



EBook Gratuito

APPENDIMENTO epplus

Free unaffiliated eBook created from
Stack Overflow contributors.

#epplus

Sommario

Di.....	1
Capitolo 1: Iniziare con epplus.....	2
Osservazioni.....	2
Versioni.....	2
Examples.....	3
Installazione.....	3
Iniziare.....	3
Capitolo 2: Aggiungi dati al documento esistente.....	5
introduzione.....	5
Examples.....	5
Aggiunta di dati.....	5
Capitolo 3: Colonne e righe.....	6
introduzione.....	6
Examples.....	6
Colonne di autofit.....	6
Nascondi colonne e righe.....	6
Ridimensionamento di righe e colonne.....	6
Copia colonne o righe.....	7
Capitolo 4: Convalida dell'input dell'utente.....	8
introduzione.....	8
Examples.....	8
Elenco di convalida.....	8
Convalida intero.....	8
Data / ora di convalida.....	9
Convalida della lunghezza del testo.....	9
Capitolo 5: Creare formule e calcolare intervalli.....	10
introduzione.....	10
Examples.....	10
Aggiungi le formule a una cella.....	10
Formula con più fogli.....	10

Calcolo manuale.....	10
Completa l'esempio con le formule.....	11
Capitolo 6: Creazione di grafici.....	13
introduzione.....	13
Examples.....	13
Grafico a torta.....	13
Grafico a linee.....	13
Capitolo 7: Disegnare il documento Excel.....	15
introduzione.....	15
Examples.....	15
Colore di sfondo.....	15
Stili di confine.....	15
Stili di carattere.....	15
Allineamento del testo e avvolgimento automatico.....	16
Completa l'esempio con tutti gli stili.....	16
Aggiungi un'immagine a un foglio.....	17
Capitolo 8: Importazione di dati dal file esistente.....	18
introduzione.....	18
Examples.....	18
Importa i dati dal file Excel.....	18
Importa i dati dal file CSV.....	18
Importa i dati dal file Excel con FileUpload Control.....	19
Creare un DataTable dal file Excel.....	19
Capitolo 9: Rich Text in cells.....	21
introduzione.....	21
Examples.....	21
Aggiunta di RichText a una cella.....	21
Proprietà di formattazione del testo.....	22
Inserimento di RichText in una cella.....	22
Capitolo 10: Riempiendo il documento con i dati.....	24
introduzione.....	24
Examples.....	24

Riempi con un DataTable.....	24
Riempi con un DataTable da una query SQL o stored procedure.....	24
Riempire manualmente le cellule.....	25
Compila dalla raccolta.....	25
Capitolo 11: Salvataggio del documento Excel.....	27
introduzione.....	27
Examples.....	27
Salva su disco.....	27
Invia al browser.....	28
Salva su disco con SaveFileDialog.....	28
Capitolo 12: Tabella pivot.....	30
introduzione.....	30
Examples.....	30
Creazione di una tabella pivot.....	30
Capitolo 13: tabelle.....	32
introduzione.....	32
Examples.....	32
Aggiunta e formatura di una tabella.....	32
Capitolo 14: Unire le celle.....	33
introduzione.....	33
Examples.....	33
Unione di celle.....	33
Capitolo 15: Valori di formattazione.....	34
introduzione.....	34
Examples.....	34
Formattazione del numero.....	34
Formattazione della data.....	34
Formato testo.....	34
Titoli di coda.....	36

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [epplus](#)

It is an unofficial and free epplus ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official epplus.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capitolo 1: Iniziare con epplus

Osservazioni

EPPlus è una libreria .NET che legge e scrive file di Excel 2007/2010/2013 utilizzando il formato Xml di Open Office (xlsx).

EPPlus supporta:

- Gamme di cellule
- Stile cellulare (bordo, colore, riempimento, carattere, numero, allineamenti)
- Grafici
- Immagini
- forme
- Commenti
- tabelle
- Protezione
- crittografia
- Tabelle pivot
- Convalida dei dati
- Formattazione condizionale
- VBA
- Calcolo della formula

Versioni

Versione	Data di rilascio
Prima uscita	2009-11-30
2.5.0.1	2010-01-25
2.6.0.1	2010-03-23
2.7.0.1	2010-06-17
2.8.0.2	2010-11-15
2.9.0.1	2011-05-31
3.0.0.2	2012-01-31
3.1	2012-04-11
4.0.5	2016/01/08
4.1	2016/07/14

Examples

Installazione

Scarica i file da [CodePlex](#) e aggiungili al progetto.

O installa i file con il Gestore pacchetti.

```
PM> Install-Package EPPlus
```

Iniziare

```
//Create a new ExcelPackage
using (ExcelPackage excelPackage = new ExcelPackage())
{
    //Set some properties of the Excel document
    excelPackage.Workbook.Properties.Author = "VDWWD";
    excelPackage.Workbook.Properties.Title = "Title of Document";
    excelPackage.Workbook.Properties.Subject = "EPPlus demo export data";
    excelPackage.Workbook.Properties.Created = DateTime.Now;

    //Create the WorkSheet
    ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets.Add("Sheet 1");

    //Add some text to cell A1
    worksheet.Cells["A1"].Value = "My first EPPlus spreadsheet!";
    //You could also use [line, column] notation:
    worksheet.Cells[1,2].Value = "This is cell B1!";

    //Save your file
    FileInfo fi = new FileInfo(@"Path\To\Your\File.xlsx");
    excelPackage.SaveAs(fi);
}

//Opening an existing Excel file
FileInfo fi = new FileInfo(@"Path\To\Your\File.xlsx");
using (ExcelPackage excelPackage = new ExcelPackage(fi))
{
    //Get a WorkSheet by index. Note that EPPlus indexes are base 1, not base 0!
    ExcelWorksheet firstWorksheet = excelPackage.Workbook.Worksheets[1];

    //Get a WorkSheet by name. If the worksheet doesn't exist, throw an exeption
    ExcelWorksheet namedWorksheet = excelPackage.Workbook.Worksheets["SomeWorksheet"];

    //If you don't know if a worksheet exists, you could use LINQ,
    //So it doesn't throw an exception, but return null in case it doesn't find it
    ExcelWorksheet anotherWorksheet =
        excelPackage.Workbook.Worksheets.FirstOrDefault(x=>x.Name=="SomeWorksheet");

    //Get the content from cells A1 and B1 as string, in two different notations
    string valA1 = firstWorksheet.Cells["A1"].Value.ToString();
    string valB1 = firstWorksheet.Cells[1,2].Value.ToString();

    //Save your file
    excelPackage.Save();
}
```

Leggi Iniziare con eplus online: <https://riptutorial.com/it/eplus/topic/8070/iniziare-con-eplus>

Capitolo 2: Aggiungi dati al documento esistente

introduzione

Come aggiungere dati a un documento Excel già esistente.

