

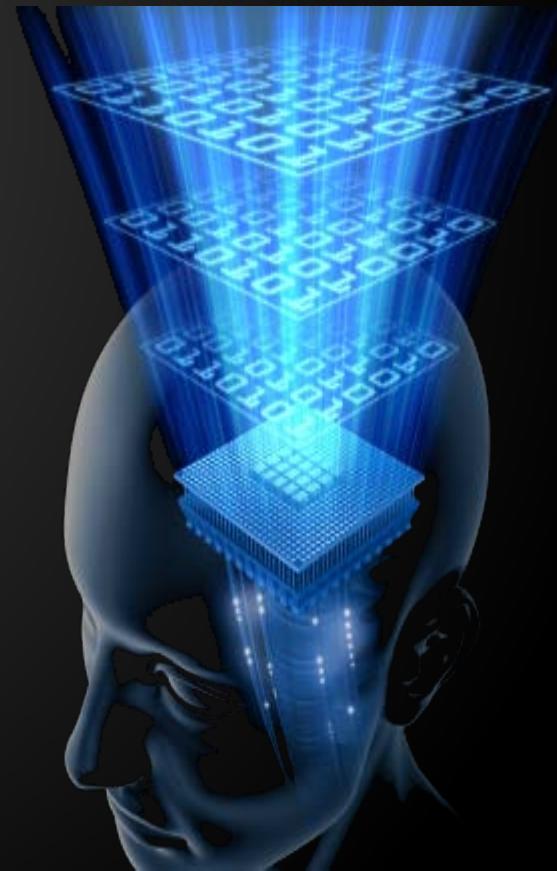


Code Formatting

Correctly Formatting the Source Code

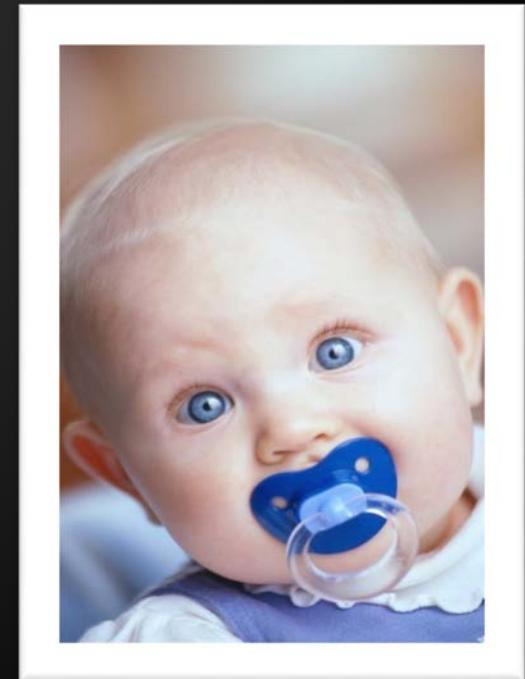


- 1. Why Do We Need Code Formatting?**
- 2. Formatting Methods**
- 3. Formatting Types**
- 4. Common Mistakes**
- 5. Alignments**
- 6. Automated Tools**



Code Formatting

Why Do We Need It?



Why Code Needs Formatting?

```
public const string FILE_NAME  
="example.bin" ; static void Main ( ) {  
FileStream fs= new FileStream(FILE_NAME, FileMode.  
.CreateNew) // Create the writer for data .  
;BinaryWriter w=new BinaryWriter ( fs );//  
Write data to Test.data.  
for( int i=0;i<11;i++){w.Write((int)i);}w .Close();  
fs . Close ( ) // Create the reader for data.  
;fs=new FileStream(FILE_NAME, FileMode.  
. FileAccess.Read) ;BinaryReader r  
= new BinaryReader(fs); // Read data from Test.data.  
for (int i = 0; i < 11; i++){ Console .WriteLine  
(r.ReadInt32 ())}  
;}r . Close ( ); fs . Close ( ) ; }
```

Code Formatting Fundamentals

- ◆ Good formatting goals
 - ◆ To improve code readability
 - ◆ To improve code maintainability
- ◆ Fundamental principle of code formatting:

The formating of the source code should disclose its logical structure.

- ◆ Any formatting style that follows the above principle is good
- ◆ Any other formatting is not good

Formatting Blocks in C#

- ◆ Put { and } alone on a line under the corresponding parent block
- ◆ Indent the block contents by a single [Tab]
 - ◆ Visual Studio will replace the [Tab] with 4 spaces
- ◆ Example:

```
if (some condition)
{
    // Block contents indented by a single [Tab]
    // VS will replace the [Tab] with 4 spaces
}
```

Telerik Academy

Formatting Blocks in JavaScript

- ◆ Put { at the end of the block and } alone on a line under the corresponding parent block
- ◆ Indent the block contents by a single [Tab]
 - ◆ Don't indent with spaces
- ◆ Example:

```
if (some condition) {  
    // Block contents indented by a single [Tab]  
    // Don't use spaces for indentation  
}
```

