

ZetCode

[כולם](#) [גו](#) [פייתון](#) [JavaScript](#) [C#](#) [הירשם כמנוי](#)

C# Excel

שונה לאחרונה ב- 3 בדצמבר 2021

ערכת לימוד C# Excel מראה כיצד לעבוד עם קבצי Excel ב-C# עם ספריית ClosedXML.

Excel xlsx

במדריך זה אנו עובדים עם קבצי xlsx. ה-xlsx הוא סיומת קובץ עבור תבנית קובץ פתוחה של גיליון אלקטרוני של XML המשמשת את Microsoft Excel. קבצי xlsx תומכים בפקודות מאקרו. ה-xltm הם קבצי תבנית המותאמים לשימוש במאקרו. תבנית xls היא תבנית בינארית קניינית בעוד ש-xlsx מבוסס על תבנית Office Open XML.

סגור XML

ClosedXML היא ספריית .NET לקריאה, טיפול וכתיבה של קבצי xlsx, xlsx (Excel 2007+).

```
$ dotnet add package closedxml
```

נוסיף את החבילה לפרויקט.

דוגמה פשוטה של C# Excel

בדוגמה הראשונה, אנו יוצרים קובץ xlsx חדש עם ClosedXML.

תוכנית CS

```
using ClosedXML.Excel;

using var wbook = new XLWorkbook();

var ws = wbook.Worksheets.Add("Sheet1");
ws.Cell("A1").Value = "150";

wbook.SaveAs("simple.xlsx");
```

אנו יוצרים קובץ Excel חדש וכותבים ערך לתא.

```
using var wbook = new XLWorkbook();
```

נוצר חדש. XLWorkbook

```
var ws = wbook.Worksheets.Add("Sheet1");
```

אנו מוסיפים גיליון חדש לחוברת העבודה.

```
ws.Cell("A1").Value = "150";
```

ערך נכתב לתא עם כתובת A1.

```
wbook.SaveAs("simple.xlsx");
```

חוברת העבודה נשמרת בשיטה. SaveAs

תא Excel C#

תא הוא הצטלבות של שורה ועמודה. לכל תא יש כתובת ייחודית המורכבת מאות העמודה ומספר השורה שלו. לדוגמה, לתא הראשון הממוקם בפינה הימנית העליונה של גיליון יש כתובת A1.

תוכנית. CS

```
using ClosedXML.Excel;

using var wbook = new XLWorkbook();

var ws = wbook.AddWorksheet("Sheet1");

ws.FirstCell().Value = 150;

ws.Cell(3, 2).Value = "Hello there!";
ws.Cell("A6").SetValue("falcon").SetActive();

ws.Column(2).AdjustToContents();

wbook.SaveAs("data.xlsx");
```

בדוגמה, אנחנו עובדים עם תאים.

```
ws.FirstCell().Value = 150;
```

האחזר את ההפניה לתא הראשון בגיליון. הגדרנו את ערכו ל-150 באמצעות הנכס. FirstCellValue

```
ws.Cell(3, 2).Value = "Hello there!";
```

דרך נוספת להפנות לתא משמשת בשיטה עמוסה מדי, המתייחסת למספרי השורות והעמודות כפרמטרים. Cell

```
ws.Cell("A6").SetValue("falcon").SetActive();
```

כאן, אנו מקבלים תא דרך הכתובת שלו (A6), ומשתמשים בשיטה כדי לכתוב מחרוזת לתא. השיטה הופכת את התא לפעיל. **SetValueSetActive**

```
ws.Column(2).AdjustToContents();
```

אנו מתאימים את רוחב העמודה השניה לתוכן; הערך גלוי לחלוטין. **!Hello there**

#C קרא קובץ Excel

בדוגמה הבאה, אנו קוראים מקובץ Excel שנוצר בעבר.