Examples

Aggiunta di dati

```
//the path of the file
string filePath = "C:\\ExcelDemo.xlsx";

//or if you use asp.net, get the relative path
filePath = Server.MapPath("ExcelDemo.xlsx");

//create a fileinfo object of an excel file on the disk
FileInfo file = new FileInfo(filePath);

//create a new Excel package from the file
using (ExcelPackage excelPackage = new ExcelPackage(file))
{
    //create an instance of the the first sheet in the loaded file
    ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets[1];

    //add some data
    worksheet.Cells[4, 1].Value = "Added data in Cell A4";
    worksheet.Cells[4, 2].Value = "Added data in Cell B4";

    //save the changes
    excelPackage.Save();
}
```

Leggi Aggiungi dati al documento esistente online:

<https://riptutorial.com/it/epplus/topic/8596/aggiungi-dati-al-documento-esistente>

Capitolo 3: Colonne e righe

introduzione

Questo argomento contiene informazioni su come lavorare con colonne e righe, come ridimensionare, nascondere, autofit

Examples

Colonne di autofit

```
//Make all text fit the cells
worksheet.Cells[worksheet.Dimension.Address].AutoFitColumns();

//Autofit with minimum size for the column.
double minimumSize = 10;
worksheet.Cells[worksheet.Dimension.Address].AutoFitColumns(minimumSize);

//Autofit with minimum and maximum size for the column.
double maximumSize = 50;
worksheet.Cells[worksheet.Dimension.Address].AutoFitColumns(minimumSize, maximumSize);

//optional use this to make all columns just a bit wider, text would sometimes still overflow
after AutoFitColumns().
for (int col = 1; col <= worksheet.Dimension.End.Column; col++)
{
    worksheet.Column(col).Width = worksheet.Column(col).Width + 1;
}
```

Nascondi colonne e righe

```
//Hide column "A"
worksheet.Column(1).Hidden = true;

//Hide row 1
worksheet.Row(1).Hidden = true;
```

Ridimensionamento di righe e colonne

```
//Set the row "A" height to 15
double rowHeight = 15;
worksheet.Row(1).Height = rowHeight;

//Set the column 1 width to 50
double columnWidth = 50;
worksheet.Column(1).Width = columnWidth;
```

Quando Bestfit è impostato su true, la colonna diventa più ampia quando un utente immette i numeri in una cella

```
worksheet.Column(1).BestFit = true;
```

Copia colonne o righe

```
workSheet.Cells[1, 5, 100, 5].Copy(workSheet.Cells[1, 2, 100, 2]);
```

Copia la colonna 5 nella colonna 2 Fondamentalmente Source.Copy (Destination)

Questo coperebbe solo le prime 100 righe.

```
Cells[RowStart, ColumnStart, RowEnd, ColumnEnd ]  
is the format so to copy a row into another row you would just switch the indexes accordingly
```

Leggi Colonne e righe online: <https://riptutorial.com/it/epplus/topic/8766/colonne-e-righe>

Capitolo 4: Convalida dell'input dell'utente

introduzione

Come convalidare gli input dell'utente. La convalida vincola i valori che un utente può immettere in una cella e / o imposta una casella combinata per l'utente seleziona il valore per la cella. Facoltativamente, un messaggio può essere visualizzato quando l'utente fa clic su una cella e un messaggio di errore, quando la convalida non riesce.

Examples

Elenco di convalida

```
//Add a List validation to B column. Values should be in a list
var val = worksheet.DataValidations.AddListValidation("B:B");
//Shows error message when the input doesn't match the accepted values
val.ShowErrorMessage = true;
//Style of warning. "information" and "warning" allow users to ignore the validation,
//while "stop" and "undefined" doesn't
val.ErrorStyle = OfficeOpenXml.DataValidation.ExcelDataValidationWarningStyle.information;
//Title of the error message box
val.ErrorTitle = "This is the title";
//Message of the error
val.Error = "This is the message";
//Set to true to show a prompt when user clics on the cell
val.ShowInputMessage = true;
//Set the message for the prompt
val.Prompt = "This is a input message";
//Set the title for the prompt
val.PromptTitle = "This is the title from the input message";
//Define the accepted values
val.Formula.Values.Add("This is accepted");
val.Formula.Values.Add("This is also accepted");
val.Formula.Values.Add("Any other thing is rejected");
//Set to true if blank value is accepted
val.AllowBlank = false;

//Add a List validation to the C column
var val2 = worksheet.DataValidations.AddListValidation("C:C");
//Define the Cells with the accepted values
val2.Formula.ExcelFormula = "=$D$3:$D$5";
//Fill the cells with the accepted values
worksheet.Cells["D3"].Value = "Val1";
worksheet.Cells["D4"].Value = "Val2";
worksheet.Cells["D5"].Value = "Val3";
```

Convalida intero

```
//Add a List validation to the C column
var val3 = worksheet.DataValidations.AddIntegerValidation("E:E");
//For Integer Validation, you have to set error message to true
val3.ShowErrorMessage = true;
```

```
val3.Error = "The value must be an integer between 0 and 10";  
//Minimum allowed Value  
val3.Formula.Value = 0;  
//Maximum allowed Value  
val3.Formula2.Value = 10;  
//If the cells are not filled, allow blanks or fill with a valid value,  
//otherwise it could generate a error when saving  
val3.AllowBlank = true;
```

Data / ora di convalida

```
//Add a DateTime Validation to column F  
var val4 = worksheet.DataValidations.AddDateTimeValidation("F:F");  
//For DateTime Validation, you have to set error message to true  
val4.ShowErrorMessage = true;  
//Minimum allowed date  
val4.Formula.Value = new DateTime(2017,03,15, 01, 0,0);  
//Maximum allowed date  
val4.Formula2.Value= new DateTime(2017, 03, 16, 12, 0, 0);  
val4.AllowBlank = true;
```

Convalida della lunghezza del testo

```
//Add a TextLength Validation to column G  
var val5 = worksheet.DataValidations.AddTextLengthValidation("G:G");  
//For TextLenght Validation, you have to set error message to true  
val5.ShowErrorMessage = true;  
//Minimum allowed text lenght  
val5.Formula.Value = 3;  
//Maximum allowed text lenght  
val5.Formula2.Value = 5;  
val5.AllowBlank = true;
```

Leggi Convalida dell'input dell'utente online: <https://riptutorial.com/it/epplus/topic/8739/convalida-dell-input-dell-utente>

Capitolo 5: Creare formule e calcolare intervalli

introduzione

Esempi di base su come creare celle con una formula per i calcoli all'interno del foglio Excel.