Empty Lines between Methods

- ◆ Use empty line for separation between methods:

```
public class Factorial
{
    private static ulong CalcFactorial(uint num)
    {
        if (num == 0)
            return 1;
        else
            return num * CalcFactorial(num - 1);
    }

    static void Main()
    {
        ulong factorial = CalcFactorial(5);
        Console.WriteLine(factorial);
    }
}
```

Always use { and } after if
(there is no space to do it here)

Leave empty line
between methods

Methods Indentation

- ◆ Methods should be indented with a single [Tab] from the class body
- ◆ Methods body should be indented with a single [Tab] as well

```
public class IndentationExample
{
    private int zero()
    {
        return 0;
    }
}
```

The entire method is
indented with a single [Tab]

Method body is also indented

Brackets in Methods Declaration

- ◆ Brackets in the method declaration should be formatted as follows:

```
private static ulong CalcFactorial(uint num)
```



- ◆ Don't use spaces between the brackets:

```
private static ulong CalcFactorial ( uint num )
```



```
private static ulong CalcFactorial (uint num)
```

- ◆ The same applies for if-conditions and for-loops:

```
if (condition) { ... }
```

Separating Parameters

- ◆ Separate method parameters by comma followed by a space
 - ◆ Don't put space before the comma
 - ◆ Examples:

```
private void RegisterUser(string username, string password)
```

```
RegisterUser("nakov", "s3cr3t!p@ssw0rd");
```

- ◆ Incorrect examples:

```
private void RegisterUser(string username, string password)
```

```
private void RegisterUser(string username ,string password)
```

```
private void RegisterUser(string username , string password)
```

Empty Lines in Method Body

- ◆ Use an empty line to separate logically related sequences of lines:

```
private List<Report> PrepareReports()
{
    List<Report> reports = new List<Report>();
    // Create incomes reports
    Report incomesSalesReport = PrepareIncomesSalesReport();
    reports.Add(incomesSalesReport);
    Report incomesSupportReport = PrepareIncomesSupportReport();
    reports.Add(incomesSupportReport);
    // Create expenses reports
    Report expensesPayrollReport = PrepareExpensesPayrollReport();
    reports.Add(expensesPayrollReport);
    Report expensesMarketingReport = PrepareExpensesMarketingReport();
    reports.Add(expensesMarketingReport);
    return reports;
}
```

The code snippet illustrates the use of empty lines to separate logically related sequences of lines. Three callout boxes point to specific empty lines in the code:

- A callout box points to the empty line after the declaration of the `reports` list.
- A callout box points to the empty line between the two sections of report creation.
- A callout box points to the empty line before the `return` statement.

- ◆ **Formatting classes / structures / interfaces / enumerations**
 - ◆ **Indent the class body with a single [Tab]**
 - ◆ **Use the following order of definitions:**
 - ◆ **Constants, delegates, inner types, fields, constructors, properties, methods**
 - ◆ **Static members, public members, protected members, internal members, private members**
 - ◆ **The above order of definitions is not the only possible correct one**

Formatting Types – Example in C#

```
public class Dog
{
    // Static variables

    public const string SPECIES =
        "Canis Lupus Familiaris";

    // Instance variables

    private int age;

    // Constructors

    public Dog(string name, int age)
    {
        this.Name = name;
        this.age = age;
    }
}
```

(continues on the next slide)

Formatting Types – Example in C# (2)

```
// Properties  
  
public string Name { get; set; }  
  
// Methods  
  
public void Breath()  
{  
    // TODO: breathing process  
}  
  
public void Bark()  
{  
    Console.WriteLine("wow-wow");  
}  
}
```

Formatting Conditional Statements and Loops

- ◆ **Formatting conditional statements and loops**
 - Always use { } block after if / for / while, even when a single operator follows
 - Indent the block body after if / for / while
 - Always put a new line after a if / for / while block!
 - Always put the { on the next line (in C#)
 - Always put the { on the same line (in JavaScript)
 - Never indent with more than one [Tab]

Conditional Statements and Loops Formatting – C# Examples

◆ Example:

```
for (int i=0; i<10; i++)
{
    Console.WriteLine("i={0}", i);
}
```

The { and }
are missing

◆ Incorrect examples:

```
for (int i=0; i<10; i++)
    Console.WriteLine("i={0}", i);
```

Never put multiple
statements on the
same line!

```
for (int i=0; i<10; i++) Console.WriteLine("i={0}", i);
```

```
for (int i=0; i<10; i++) {
    Console.WriteLine("i={0}", i);
}
```

In C# the { should
be on the next line

Using Empty Lines

- ◆ Empty lines are used to separate logically unrelated parts of the source code

```
public static void PrintList(List<int> ints)
{
    Console.WriteLine("{ ");
    foreach (int item in ints)
    {
        Console.Write(item);
        Console.WriteLine(" ");
    }
    Console.WriteLine("}");
}

static void Main()
{
    // ...
}
```

An empty line
after the foreach block

An empty line
separates the
methods

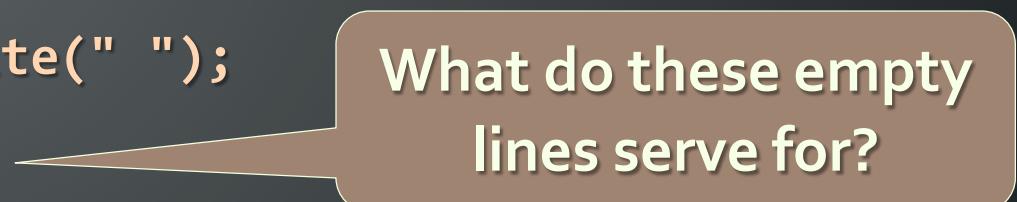
- ◆ Don't put empty lines when not needed!

