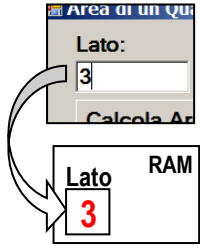
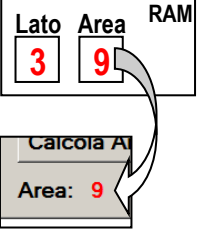


VARIABILI, ASSEGNAZIONE, INPUT/OUTPUT - Linguaggio C#

| | | | |
|--|---|--|---|
| <p style="text-align: center;">Istruzione MessageBox.Show</p> | <p style="text-align: center;">MessageBox.Show ("<sequenza di caratteri>")</p> <p><i>Esempio:</i> MessageBox.Show ("SALUTO TUTTI I MIEI AMICI");</p> | | |
| <p style="text-align: center;">Variabile</p> | <p style="text-align: center;">E' una porzione di memoria RAM utile a contenere un Dato</p> <p style="text-align: center;">Ha un Nome (es. Voto) e contiene un Valore (es. 7) di un certo Tipo (es. intero): il Valore può cambiare durante l'esecuzione</p> | <p style="text-align: center;"><i>Esempio di rappresentazione di una Variabile:</i></p> <div style="text-align: center;"> <p>Voto</p> <table border="1" style="margin: auto;"> <tr> <td style="padding: 5px;">7</td> </tr> </table> <p>int</p> </div> | 7 |
| 7 | | | |
| <p style="text-align: center;">Dichiarazione di una Variabile</p> | <p style="text-align: center;"><tipo> <nome></p> | <p><i>Esempio:</i> double Lato; int Anno;</p> | |
| <p style="text-align: center;">Regola per scrivere un Nome (identificatore)</p> | <p style="text-align: center;">un Nome deve iniziare con una lettera</p> <p style="text-align: center;">... poi è possibile usare liberamente lettere o numeri, non altri simboli.</p> | <p><i>Esempi:</i></p> <p style="text-align: center;">Lato Num1 C1B8</p> <p>2B ... NO! ...questo non è un Nome valido!</p> | |
| <p style="text-align: center;">Tipo di Dato</p> | <p style="text-align: center;">Il Tipo determina quali valori può memorizzare una variabile</p> <p style="text-align: center;"><i>Ecco alcuni Tipi di Dato utilizzabili in C#:</i></p> <p>int numeri interi double numeri decimali</p> | <p><i>Esempi di dichiarazioni di variabili:</i></p> <p>int NumeroAlunni double Ipotenusa</p> | |
| <p style="text-align: center;">Istruzione di Assegnazione</p> | <p style="text-align: center;">Memorizza un Valore in una Variabile</p> <p style="text-align: center;"><variabile> = <valore></p> <p style="text-align: center;">... il Valore può essere anche il valore contenuto in un'altra variabile ...</p> <p style="text-align: center;"><variabile> = <altra variabile></p> <p style="text-align: center;">... oppure, più in generale, il risultato di un calcolo (espressione)</p> <p style="text-align: center;"><variabile> = <espressione></p> <p style="text-align: center;">Il risultato dell'espressione deve essere dello STESSO TIPO (o di un tipo "compatibile") della variabile a cui si assegna</p> | <p><i>Esempi:</i></p> <p>A = 5 ; (assegno un valore) A = B ; (assegno una variabile) A = (X1 + X2) / 2 ; (assegno un'espressione)</p> | |
| <p style="text-align: center;">Accesso alla Proprietà di un Oggetto</p> | <p style="text-align: center;"><oggetto>.<proprietà></p> | <p><i>Esempi:</i> txtDato.Text lblRisultato.BackColor</p> | |
| <p style="text-align: center;">Impostare il Valore di una Proprietà di un Oggetto</p> | <p style="text-align: center;">Le Proprietà sono come le Variabili: è possibile usare l'Istruzione di Assegnazione per impostarne il Valore da codice</p> <p style="text-align: center;"><oggetto>.Text = "<testo>" <oggetto>.BackColor = Color.<colore> <oggetto>.ForeColor = Color.<colore></p> | <p><i>Esempi di Istruzioni di Assegnazione per impostare il valore di una proprietà:</i></p> <p>txtDato.Text = "CIAO A TUTTI!" ; lblRisultato.BackColor = Color.Red ; lblRisultato.ForeColor = Color.White ;</p> | |

| | | |
|--|--|--|
| <p>INPUT DI UN DATO (dalla Form alla RAM)</p> <p>copiare in una Variabile il contenuto di una TextBox</p> | <p>Si usa una Istruzione di Assegnazione per memorizzare nella variabile, il dato che è stato digitato nella TextBox:</p> <p style="text-align: center;"><code><variabile> = <nome-TextBox>.Text</code></p> <p>La proprietà Text è un TESTO mentre la variabile magari è di tipo INTERO. In questi casi è sempre necessaria una "conversione" ...</p> <p style="text-align: center;"><code><variabile> = Convert.ToInt32 (<nome-TextBox>.Text)</code></p> <p>Se la variabile è di tipo DOUBLE si usa la funzione di conversione Convert.ToDouble</p> <p><u>Esempi:</u> <code>Lato = Convert.ToInt32 (txtLato.Text) ;</code> <code>Ipotenusa = Convert.ToDouble (txtIpotenusa.Text) ;</code></p>  | |
| <p>OUTPUT DI UN DATO (dalla RAM alla Form)</p> <p>copiare in una Label il contenuto di una Variabile</p> | <p>Si usa una Istruzione di Assegnazione per memorizzare nella variabile, il dato che è stato digitato nella TextBox:</p> <p style="text-align: center;"><code><nome-Label>.Text = <variabile></code></p> <p>La Variabile, magari, è un INTERO mentre la proprietà TEXT è un TESTO. Anche in questo caso è sempre necessaria una "conversione" ...</p> <p style="text-align: center;"><code><nome-Label>.Text = Convert.ToString(<variabile>)</code></p> <p>E' possibile convertire anche scrivendo <code><variabile>.ToString()</code></p> <p><u>Esempi:</u> <code>lblArea.Text = Convert.ToString(AreaQuadrato) ;</code> <code>lblRisultato.Text = Perimetro.ToString() ;</code></p>  | |
| <p>Uso del ; (punto e virgola)</p> | <p>In C# le istruzioni sono sempre terminate con un punto e virgola (simbolo ;)</p> | <p><u>Esempi:</u> <code>Lato = Convert.toSingle (txtLato.Text) ;</code> <code>Area = Lato * Lato ;</code> <code>lblArea.Text = Convert.toString (Area) ;</code></p> |
| <p>C# è un Linguaggio "case-sensitive"</p> | <p>In C# scrivere in minuscolo o in maiuscolo <u>non è la stessa cosa</u>.</p> <p><u>Esempi:</u> la variabile Lato è diversa dalla variabile lato la proprietà Text esiste, la proprietà text NO!</p> | |
| <p>Commenti nel codice C#</p> | <p>I Commenti sono righe di testo inserite nel codice per spiegare (commentare) il significato del codice stesso. Esse sono ignorate dal compilatore.</p> <p><u>Singola riga di commento:</u> <code>// <testo del commento></code></p> <p><u>Testo (più righe) di commento:</u> <code>/* <testo del commento></code> <code>.....</code> <code>..... */</code></p> <p><u>Esempi:</u> <code>// effettuo l'input dei dati ...</code></p> <p><code>/* questo programma è stato sviluppato il 05/10/2023 dalla 3A Informatica */</code></p> | |