

Ordered and unordered (bulleted) lists in PDF

Written by Apitron Documentation Team

Introduction

Lists are well known feature of HTML and CSS, and can be defined to be either unordered, using `` html element or ordered using `` html element. List item can be defined using `` element. In opposite to HTML, PDF doesn't have a special command or object to produce lists. They can only be crafted manually using all available PDF's drawing capabilities.

Our .NET pdf component, [Apitron PDF Kit](#), offers a flow layout API which makes it possible to create lists just like in HTML, taking care of all PDF specifics. It also supports standard html list bullets and numerations, e.g. circle, diamond, square, triangle; latin, roman, decimal - with their lower and upper variations, and also, custom markers.

Lists in flow layout API

Lists in Flow layout API can be created using `Section` objects, which become containers for the list items. The following properties of the section affect the list's appearance:

- `ListStyle` – defines whether the list is ordered, unordered or it's a list item in other list
- `ListCounter` – the starting value for item's counter in ordered lists, default value is 1
- `ListMarker` – the type of marker to be used by list items
- `ListMarkerPadding` – sets the padding around item's marker

List items can be defined using `ListStyle = ListItem`, and they have the following properties:

- `ListMarker` – the type of marker used by an item
- `ListMarkerPadding` – sets the padding around the item's marker

Therefore, a list can be created using a section with appropriate style set + one or several items with `ListStyle= ListStyle.ListItem`, which is an equivalent to `` html element.

Multilevel numbered lists can be created by enclosing one ordered list into another, and this technique will be shown in one of the samples below.

Unordered lists

Unordered or bulleted lists are lists where each item is marked with a specific bullet without implying a particular order. The only order that exists is the order of items creation. Let's check the code sample below that shows how to create unordered lists:

```
public void CreateUnorderedLists()
{
    // prepare resources
    ResourceManager resourceManager = new ResourceManager();
    resourceManager.RegisterResource(new
    Apitron.PDF.Kit.FixedLayout.Resources.XObjects.Image("smile",
    "smile.jpg"));

    // create and fill document
    FlowDocument doc = new FlowDocument() { Margin = new Thickness(5, 5, 5, 5) };

    // define unordered list style
    doc.StyleManager.RegisterStyle(".ul", new Style()
    {
        ListStyle = ListStyle.Unordered,
        Width = Length.FromPercentage(20),
        ListMarkerPadding = new Thickness(2),
        Display = Display.InlineBlock,
        Font = new Font(StandardFonts.Helvetica,24)
    });

    // define style for items added to the unordered list
    doc.StyleManager.RegisterStyle(".ul > *", new Style() { ListStyle = ListStyle.ListItem });

    // create an array of markers for our lists
    ListMarker[] markers = new ListMarker[] {ListMarker.Circle,ListMarker.Diamond,
    ListMarker.Triangle, ListMarker.Square, ListMarker.FromResourceId("smile") };

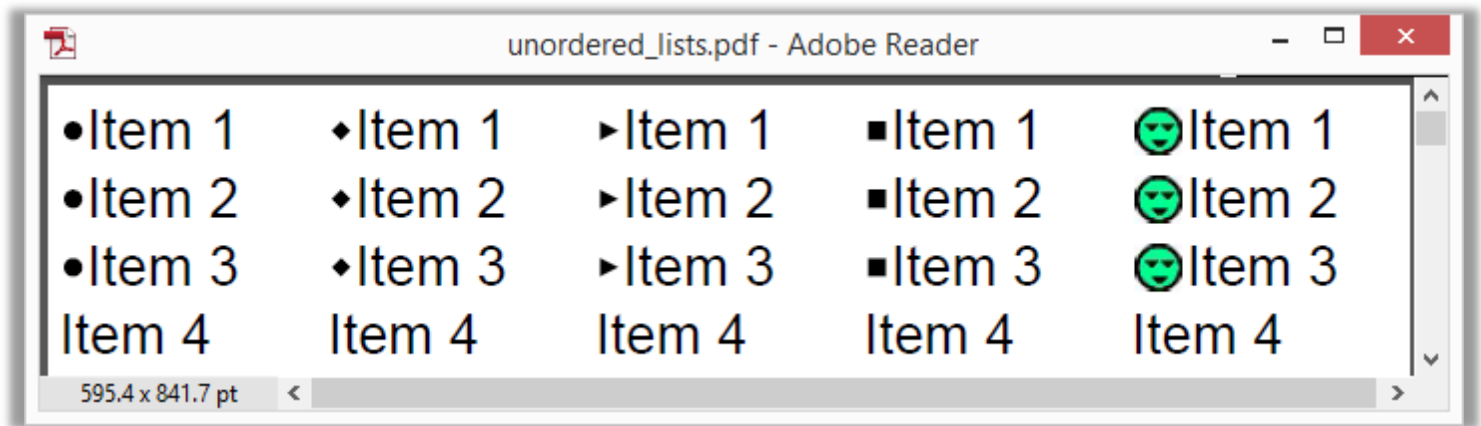
    // create 5 lists with different markers
    foreach (ListMarker marker in markers)
    {
        doc.Add(new Section(new TextBlock("Item 1"), new TextBlock("Item 2"),
        new TextBlock("Item 3"),new TextBlock("Item 4"){ListStyle = ListStyle.None})
        {Class = "ul", ListMarker = marker});
    }

    // save document
    using (Stream stream = File.Create("unordered_lists.pdf"))
    {
        doc.Write(stream, resourceManager);
    }
}
```