Examples

Aggiungi le formule a una cella

```
//set the total value of cells in range A1 - A25 into A27
worksheet.Cells["A27"].Formula = "=SUM(A1:A25)";

//set the number of cells with content in range C1 - C25 into C27
worksheet.Cells["C27"].Formula = "=COUNT(C1:C25)";

//fill column K with the sum of each row, range A - J
for (int i = 1; i <= 25; i++)
{
    var cell = worksheet.Cells[i, 12];
    cell.Formula = "=SUM(" + worksheet.Cells[i, 1].Address + ":" + worksheet.Cells[i,
10].Address + ")";
}

//calculate the quartile of range E1 - E25 into E27
worksheet.Cells[27, 5].Formula = "=QUARTILE(E1:E25,1)";
```

Formula con più fogli

```
//set the total value of all cells in Sheet 2 into G27
worksheet.Cells["G27"].Formula = "=SUM('" + worksheet2.Name + "!' +
worksheet2.Dimension.Start.Address + ":" + worksheet2.Dimension.End.Address + ")";

//set the number of cells with content in Sheet 2, range C1 - C25 into I27
worksheet.Cells["I27"].Formula = "=COUNT('" + excelPackage.Workbook.Worksheets[2].Name + "!' +
+ excelPackage.Workbook.Worksheets[2].Cells["A1:B25"] + ")";
```

Calcolo manuale

Se usi le formule, Excel ti chiederà di salvare il file ogni volta, anche se non sono state apportate modifiche. Per evitare questo comportamento è possibile impostare la modalità di calcolo su manuale.

```
excelPackage.Workbook.CalcMode = ExcelCalcMode.Manual;

//fill the sheet with data and set the formulas
```

```
excelPackage.Workbook.Calculate();
```

Completa l'esempio con le formule

```
//create a new ExcelPackage
using (ExcelPackage excelPackage = new ExcelPackage())
{
    //create 2 WorkSheets
    ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets.Add("Sheet 1");
    ExcelWorksheet worksheet2 = excelPackage.Workbook.Worksheets.Add("Sheet 2");

    //set the calculation mode to manual
    excelPackage.Workbook.CalcMode = ExcelCalcMode.Manual;

    //fill cell data with a loop, note that row and column indexes start at 1
    for (int i = 1; i <= 25; i++)
    {
        for (int j = 1; j <= 10; j++)
        {
            worksheet.Cells[i, j].Value = (i + j) - 1;
            worksheet2.Cells[i, j].Value = (i + j) - 1;
        }
    }

    //set the total value of cells in range A1 - A25 into A27
    worksheet.Cells["A27"].Formula = "=SUM(A1:A25)";

    //set the number of cells with content in range C1 - C25 into C27
    worksheet.Cells["C27"].Formula = "=COUNT(C1:C25)";

    //fill column K with the sum of each row, range A - J
    for (int i = 1; i <= 25; i++)
    {
        var cell = worksheet.Cells[i, 12];
        cell.Formula = "=SUM(" + worksheet.Cells[i, 1].Address + ":" + worksheet.Cells[i,
10].Address + ")";
    }

    //calculate the quartile of range E1 - E25 into E27
    worksheet.Cells[27, 5].Formula = "=QUARTILE(E1:E25,1)";

    //set the total value of all cells in Sheet 2 into G27
    worksheet.Cells["G27"].Formula = "=SUM('" + worksheet2.Name + "!' +
worksheet2.Dimension.Start.Address + ":" + worksheet2.Dimension.End.Address + ")";

    //set the number of cells with content in Sheet 2, range C1 - C25 into I27
    worksheet.Cells["I27"].Formula = "=COUNT('" + excelPackage.Workbook.Worksheets[2].Name +
"!' + excelPackage.Workbook.Worksheets[2].Cells["A1:B25"] + ")";

    //calculate all the values of the formulas in the Excel file
    excelPackage.Workbook.Calculate();

    //Save the file
    FileInfo fi = new FileInfo("FormulaExample.xlsx");
    excelPackage.SaveAs(fi);
}
```

Leggi Creare formule e calcolare intervalli online: <https://riptutorial.com/it/epplus/topic/8227/creare->

Capitolo 6: Creazione di grafici

introduzione

Come creare grafici con EPPlus

Examples

Grafico a torta

```
//create a new ExcelPackage
using (ExcelPackage excelPackage = new ExcelPackage())
{
    //create a WorkSheet
    ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets.Add("Sheet 1");

    //fill cell data with a loop, note that row and column indexes start at 1
    Random rnd = new Random();
    for (int i = 1; i <= 10; i++)
    {
        worksheet.Cells[1, i].Value = "Value " + i;
        worksheet.Cells[2, i].Value = rnd.Next(5, 15);
    }

    //create a new piechart of type Pie3D
    ExcelPieChart pieChart = worksheet.Drawings.AddChart("pieChart", eChartType.Pie3D) as
ExcelPieChart;

    //set the title
    pieChart.Title.Text = "PieChart Example";

    //select the ranges for the pie. First the values, then the header range
    pieChart.Series.Add(ExcelRange.GetAddress(2, 1, 2, 10), ExcelRange.GetAddress(1, 1, 1,
10));

    //position of the legend
    pieChart.Legend.Position = eLegendPosition.Bottom;

    //show the percentages in the pie
    pieChart.DataLabel.ShowPercent = true;

    //size of the chart
    pieChart.SetSize(500, 400);

    //add the chart at cell C5
    pieChart.SetPosition(4, 0, 2, 0);
}
```

Grafico a linee

```
//create a new ExcelPackage
using (ExcelPackage excelPackage = new ExcelPackage())
{
```

```

//create a WorkSheet
ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets.Add("Sheet 1");

//fill cell data with a loop, note that row and column indexes start at 1
Random rnd = new Random();
for (int i = 2; i <= 11; i++)
{
    worksheet.Cells[1, i].Value = "Value " + (i - 1);
    worksheet.Cells[2, i].Value = rnd.Next(5, 25);
    worksheet.Cells[3, i].Value = rnd.Next(5, 25);
}
worksheet.Cells[2, 1].Value = "Age 1";
worksheet.Cells[3, 1].Value = "Age 2";

//create a new piechart of type Line
ExcelLineChart lineChart = worksheet.Drawings.AddChart("lineChart", eChartType.Line) as
ExcelLineChart;

//set the title
lineChart.Title.Text = "LineChart Example";

//create the ranges for the chart
var rangeLabel = worksheet.Cells["B1:K1"];
var range1 = worksheet.Cells["B2:K2"];
var range2 = worksheet.Cells["B3:K3"];

//add the ranges to the chart
lineChart.Series.Add(range1, rangeLabel);
lineChart.Series.Add(range2, rangeLabel);

//set the names of the legend
lineChart.Series[0].Header = worksheet.Cells["A2"].Value.ToString();
lineChart.Series[1].Header = worksheet.Cells["A3"].Value.ToString();

//position of the legend
lineChart.Legend.Position = eLegendPosition.Right;

//size of the chart
lineChart.SetSize(600, 300);

//add the chart at cell B6
lineChart.SetPosition(5, 0, 1, 0);
}

```

Leggi Creazione di grafici online: <https://riptutorial.com/it/epplus/topic/8286/creazione-di-grafici>

Capitolo 7: Disegnare il documento Excel

introduzione

Come applicare lo stile alle celle con tipi di carattere, colore di sfondo, stili di bordo, ecc.