Misplaced Empty Lines – Example

```
public static void PrintList(List<int> ints)
{
    Console.Write("{ ");
    foreach (int item in ints)
    {
        Console.Write(item);

        Console.WriteLine(" ");
    }
    Console.WriteLine("}");
}
```

static void Main()
{
 // ...
}



What do these empty
lines serve for?

- ◆ Break long lines after punctuation
- ◆ Indent the second line by single [Tab]
- ◆ Do not additionally indent the third line
- ◆ Examples:

```
if (matrix[x, y] == 0 || matrix[x-1, y] == 0 ||
    matrix[x+1, y] == 0 || matrix[x, y-1] == 0 ||
    matrix[x, y+1] == 0)
{ ...
```

```
DictionaryEntry<K, V> newEntry =
    new DictionaryEntry<K, V>(oldEntry.Key,
                                oldEntry.Value);
```

Incorrect Ways To Break Long Lines (in C#)

```
if (matrix[x, y] == 0 || matrix[x-1, y] ==
    0 || matrix[x+1, y] == 0 || matrix[x,
    y-1] == 0 || matrix[x, y+1] == 0)
{ ...
```

```
if (matrix[x, y] == 0 || matrix[x-1, y] == 0 ||
    matrix[x+1, y] == 0 || matrix[x, y-1] == 0 ||
    matrix[x, y+1] == 0)
{ ...
```

```
DictionaryEntry<K, V> newEntry
    = new DictionaryEntry<K, V>(oldEntry
        .Key, oldEntry.Value);
```

Breaking Long Lines in C# and JavaScript

- ◆ In C# use single [Tab] after breaking a long line:

```
if (matrix[x, y] == 0 || matrix[x-1, y] == 0 ||
    → matrix[x+1, y] == 0 || matrix[x, y-1] == 0 ||
    → matrix[x, y+1] == 0)
{
    → matrix[x, y] == 1;
}
```

- ◆ In JavaScript use double [Tab] in the carried long lines:

```
if (matrix[x, y] == 0 || matrix[x-1, y] == 0 ||
    →→ matrix[x+1, y] == 0 || matrix[x, y-1] == 0 ||
    →→ matrix[x, y+1] == 0) {
    → matrix[x, y] == 1;
}
```

- ◆ All types of alignments are considered harmful
 - ◆ Alignments are hard-to-maintain!
- ◆ Incorrect examples:

```
int          count      = 0;  
DateTime    date       = DateTime.Now.Date;  
Student     student    = new Student();  
List<Student> students = new List<Student>();
```

```
matrix[x, y]           = 0;  
matrix[x + 1, y + 1]   = 0;  
matrix[2 * x + y, 2 * y + x] = 0;  
matrix[x * y, x * y]   = 0;
```

Think about
renaming Student
to SchoolStudent

- ◆ Take advantage of your IDE to help formatting the code [Ctrl+K+D]
 - ◆ Automatic alignment
 - ◆ Indentation
- ◆ Style Code analysis
 - ◆ Visual Studio – StyleCop
 - ◆ <http://code.msdn.microsoft.com/sourceanalysis>
 - ◆ Eclipse – CheckStyle
 - ◆ <http://sourceforge.net/projects/eclipse-cs/>
 - ◆ JSHint, JSLint – JavaScript code analysis (all IDEs)
 - ◆ <http://www.jshint.com/>, <http://www.jslint.com/>

Code Formatting

Questions?

1. Format correctly the following source code (given in **Code-Formatting-Homework.zip**):
 - C# code given in the file **events.cs**.
 - JavaScript code given in the file **code.js**.
- Use the official code conventions for C# / JavaScript / Java / PHP:
 - [Official C# Coding Conventions \(MSDN\)](#)
 - [Google JavaScript Style Guide](#)
 - [Official Java Code Conventions \(by Oracle\)](#)
 - [Zend code convention for PHP](#)

Free Trainings @ Telerik Academy

- ◆ C# Programming @ Telerik Academy

- ◆ csharpfundamentals.telerik.com



- ◆ Telerik Software Academy

- ◆ academy.telerik.com

Telerik Academy

- ◆ Telerik Academy @ Facebook

- ◆ facebook.com/TelerikAcademy



- ◆ Telerik Software Academy Forums

- ◆ forums.academy.telerik.com