תוכנית .CS

```
using ClosedXML.Excel;

using var wbook = new XLWorkbook("simple.xlsx");

var ws1 = wbook.Worksheet(1);
var data = ws1.Cell("A1").GetValue<string>();

Console.WriteLine(data);
```

הדוגמה קוראת ערך מתא.

```
using var wbook = new XLWorkbook("simple.xlsx");
```

אנחנו פותחים חוברת עבודה.

```
var ws1 = wbook.Worksheet(1);
```

אנחנו מנווטים אל הגיליון הראשון.

```
var data = ws1.Cell("A1").GetValue<string>();
```

באמצעות השיטה, אנו קוראים ערך מתא **A1.GetValue**

סגנון החלת #C Excel

ניתן להחיל סגנון באמצעות מקום האירוח. **Style**

תוכנית .CS

```

using ClosedXML.Excel;

using var wbook = new XLWorkbook();

var ws = wbook.Worksheets.Add("Sheet1");

var c1 = ws.Column("A");
c1.Width = 25;

var c2 = ws.Column("B");
c2.Width = 15;

ws.Cell("A3").Value = "an old falcon";
ws.Cell("B2").Value = "150";
ws.Cell("B5").Value = "Sunny day";

ws.Cell("A3").Style.Alignment.Horizontal = XLAlignmentHorizontalValues.Center;
ws.Cell("A3").Style.Alignment.Vertical = XLAlignmentVerticalValues.Center;
ws.Cell("A3").Style.Font.Italic = true;

ws.Cell("B2").Style.Border.OutsideBorder = XLBorderStyleValues.Thin;
ws.Cell("B5").Style.Font.FontColor = XLColor.Red;

wbook.SaveAs("styled.xlsx");

```

בדוגמה, החל סגנונות על תאים.

```

var c1 = ws.Column("A");
c1.Width = 25;

var c2 = ws.Column("B");
c2.Width = 15;

```

הגדרנו את הרוחב של עמודות A ו-B.

```

ws.Cell("A3").Style.Alignment.Horizontal = XLAlignmentHorizontalValues.Center;
ws.Cell("A3").Style.Alignment.Vertical = XLAlignmentVerticalValues.Center;
ws.Cell("A3").Style.Font.Italic = true;

```

בתא A3, אנו מרכזים את הטקסט אופקית ואנכית ובוחרים גופן תוחם.

```
ws.Cell("B2").Style.Border.OutsideBorder = XLBorderStyleValues.Thin;
```

קבענו גבול דק לתא B2.

```
ws.Cell("B5").Style.Font.FontColor = XLColor.Red;
```

בתא B5, צבע הגופן מוגדר לאדום.

טווחי Excel C#

טווח הוא קבוצה של תא אחד או יותר. כתובת טווח מצוינת עם כתובת תא למעלה וכתובת תחתון, המופרדת באמצעות נקודתיים.

נוצר טווח יחיד עם ; טווחים מרובים נוצרים באמצעות . RangeRanges

CS. תוכנית

```
using ClosedXML.Excel;

using var wbook = new XLWorkbook();

var ws = wbook.Worksheets.Add("Sheet1");

ws.Range("D2:E2").Style.Fill.BackgroundColor = XLColor.Gray;
ws.Ranges("C5, F5:G8").Style.Fill.BackgroundColor = XLColor.Gray;

var rand = new Random();
var range = ws.Range("C10:E15");

foreach (var cell in range.Cells())
{
    cell.Value = rand.Next();
}

ws.Column("C").AdjustToContents();
ws.Column("D").AdjustToContents();
ws.Column("E").AdjustToContents();

wbook.SaveAs("ranges.xlsx");
```

בדוגמה, אנו עובדים עם טווחים.

```
ws.Range("D2:E2").Style.Fill.BackgroundColor = XLColor.Gray;
```

אנו משנים את צבע הרקע לטווח. D2:E2

```
ws.Ranges("C5, F5:G8").Style.Fill.BackgroundColor = XLColor.Gray;
```

כאן, אנו משנים את צבע הרקע עבור שני טווחים.

```
var rand = new Random();
var range = ws.Range("C10:E15");

foreach (var cell in range.Cells())
{
    cell.Value = rand.Next();
}
```

בכל תא של הטווח, אנו מגדירים ערך אקראי. אנו משתמשים בשיטה כדי לקבל את התאים של הטווח.