Examples

Colore di sfondo

```
//fill column A with solid red color from hex
worksheet.Column(1).Style.Fill.PatternType = ExcelFillStyle.Solid;
worksheet.Column(1).Style.Fill.BackgroundColor.SetColor(ColorTranslator.FromHtml("#FF0000"));

//fill row 4 with striped orange background
worksheet.Row(4).Style.Fill.PatternType = ExcelFillStyle.DarkHorizontal;
worksheet.Row(4).Style.Fill.BackgroundColor.SetColor(Color.Orange);
```

Stili di confine

```
//make the borders of cell F6 thick
worksheet.Cells[6, 6].Style.Border.Top.Style = ExcelBorderStyle.Thick;
worksheet.Cells[6, 6].Style.Border.Right.Style = ExcelBorderStyle.Thick;
worksheet.Cells[6, 6].Style.Border.Bottom.Style = ExcelBorderStyle.Thick;
worksheet.Cells[6, 6].Style.Border.Left.Style = ExcelBorderStyle.Thick;

//make the borders of cells A18 - J18 double and with a purple color
worksheet.Cells["A18:J18"].Style.Border.Top.Style = ExcelBorderStyle.Double;
worksheet.Cells["A18:J18"].Style.Border.Bottom.Style = ExcelBorderStyle.Double;
worksheet.Cells["A18:J18"].Style.Border.Top.Color.SetColor(Color.Purple);
worksheet.Cells["A18:J18"].Style.Border.Bottom.Color.SetColor(Color.Purple);
```

Stili di carattere

```
//set the font type for cells C1 - C30
worksheet.Cells["C1:C30"].Style.Font.Size = 13;
worksheet.Cells["C1:C30"].Style.Font.Name = "Calibri";
worksheet.Cells["C1:C30"].Style.Font.Bold = true;
worksheet.Cells["C1:C30"].Style.Font.Color.SetColor(Color.Blue);

//Multiple Fonts in the same cell
ExcelRange rg = worksheet.Cells["A1"];
rg.IsRichText = true;
//ExcelRichText uses "using OfficeOpenXml.Style;"
ExcelRichText text1 = rg.RichText.Add("Text with Font1");
text1.Bold = true;
text1.Italic = true;
text1.Color = System.Drawing.Color.Blue;
ExcelRichText text2 = rg.RichText.Add("Text with Font2");
text2.UnderLine = true;
text2.Bold = false;
```

```
text2.Color = System.Drawing.Color.Red;
ExcelRichText text3 = rg.RichText.Add("Text with Font3");
text3.UnderLine = false;
text3.Strike = true;
```

Allineamento del testo e avvolgimento automatico

```
//make column H wider and set the text align to the top and right
worksheet.Column(8).Width = 25;
worksheet.Column(8).Style.HorizontalAlignment = ExcelHorizontalAlignment.Right;
worksheet.Column(8).Style.VerticalAlignment = ExcelVerticalAlignment.Top;

//wrap text in the cells
worksheet.Column(8).Style.WrapText = true;
```

Completa l'esempio con tutti gli stili

```
//create a new ExcelPackage
using (ExcelPackage excelPackage = new ExcelPackage())
{
    //create the WorkSheet
    ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets.Add("Sheet 1");

    //add some dummy data, note that row and column indexes start at 1
    for (int i = 1; i <= 30; i++)
    {
        for (int j = 1; j <= 15; j++)
        {
            worksheet.Cells[i, j].Value = "Row " + i + ", Column " + j;
        }
    }

    //fill column A with solid red color
    worksheet.Column(1).Style.Fill.PatternType = ExcelFillStyle.Solid;

    worksheet.Column(1).Style.Fill.BackgroundColor.SetColor(ColorTranslator.FromHtml("#FF0000"));

    //set the font type for cells C1 - C30
    worksheet.Cells["C1:C30"].Style.Font.Size = 13;
    worksheet.Cells["C1:C30"].Style.Font.Name = "Calibri";
    worksheet.Cells["C1:C30"].Style.Font.Bold = true;
    worksheet.Cells["C1:C30"].Style.Font.Color.SetColor(Color.Blue);

    //fill row 4 with striped orange background
    worksheet.Row(4).Style.Fill.PatternType = ExcelFillStyle.DarkHorizontal;
    worksheet.Row(4).Style.Fill.BackgroundColor.SetColor(Color.Orange);

    //make the borders of cell F6 thick
    worksheet.Cells[6, 6].Style.Border.Top.Style = ExcelBorderStyle.Thick;
    worksheet.Cells[6, 6].Style.Border.Right.Style = ExcelBorderStyle.Thick;
    worksheet.Cells[6, 6].Style.Border.Bottom.Style = ExcelBorderStyle.Thick;
    worksheet.Cells[6, 6].Style.Border.Left.Style = ExcelBorderStyle.Thick;

    //make the borders of cells A18 - J18 double and with a purple color
    worksheet.Cells["A18:J18"].Style.Border.Top.Style = ExcelBorderStyle.Double;
    worksheet.Cells["A18:J18"].Style.Border.Bottom.Style = ExcelBorderStyle.Double;
    worksheet.Cells["A18:J18"].Style.Border.Top.Color.SetColor(Color.Purple);
```

```

worksheet.Cells["A18:J18"].Style.Border.Bottom.Color.SetColor(Color.Purple);

//make all text fit the cells
worksheet.Cells[worksheet.Dimension.Address].AutoFitColumns();

//i use this to make all columns just a bit wider, text would sometimes still overflow
after AutoFitColumns(). Bug?
for (int col = 1; col <= worksheet.Dimension.End.Column; col++)
{
    worksheet.Column(col).Width = worksheet.Column(col).Width + 1;
}

//make column H wider and set the text align to the top and right
worksheet.Column(8).Width = 25;
worksheet.Column(8).Style.HorizontalAlignment = ExcelHorizontalAlignment.Right;
worksheet.Column(8).Style.VerticalAlignment = ExcelVerticalAlignment.Top;

//get the image from disk
using (System.Drawing.Image image =
System.Drawing.Image.FromFile(HttpContext.Current.Server.MapPath("logo.jpg")))
{
    var excelImage = worksheet.Drawings.AddPicture("My Logo", image);

    //add the image to row 20, column E
    excelImage.SetPosition(20, 0, 5, 0);
}
}

```

Aggiungi un'immagine a un foglio

```

//get the image from disk
using (System.Drawing.Image image =
System.Drawing.Image.FromFile(HttpContext.Current.Server.MapPath("logo.jpg")))
{
    var excelImage = worksheet.Drawings.AddPicture("My Logo", image);

    //add the image to row 20, column E
    excelImage.SetPosition(20, 0, 5, 0);
}

```

Leggi Disegnare il documento Excel online: <https://riptutorial.com/it/epplus/topic/8219/disegnare-il-documento-excel>

Capitolo 8: Importazione di dati dal file esistente

introduzione

Come importare dati da un file Excel o CSV esistente.