This code creates five bulleted lists with different list markers. Lists are made of sections styled via the class selector “ul”. All their child items become list items because of another style matching children of an element with class “ul” set. Please also note that each list has the unmarked item which is not the list item at all, it’s just a child item in the section. We changed its appearance by setting an inline style property `ListStyle` to `None`.

Last list has the custom image list marker set; it’s possible to use any `XObject` as a list marker for unordered list elements.

See the resulting image below:



Pic. 1 Unordered lists sample

Ordered lists

Ordered lists or numbered lists can be used to show the explicit order of the elements, and usually apply decimal, roman, or latin numeration to their items.

The code below creates ordered lists:

```
public void CreateOrderedList()
{
    // prepare resources
    ResourceManager resourceManager = new ResourceManager();

    // create and fill document
    FlowDocument doc = new FlowDocument() { Margin = new Thickness(5, 5, 5, 5) };

    // define ordered list style
    doc.StyleManager.RegisterStyle(".ol", new Style()
    {
        ListStyle = ListStyle.Ordered,
        Width = Length.FromPercentage(20),
        ListMarkerPadding = new Thickness(2),
        Display = Display.InlineBlock, Font = new Font(StandardFonts.Helvetica, 24)
    });
    // define style for items added to the ordered list
    doc.StyleManager.RegisterStyle(".ol > *", new Style() { ListStyle = ListStyle.ListItem });

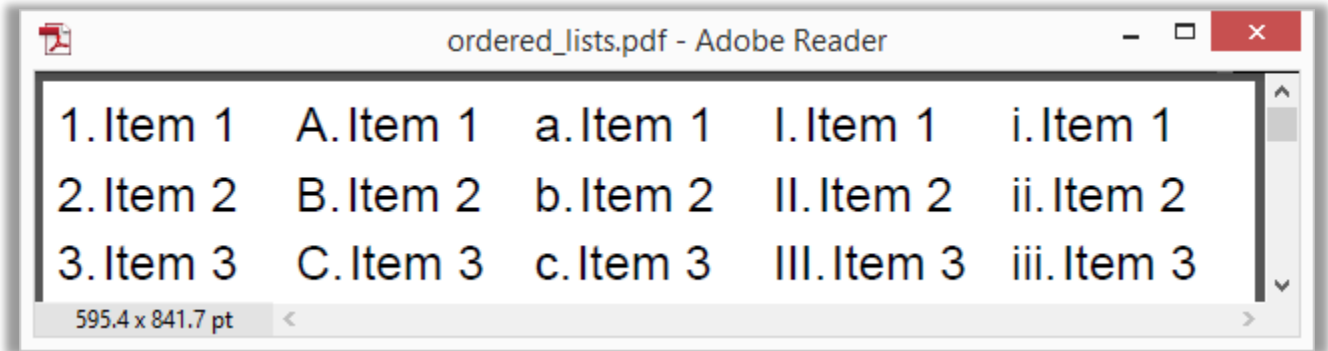
    // create an array of markers for our lists
    ListMarker[] markers = new ListMarker [] { ListMarker.Decimal, ListMarker.UpperLatin,
    ListMarker.LowerLatin, ListMarker.UpperRoman, ListMarker.LowerRoman };

    // create 5 ordered lists with different markers
    foreach (ListMarker marker in markers)
    {
        doc.Add(new Section(new TextBlock("Item 1"), new TextBlock("Item 2"),
        new TextBlock("Item3")))
        {
            Class = ".ol", ListMarker = marker,
            ListCounter = (marker == ListMarker.LowerLatin || marker == ListMarker.UpperLatin) ?
            new ListCounter(0) : null
        });
    }

    // save document
    using (Stream stream = File.Create("ordered_lists.pdf"))
    {
        doc.Write(stream, resourceManager);
    }
}
```

This code creates five numbered lists each having its own marker, for latin markers we set initial value of the list counter to zero because default value for it is 1.

