



Region

Definizione

- L'utilizzo delle **Region** consente di comprimere o espandere un blocco di codice.
- Utilizzando le **Region**, gli utenti possono dividere il proprio codice in blocchi logici più semplici da comprendere.
- In fase di compilazione le Region vengono ignorate (come i commenti) non hanno alcun impatto sul comportamento del programma durante l'esecuzione.
- **Servono per migliorare la leggibilità e l'organizzazione del codice.**

Sintassi

Di seguito è riportata la sintassi generale della regione in C#

```
#region Region_Name
```

```
    Public static void Main(){  
        //Blocco di codice  
    }
```

```
#endregion
```

XML

VS offre la possibilità di creare la documentazione del codice utilizzando una sintassi speciale per i commenti (uso di tag).

I campi dei commenti includono elementi XML che descrivono il blocco di codice.

```
/// <summary>  
/// This class performs an important function.  
/// </summary>  
public class MyClass {  
}
```

Attivare la generazione del XML

Applicazione

Compilazione

Eventi di compilazione

Debug

Risorse

Servizi

Impostazioni

Percorsi del riferimento

Firma

Sicurezza

Pubblica

Code Analysis

Configurazione: Attiva (Debug) Piattaforma: Attiva (Any CPU)

Generale

Simboli di compilazione condizionale:

☒ Definisci costante DEBUG

☒ Definisci costante TRACE

Piattaforma di destinazione: Any CPU

☒ Preferisci 32 bit

☐ Consenti codice unsafe

☐ Ottimizza codice

Errori e avvisi

Livello di avviso: 4

Non visualizzare avvisi:

Considera gli avvisi come errori

☒ Nessuno

☐ Tutto

☐ Avvisi specifici:

Output

Percorso di output: bin\Debug\ Sfoggia...

☒ File di documentazione XML: bin\Debug\ConsoleApp1.xml

☐ Registra per interoperabilità COM

Genera assembly di serializzazione: Automatico

Avanzate...

Esempio tag

`/// <summary>`

Descrive il metodo

`///<param name=" "></param>`

Descrivere un parametro per un metodo o un costruttore

`/// <returns> </returns>`

Descrivere il valore restituito da un metodo

```
/// <summary>
/// Ottenere il tragitto
/// </summary>
/// <param name="destinazione">destinazione da raggiungere</param>
/// <returns>>Percorso da effettuare</returns>
public double TrovarePercorso(string destinazione)
{
    //Blocco codice
}
```

Lista tag

Tag	Reference	Purpose
<code><c></code>	§D.3.2	Set text in a code-like font
<code><code></code>	§D.3.3	Set one or more lines of source code or program output
<code><example></code>	§D.3.4	Indicate an example
<code><exception></code>	§D.3.5	Identifies the exceptions a method can throw
<code><include></code>	§D.3.6	Includes XML from an external file
<code><list></code>	§D.3.7	Create a list or table
<code><para></code>	§D.3.8	Permit structure to be added to text
<code><param></code>	§D.3.9	Describe a parameter for a method or constructor
<code><paramref></code>	§D.3.10	Identify that a word is a parameter name
<code><permission></code>	§D.3.11	Document the security accessibility of a member
<code><remarks></code>	§D.3.12	Describe additional information about a type
<code><returns></code>	§D.3.13	Describe the return value of a method
<code><see></code>	§D.3.14	Specify a link
<code><seealso></code>	§D.3.15	Generate a <i>See Also</i> entry
<code><summary></code>	§D.3.16	Describe a type or a member of a type
<code><typeparam></code>	§D.3.17	Describe a type parameter for a generic type or method
<code><typeparamref></code>	§D.3.18	Identify that a word is a type parameter name
<code><value></code>	§D.3.19	Describe a property