Examples

Importa i dati dal file Excel

```
//create a list to hold all the values
List<string> excelData = new List<string>();

//read the Excel file as byte array
byte[] bin = File.ReadAllBytes("C:\\ExcelDemo.xlsx");

//or if you use asp.net, get the relative path
byte[] bin = File.ReadAllBytes(Server.MapPath("ExcelDemo.xlsx"));

//create a new Excel package in a memorystream
using (MemoryStream stream = new MemoryStream(bin))
using (ExcelPackage excelPackage = new ExcelPackage(stream))
{
    //loop all worksheets
    foreach (ExcelWorksheet worksheet in excelPackage.Workbook.Worksheets)
    {
        //loop all rows
        for (int i = worksheet.Dimension.Start.Row; i <= worksheet.Dimension.End.Row; i++)
        {
            //loop all columns in a row
            for (int j = worksheet.Dimension.Start.Column; j <=
worksheet.Dimension.End.Column; j++)
            {
                //add the cell data to the List
                if (worksheet.Cells[i, j].Value != null)
                {
                    excelData.Add(worksheet.Cells[i, j].Value.ToString());
                }
            }
        }
    }
}
```

Importa i dati dal file CSV

```
//set the formatting options
ExcelTextFormat format = new ExcelTextFormat();
format.Delimiter = ',';
format.Culture = new CultureInfo(Thread.CurrentThread.CurrentCulture.ToString());
format.Culture.DateTimeFormat.ShortDatePattern = "dd-mm-yyyy";
```

```

format.Encoding = new UTF8Encoding();

//read the CSV file from disk
FileInfo file = new FileInfo("C:\\\\CSVDemo.csv");

//or if you use asp.net, get the relative path
FileInfo file = new FileInfo(Server.MapPath("CSVDemo.csv"));

//create a new Excel package
using (ExcelPackage excelPackage = new ExcelPackage())
{
    //create a WorkSheet
    ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets.Add("Sheet 1");

    //load the CSV data into cell A1
    worksheet.Cells["A1"].LoadFromText(file, format);
}

```

Importa i dati dal file Excel con FileUpload Control

```

//check if there is actually a file being uploaded
if (FileUpload1.HasFile)
{
    //load the uploaded file into the memorystream
    using (MemoryStream stream = new MemoryStream(FileUpload1.FileBytes))
    using (ExcelPackage excelPackage = new ExcelPackage(stream))
    {
        //loop all worksheets
        foreach (ExcelWorksheet worksheet in excelPackage.Workbook.Worksheets)
        {
            //loop all rows
            for (int i = worksheet.Dimension.Start.Row; i <= worksheet.Dimension.End.Row; i++)
            {
                //loop all columns in a row
                for (int j = worksheet.Dimension.Start.Column; j <=
worksheet.Dimension.End.Column; j++)
                {
                    //add the cell data to the List
                    if (worksheet.Cells[i, j].Value != null)
                    {
                        excelData.Add(worksheet.Cells[i, j].Value.ToString());
                    }
                }
            }
        }
    }
}
}

```

Creare un DataTable dal file Excel

```

public static DataTable ExcelPackageToDataTable(ExcelPackage excelPackage)
{
    DataTable dt = new DataTable();
    ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets[1];

    //check if the worksheet is completely empty
    if (worksheet.Dimension == null)

```

```

{
    return dt;
}

//create a list to hold the column names
List<string> columnNames = new List<string>();

//needed to keep track of empty column headers
int currentColumn = 1;

//loop all columns in the sheet and add them to the datatable
foreach (var cell in worksheet.Cells[1, 1, 1, worksheet.Dimension.End.Column])
{
    string columnName = cell.Text.Trim();

    //check if the previous header was empty and add it if it was
    if (cell.Start.Column != currentColumn)
    {
        columnNames.Add("Header_" + currentColumn);
        dt.Columns.Add("Header_" + currentColumn);
        currentColumn++;
    }

    //add the column name to the list to count the duplicates
    columnNames.Add(columnName);

    //count the duplicate column names and make them unique to avoid the exception
    //A column named 'Name' already belongs to this DataTable
    int occurrences = columnNames.Count(x => x.Equals(columnName));
    if (occurrences > 1)
    {
        columnName = columnName + "_" + occurrences;
    }

    //add the column to the datatable
    dt.Columns.Add(columnName);

    currentColumn++;
}

//start adding the contents of the excel file to the datatable
for (int i = 2; i <= worksheet.Dimension.End.Row; i++)
{
    var row = worksheet.Cells[i, 1, i, worksheet.Dimension.End.Column];
    DataRow newRow = dt.NewRow();

    //loop all cells in the row
    foreach (var cell in row)
    {
        newRow[cell.Start.Column - 1] = cell.Text;
    }

    dt.Rows.Add(newRow);
}

return dt;
}

```

Leggi Importazione di dati dal file esistente online:

<https://riptutorial.com/it/epplus/topic/8290/importazione-di-dati-dal-file-esistente>

Capitolo 9: Rich Text in cells

introduzione

La maggior parte delle volte, quando creiamo fogli di calcolo, utilizziamo semplicemente la proprietà Valore di una cella per inserire il contenuto nella cella e la proprietà Style per formattarla.

Occasionalmente, tuttavia, potremmo desiderare di applicare più stili a una cella - magari mettere un titolo grassetto e sottolineato prima del resto del contenuto, o evidenziare una particolare parte del testo in Red - questo è dove entra in gioco la proprietà RichText della cella .

Examples

Aggiunta di RichText a una cella

Ogni elemento di testo su cui si desidera utilizzare una formattazione distinta deve essere aggiunto separatamente, aggiungendo alla proprietà della raccolta RichText della cella.

```
var cell = ws.Cells[1,1];
cell.IsRichText = true; // Cell contains RichText rather than basic values
cell.Style.WrapText = true; // Required to honor new lines

var title = cell.RichText.Add("This is my title");
var text = cell.RichText.Add("\nAnd this is my text");
```

Nota che ogni volta che aggiungi () una nuova stringa, erediterà la formattazione dalla sezione precedente. Pertanto, se si desidera modificare la formattazione predefinita, sarà necessario modificarla solo sulla prima stringa aggiunta.