C10:E15Cells

```
ws.Column("C").AdjustToContents();
```

אנו מתאימים את רוחב העמודה לתוכן הכתוב כך שכל הערך יהיה גלוי. C

Excel C# מזג תאים

כדי למזג תאים, אנו משתמשים בשיטה. Merge

CS. תוכנית

```
using ClosedXML.Excel;

using var wbook = new XLWorkbook();

var ws = wbook.Worksheets.Add("Sheet1");

ws.Cell("A1").Value = "Sunny day";
ws.Cell("A1").Style.Alignment.Horizontal = XLAlignmentHorizontalValues.Center;
ws.Cell("A1").Style.Alignment.Vertical = XLAlignmentVerticalValues.Center;

ws.Range("A1:B2").Merge();

wbook.SaveAs("merged.xlsx");
```

אנחנו ממזגים ארבעה תאים לאחד.

```
ws.Range("A1:B2").Merge();
```

אנו קוראים לשיטה על טווח של ארבעה תאים; A1 עד B2. Merge

מיזן Excel C#

ניתן למיין עמודה באמצעות פעולת השירות. Sort

CS. תוכנית

```
using ClosedXML.Excel;

using var wbook = new XLWorkbook();

var ws = wbook.Worksheets.Add("Sheet1");

var rand = new Random();
var range = ws.Range("A1:A15");
```

```
foreach (var cell in range.Cells())
{
    cell.Value = rand.Next(1, 100);
}

ws.Sort("A");

wbook.SaveAs("sorted.xlsx");
```

We add fifteen random values to the A column. Then we sort the column with . Sort

C# Excel CellsUsed

The method returns a collection of cells that have a value. CellsUsed

Program.cs

```
using ClosedXML.Excel;

using var wbook = new XLWorkbook();

var ws = wbook.Worksheets.Add("Sheet1");

ws.Cell("A1").Value = "sky";
ws.Cell("A2").Value = "cloud";
ws.Cell("A3").Value = "book";
ws.Cell("A4").Value = "cup";
ws.Cell("A5").Value = "snake";
ws.Cell("A6").Value = "falcon";
ws.Cell("B1").Value = "in";
ws.Cell("B2").Value = "tool";
ws.Cell("B3").Value = "war";
ws.Cell("B4").Value = "snow";
ws.Cell("B5").Value = "tree";
ws.Cell("B6").Value = "ten";

var n = ws.Range("A1:C10").CellsUsed().Count();
Console.WriteLine($"There are {n} words in the range");

Console.WriteLine("The following words have three latin letters:");

var words = ws.Range("A1:C10")
    .CellsUsed()
    .Select(c => c.Value.ToString())
    .Where(c => c?.Length == 3)
    .ToList();

words.ForEach(Console.WriteLine);

wbook.SaveAs("usedcells.xlsx");
```

In the example, we write twelve words to two columns of a sheet.

```
var n = ws.Range("A1:C10").CellsUsed().Count();
Console.WriteLine($"There are {n} words in the range");
```

We define a range that is larger than the cells we have written to. With the help of the method, we get all non-empty cells. The method counts the number of non-empty cells. `CellsUsedCount`

```
var words = ws.Range("A1:C10")
    .CellsUsed()
    .Select(c => c.Value.ToString())
    .Where(c => c?.Length == 3)
    .ToList();
```

Here we filter all words that have three latin letters.

C# Excel expression evaluation

With , we can evaluate an expression. `Evaluate`

Program.cs

```
using ClosedXML.Excel;

using var wbook = new XLWorkbook("data.xlsx");

var ws = wbook.Worksheet(1);

var sum = ws.Evaluate("SUM(A1:A7)");
var max = ws.Evaluate("MAX(A1:A7)");

Console.WriteLine($"The sum is: {sum}");
Console.WriteLine($"The maximum value is: {max}");
```

In the example, we have values in the A column. We evaluate SUM and MAX expressions on the values.

C# Excel formula

With the property, we can insert a formula in the cell. `FormulaA1`

Program.cs

```
using ClosedXML.Excel;

using var wbook = new XLWorkbook("data.xlsx");

var ws = wbook.Worksheet(1);
```



```
ws.Cell("A8").FormulaA1 = "SUM(A1:A7)";  
ws.Cell("A8").Style.Font.Bold = true;  
  
wbook.SaveAs("data2.xlsx");
```

In the example, we insert the formula in the cell below the values. SUM

In this tutorial, we have demonstrated how to work with Excel files with C# using ClosedXMLlibrary.

Read [C# tutorial](#) or list [all C# tutorials](#).

[Home](#) [Facebook](#) [Twitter](#) [Github](#) [Subscribe](#) [Privacy](#)

© 2007 - 2021 Jan Bodnar admin(at)zetcode.com