách lớp do giáo viên giảng dạy trong năm học
    /// </summary>
    /// <param name="year">
    /// </param>
    /// <returns></returns>
```



```
public List<Course> GetCourses(int year)
{
    ...
}
```

- Nên khai báo các class trong các file riêng biệt. Tên file đặt trùng với tên của class
- Viết chú thích dùng từ khóa dạng Task-List tại các vị trí tạm thời hoặc chưa hoàn chỉnh

Ví dụ:

```
Console.WriteLine(sender.ToString()); //TEMPCODE: ghi tạm để test
//TODO: Get list of students here
```