Questo comportamento può tuttavia causare confusione durante la formattazione del testo. Usando l'esempio sopra, il codice seguente renderà **tutto il testo** nella cella Grassetto e Corsivo - questo non è il comportamento desiderato:

```
// Common Mistake
var title = cell.RichText.Add("This is my title");
title.Bold = true;
title.Italic = true;

var text = cell.RichText.Add("\nAnd this is my text"); // Will be Bold and Italic too
```

L'approccio preferito è aggiungere prima tutte le sezioni di testo, quindi applicare la formattazione specifica della sezione in seguito, come mostrato qui:

```
var title = cell.RichText.Add("This is my title");
title.FontName = "Verdana"; // This will be applied to all subsequent sections as well

var text = cell.RichText.Add("\nAnd this is my text");
```

```
// Format JUST the title
title.Bold = true;
title.Italic = true;
```

Proprietà di formattazione del testo

Esistono numerose proprietà che possono essere applicate alle sezioni di RichText.

```
var title = cell.RichText.Add("This is my title");

// Data Type:      bool
// Default Value:  false
title.Bold = true;

// Data Type:      System.Drawing.Color
// Default Value:  Color.Black
title.Color = Color.Red;
title.Color = Color.FromArgb(255, 0, 0);
title.Color = ColorTranslator.FromHtml("#FF0000");

// Data Type:      string
// Default Value:  "Calibri"
title.FontName = "Verdana";

// Data Type:      bool
// Default Value:  false
title.Italic = true;

// Data Type:      bool
// Default Value:  true
// If this property is set to false, any whitespace (including new lines)
// is trimmed from the start and end of the Text
title.PreserveSpace = true;

// Data Type:      float
// Default Value:  11
// The font size is specified in Points
title.Size = 16;

// Data Type:      bool
// Default Value:  false
// Strikethrough
title.Strike = false;

// Data Type:      string
// Default Value:  Whatever was set when the text was added to the RichText collection
title.Text += " (updated)";

// Data Type:      bool
// Default Value:  false
title.UnderLine = true;

// Data Type:      OfficeOpenXml.Style.ExcelVerticalAlignmentFont
// Default Value:  ExcelVerticalAlignmentFont.None
title.VerticalAlign = ExcelVerticalAlignmentFont.None;
```

Inserimento di RichText in una cella

EPPlus supporta anche la possibilità di inserire testo in una cella usando il metodo `Insert()`. Per esempio:

```
var file = new FileInfo(filePath);
using (var p = new ExcelPackage(file))
{
    var wb = p.Workbook;
    var ws = wb.Worksheets.FirstOrDefault() ?? wb.Worksheets.Add("Sheet1");

    var cell = ws.Cells[1, 1];
    cell.IsRichText = true;
    cell.RichText.Clear(); // Remove any RichText that may be in the cell already
    var s1 = cell.RichText.Add("Section 1.");
    var s2 = cell.RichText.Add("Section 2.");

    var s3 = cell.RichText.Insert(1, "Section 3.");

    s3.Bold = true;
    p.Save();
}
```

Si noti che il metodo `Insert()` NON inserisce in un indice di carattere, ma in un indice di sezione. Poiché le sezioni sono a zero indice, il codice sopra riportato produrrà il seguente testo nella cella:

Sezione 1. Sezione 3. Sezione 2.

Leggi Rich Text in cells online: <https://riptutorial.com/it/epplus/topic/10776/rich-text-in-cells>

Capitolo 10: Riempiendo il documento con i dati

introduzione

Come compilare il foglio Excel creato con dati provenienti da fonti diverse.

Examples

Riempi con un DataTable

```
//create a new ExcelPackage
using (ExcelPackage excelPackage = new ExcelPackage())
{
    //create a datatable
    DataTable dataTable = new DataTable();

    //add three columns to the datatable
    dataTable.Columns.Add("ID", typeof(int));
    dataTable.Columns.Add("Type", typeof(string));
    dataTable.Columns.Add("Name", typeof(string));

    //add some rows
    dataTable.Rows.Add(0, "Country", "Netherlands");
    dataTable.Rows.Add(1, "Country", "Japan");
    dataTable.Rows.Add(2, "Country", "America");
    dataTable.Rows.Add(3, "State", "Gelderland");
    dataTable.Rows.Add(4, "State", "Texas");
    dataTable.Rows.Add(5, "State", "Echizen");
    dataTable.Rows.Add(6, "City", "Amsterdam");
    dataTable.Rows.Add(7, "City", "Tokyo");
    dataTable.Rows.Add(8, "City", "New York");

    //create a WorkSheet
    ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets.Add("Sheet 1");

    //add all the content from the DataTable, starting at cell A1
    worksheet.Cells["A1"].LoadFromDataTable(dataTable, true);
}
```

Riempi con un DataTable da una query SQL o stored procedure

```
//create a new ExcelPackage
using (ExcelPackage excelPackage = new ExcelPackage())
{
    //the query or stored procedure name for the database
    string sqlQuery = "SELECT * FROM myTable";

    //create a datatable
    DataTable dataTable = loadExternalDataSet(sqlQuery);
}
```

```

//create a WorkSheet
ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets.Add("Sheet 1");

//add all the content from the DataTable, starting at cell A1
worksheet.Cells["A1"].LoadFromDataTable(dataTable, true);
}

//method for retrieving data from the database and return it as a datatable
public static DataTable loadExternalDataSet(string sqlQuery)
{
    DataTable dt = new DataTable();

    using (SqlConnection connection = new
SqlConnection(ConfigurationManager.ConnectionStrings["myConnStr"].ConnectionString))
    using (SqlDataAdapter adapter = new SqlDataAdapter(sqlQuery, connection))
    {
        try
        {
            adapter.Fill(dt);
        }
        catch
        {
        }
    }

    return dt;
}

```

Riempire manualmente le cellule

Riempi alcune celle con il testo.

```

worksheet.Cells["A1"].Value = "Lorem ipsum";
worksheet.Cells["B2"].Value = "dolor sit amet";
worksheet.Cells["C3"].Value = "consectetur adipiscing";
worksheet.Cells["D4"].Value = "elit sed do eiusmod";

worksheet.Cells["E5"].Value = 12345;
worksheet.Cells["F6"].Value = DateTime.Now;

```

Compila i dati della cella con un ciclo, nota che gli indici di righe e colonne iniziano da 1

```

for (int i = 1; i <= 30; i++)
{
    for (int j = 1; j <= 15; j++)
    {
        worksheet.Cells[i, j].Value = "Row " + i + ", Column " + j;
    }
}

```

Compila dalla raccolta

```

//create a new ExcelPackage
using (ExcelPackage excelPackage = new ExcelPackage())
{
    //create a WorkSheet

```

```
ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets.Add("Sheet 1");

//create a new list with books
List<Book> books = new List<Book>();

//add some books to the list
for (int i = 0; i < 10; i++)
{
    Book b = new Book();

    b.id = i;
    b.name = "Name " + i;
    b.category = "Category " + i;
    b.date = DateTime.Now.AddDays(i).AddHours(i);

    books.Add(b);
}

//add all the content from the List<Book> collection, starting at cell A1
worksheet.Cells["A1"].LoadFromCollection(books);
}
```

Leggi Riempiendo il documento con i dati online:

<https://riptutorial.com/it/epplus/topic/8223/riempiendo-il-documento-con-i-dati>

Capitolo 11: Salvataggio del documento Excel

introduzione

Esempi su come salvare il foglio Excel creato sul disco o inviarlo al browser.

Examples

Salva su disco

```
//Using File.WriteAllBytes
using (ExcelPackage excelPackage = new ExcelPackage())
{
    //create a new Worksheet
    ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets.Add("Sheet 1");

    //add some text to cell A1
    worksheet.Cells["A1"].Value = "My second EPPlus spreadsheet!";

    //convert the excel package to a byte array
    byte[] bin = excelPackage.GetAsByteArray();

    //the path of the file
    string filePath = "C:\\ExcelDemo.xlsx";

    //or if you use asp.net, get the relative path
    filePath = Server.MapPath("ExcelDemo.xlsx");

    //write the file to the disk
    File.WriteAllBytes(filePath, bin);

    //Instead of converting to bytes, you could also use FileInfo
    FileInfo fi = new FileInfo(filePath);
    excelPackage.SaveAs(fi);
}

//Using SaveAs
using (ExcelPackage excelPackage = new ExcelPackage())
{
    //create a new Worksheet
    ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets.Add("Sheet 1");

    //add some text to cell A1
    worksheet.Cells["A1"].Value = "My second EPPlus spreadsheet!";
    //the path of the file
    string filePath = "C:\\ExcelDemo.xlsx";

    //or if you use asp.net, get the relative path
    filePath = Server.MapPath("ExcelDemo.xlsx");

    //Write the file to the disk
    FileInfo fi = new FileInfo(filePath);
```

```
    excelPackage.SaveAs(fi);  
}
```

Invia al browser

```
//create a new ExcelPackage  
using (ExcelPackage excelPackage = new ExcelPackage())  
{  
    //create the WorkSheet  
    ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets.Add("Sheet 1");  
  
    //add some text to cell A1  
    worksheet.Cells["A1"].Value = "My second EPPlus spreadsheet!";  
  
    //convert the excel package to a byte array  
    byte[] bin = excelPackage.GetAsByteArray();  
  
    //clear the buffer stream  
    Response.ClearHeaders();  
    Response.Clear();  
    Response.Buffer = true;  
  
    //set the correct contenttype  
    Response.ContentType = "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet";  
  
    //set the correct length of the data being send  
    Response.AddHeader("content-length", bin.Length.ToString());  
  
    //set the filename for the excel package  
    Response.AddHeader("content-disposition", "attachment; filename=\"ExcelDemo.xlsx\"");  
  
    //send the byte array to the browser  
    Response.OutputStream.Write(bin, 0, bin.Length);  
  
    //cleanup  
    Response.Flush();  
    HttpContext.Current.ApplicationInstance.CompleteRequest();  
}
```

Salva su disco con SaveFileDialog

```
//Using File.WriteAllBytes  
using (ExcelPackage excelPackage = new ExcelPackage())  
{  
    //create a new Worksheet  
    ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets.Add("Sheet 1");  
  
    //add some text to cell A1  
    worksheet.Cells["A1"].Value = "My fourth EPPlus spreadsheet!";  
  
    //convert the excel package to a byte array  
    byte[] bin = excelPackage.GetAsByteArray();  
  
    //create a SaveFileDialog instance with some properties  
    SaveFileDialog saveFileDialog1 = new SaveFileDialog();  
    saveFileDialog1.Title = "Save Excel sheet";  
}
```



```

saveFileDialog1.Filter = "Excel files|*.xlsx|All files|*.*";
saveFileDialog1.FileName = "ExcelSheet_" + DateTime.Now.ToString("dd-MM-yyyy") + ".xlsx";

//check if user clicked the save button
if (saveFileDialog1.ShowDialog() == DialogResult.OK)
{
    //write the file to the disk
    File.WriteAllBytes(saveFileDialog1.FileName, bin);
}
}

//Using SaveAs
using (ExcelPackage excelPackage = new ExcelPackage())
{
    //create a new Worksheet
    ExcelWorksheet worksheet = excelPackage.Workbook.Worksheets.Add("Sheet 1");

    //add some text to cell A1
    worksheet.Cells["A1"].Value = "My fourth EPPlus spreadsheet!";

    //create a SaveFileDialog instance with some properties
    SaveFileDialog saveFileDialog1 = new SaveFileDialog();
    saveFileDialog1.Title = "Save Excel sheet";
    saveFileDialog1.Filter = "Excel files|*.xlsx|All files|*.*";
    saveFileDialog1.FileName = "ExcelSheet_" + DateTime.Now.ToString("dd-MM-yyyy") + ".xlsx";

    //check if user clicked the save button
    if (saveFileDialog1.ShowDialog() == DialogResult.OK)
    {
        //Get the FileInfo
        FileInfo fi = new FileInfo(saveFileDialog1.FileName);
        //write the file to the disk
        excelPackage.SaveAs(fi);
    }
}
}

```

Leggi Salvataggio del documento Excel online:

<https://riptutorial.com/it/epplus/topic/8202/salvataggio-del-documento-excel>

Capitolo 12: Tabella pivot

introduzione

La tabella pivot è un tipo di tabella interattiva, che può essere utilizzata per calcolare i dati, come ottenere dati di somma o conteggio. Inoltre, gli utenti possono modificare il layout della tabella pivot per analizzare i dati in modi diversi o riassegnare l'etichetta riga / colonna. Ogni volta che gli utenti cambiano layout, i dati verranno ricalcolati nella tabella pivot.

Examples

Creazione di una tabella pivot

```
//create a new ExcelPackage
using (ExcelPackage excelPackage = new ExcelPackage())
{
    //create 2 WorkSheets. One for the source data and one for the Pivot table
    ExcelWorksheet worksheetPivot = excelPackage.Workbook.Worksheets.Add("Pivot");
    ExcelWorksheet worksheetData = excelPackage.Workbook.Worksheets.Add("Data");

    //add some source data
    worksheetData.Cells["A1"].Value = "Column A";
    worksheetData.Cells["A2"].Value = "Group A";
    worksheetData.Cells["A3"].Value = "Group B";
    worksheetData.Cells["A4"].Value = "Group C";
    worksheetData.Cells["A5"].Value = "Group A";
    worksheetData.Cells["A6"].Value = "Group B";
    worksheetData.Cells["A7"].Value = "Group C";
    worksheetData.Cells["A8"].Value = "Group A";
    worksheetData.Cells["A9"].Value = "Group B";
    worksheetData.Cells["A10"].Value = "Group C";
    worksheetData.Cells["A11"].Value = "Group D";

    worksheetData.Cells["B1"].Value = "Column B";
    worksheetData.Cells["B2"].Value = "emc";
    worksheetData.Cells["B3"].Value = "fma";
    worksheetData.Cells["B4"].Value = "h2o";
    worksheetData.Cells["B5"].Value = "emc";
    worksheetData.Cells["B6"].Value = "fma";
    worksheetData.Cells["B7"].Value = "h2o";
    worksheetData.Cells["B8"].Value = "emc";
    worksheetData.Cells["B9"].Value = "fma";
    worksheetData.Cells["B10"].Value = "h2o";
    worksheetData.Cells["B11"].Value = "emc";

    worksheetData.Cells["C1"].Value = "Column C";
    worksheetData.Cells["C2"].Value = 299;
    worksheetData.Cells["C3"].Value = 792;
    worksheetData.Cells["C4"].Value = 458;
    worksheetData.Cells["C5"].Value = 299;
    worksheetData.Cells["C6"].Value = 792;
    worksheetData.Cells["C7"].Value = 458;
    worksheetData.Cells["C8"].Value = 299;
    worksheetData.Cells["C9"].Value = 792;
}
```

```

worksheetData.Cells["C10"].Value = 458;
worksheetData.Cells["C11"].Value = 299;

worksheetData.Cells["D1"].Value = "Column D";
worksheetData.Cells["D2"].Value = 40075;
worksheetData.Cells["D3"].Value = 31415;
worksheetData.Cells["D4"].Value = 384400;
worksheetData.Cells["D5"].Value = 40075;
worksheetData.Cells["D6"].Value = 31415;
worksheetData.Cells["D7"].Value = 384400;
worksheetData.Cells["D8"].Value = 40075;
worksheetData.Cells["D9"].Value = 31415;
worksheetData.Cells["D10"].Value = 384400;
worksheetData.Cells["D11"].Value = 40075;

//define the data range on the source sheet
var dataRange = worksheetData.Cells[worksheetData.Dimension.Address];

//create the pivot table
var pivotTable = worksheetPivot.PivotTables.Add(worksheetPivot.Cells["B2"], dataRange,
"PivotTable");

//label field
pivotTable.RowFields.Add(pivotTable.Fields["Column A"]);
pivotTable.DataOnRows = false;

//data fields
var field = pivotTable.DataFields.Add(pivotTable.Fields["Column B"]);
field.Name = "Count of Column B";
field.Function = DataFieldFunctions.Count;

field = pivotTable.DataFields.Add(pivotTable.Fields["Column C"]);
field.Name = "Sum of Column C";
field.Function = DataFieldFunctions.Sum;
field.Format = "0.00";

field = pivotTable.DataFields.Add(pivotTable.Fields["Column D"]);
field.Name = "Sum of Column D";
field.Function = DataFieldFunctions.Sum;
field.Format = "€#,##0.00";
}

```

Leggi Tabella pivot online: <https://riptutorial.com/it/epplus/topic/8767/tabella-pivot>

Capitolo 13: tabelle

introduzione

Questo argomento descrive come aggiungere e assegnare stili alle tabelle

Examples

Aggiunta e formatura di una tabella

```
//Using statement for ExcelTable and TableStyles
using OfficeOpenXml.Table;

//Defining the tables parameters
int firstRow =1;
int lastRow = worksheet.Dimension.End.Row;
int firstColumn = 1;
int lastColumn = worksheet.Dimension.End.Column;
ExcelRange rg = worksheet.Cells[firstRow, firstColumn, lastRow, LastColumn];
string tableName = "Table1";

//Ading a table to a Range
ExcelTable tab = worksheet.Tables.Add(rg, tableName);

//Formating the table style
tab.TableStyle = TableStyles.Light8;
```

Leggi tabelle online: <https://riptutorial.com/it/epplus/topic/8720/tabelle>

Capitolo 14: Unire le celle

introduzione

Come unire le cellule

Examples

Unione di celle

```
//By range address  
worksheet.Cells["A1:B5"].Merge = true;  
  
//By indexes  
worksheet.Cells[1,1,5,2].Merge = true;
```

Leggi Unire le celle online: <https://riptutorial.com/it/epplus/topic/8728/unire-le-celle>

Capitolo 15: Valori di formattazione

introduzione

Come ottenere la formattazione desiderata di DateTime e valori numerici.

Examples

Formattazione del numero

```
//integer (not really needed unless you need to round numbers, Excel with use default cell
properties)
worksheet.Cells["A1:A25"].Style.Numberformat.Format = "0";

//integer without displaying the number 0 in the cell
worksheet.Cells["A1:A25"].Style.Numberformat.Format = "#";

//number with 1 decimal place
worksheet.Cells["A1:A25"].Style.Numberformat.Format = "0.0";

//number with 2 decimal places
worksheet.Cells["A1:A25"].Style.Numberformat.Format = "0.00";

//number with 2 decimal places and thousand separator
worksheet.Cells["A1:A25"].Style.Numberformat.Format = "#,##0.00";

//number with 2 decimal places and thousand separator and money symbol
worksheet.Cells["A1:A25"].Style.Numberformat.Format = "€#,##0.00";

//percentage (1 = 100%, 0.01 = 1%)
worksheet.Cells["A1:A25"].Style.Numberformat.Format = "0%";
```

Formattazione della data

```
//default DateTime patterns
worksheet.Cells["A1:A25"].Style.Numberformat.Format =
DateTimeFormatInfo.CurrentInfo.ShortDatePattern;

//custom DateTime patterns
worksheet.Cells["A1:A25"].Style.Numberformat.Format = "dd-MM-yyyy HH:mm";

//or overwrite the patterns in the CurrentThread with your own
Thread.CurrentThread.CurrentCulture = new CultureInfo("nl-NL")
{
    DateTimeFormat = { YearMonthPattern = "MMM yy" }
};
worksheet.Cells["A1:A25"].Style.Numberformat.Format =
DateTimeFormatInfo.CurrentInfo.YearMonthPattern;
```

Formato testo

```
worksheet.Cells["A1:A25"].Style.Numberformat.Format = "@";
```

Leggi Valori di formattazione online: <https://riptutorial.com/it/epplus/topic/8080/valori-di-formattazione>

Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con eppplus	Community , Magnetron , VDWWD
2	Aggiungi dati al documento esistente	VDWWD
3	Colonne e righe	hellyale , Magnetron , VDWWD
4	Convalida dell'input dell'utente	Magnetron
5	Creare formule e calcolare intervalli	Magnetron , VDWWD
6	Creazione di grafici	VDWWD
7	Disegnare il documento Excel	Magnetron , VDWWD
8	Importazione di dati dal file esistente	VDWWD
9	Rich Text in cells	Pete
10	Riempiendo il documento con i dati	VDWWD
11	Salvataggio del documento Excel	Magnetron , VDWWD
12	Tabella pivot	VDWWD
13	tabelle	Magnetron
14	Unire le celle	Magnetron
15	Valori di formattazione	Magnetron , VDWWD