Resulting image is provided below:



Pic. 2 Ordered lists sample

Multi-level ordered lists

It's possible to combine several ordered lists to create multi-level lists, see the code below:

```
public static void MultilevelOrderedList()
{
    // prepare resources
    ResourceManager resourceManager = new ResourceManager();

    // create and fill document
    FlowDocument doc = new FlowDocument() { Margin = new Thickness(5, 5, 5, 5) };

    // define ordered list style
    doc.StyleManager.RegisterStyle(".ol", new Style()
    {
        ListStyle = ListStyle.Ordered,
        ListMarkerPadding = new Thickness(2),
        Font = new Font(StandardFonts.Helvetica, 20)
    });

    // define style for items added to the ordered list
    doc.StyleManager.RegisterStyle(".ol > textblock", new Style(){ListStyle = ListStyle.ListItem});

    // create top list
    Section topList = new Section(new TextBlock("Item 1"), new TextBlock("Item 2"),
    new TextBlock("Item3")){ Class = "ol", ListMarker = ListMarker.Decimal };

    // create inner list level 1
    Section firstLevelList = new Section(new TextBlock("Inner item 1"),
    new TextBlock("Inner item 2"), new TextBlock("Inner item 3"))
    { Class = "ol", ListMarker = ListMarker.Decimal };

    // create inner list level2
    Section secondLevelList = new Section(new TextBlock("Inner item 1"),
    new TextBlock("Inner item 2"), new TextBlock("Inner item 3"))
    { Class = "ol", ListMarker = ListMarker.Decimal };

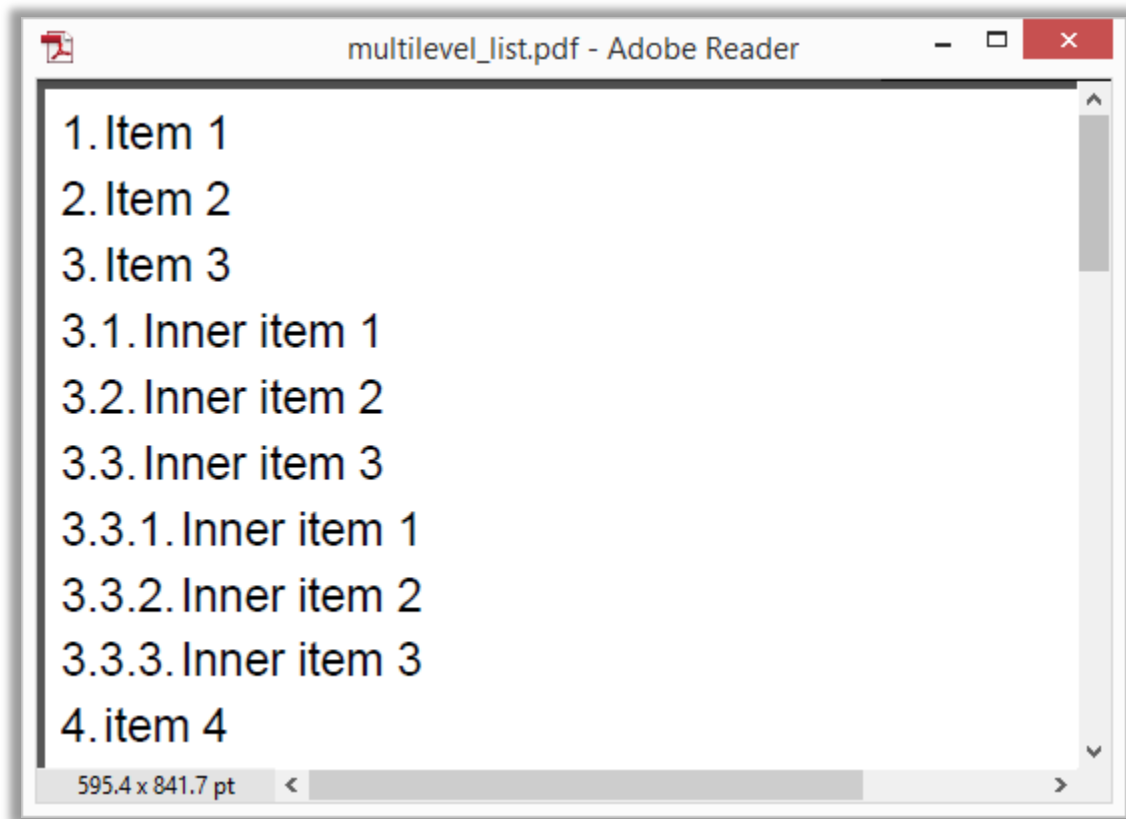
    // combine lists
    firstLevelList.Add(secondLevelList);
    topList.Add(firstLevelList);

    // add text block to the top list
    topList.Add(new TextBlock("item 4"));

    // insert to list in the document
    doc.Add(topList);

    // save document
    using (Stream stream = File.Create("multilevel_list.pdf"))
    {
        doc.Write(stream, resourceManager);
    }
}
```

This code creates a list having three levels, see the image representing produced document. Notice the dots separating the list levels.



Pic. 3 Multi-level list sample

Conclusion

Working with ordered and unordered lists is easy and simple with Flow layout API offered by [Aptron PDF Kit](#) component. This pdf library offers unmatched flexibility and can be used to create applications for many modern mobile, desktop and cloud platforms. Develop for Windows Forms, WPF, Android, iOS, Windows Phone, ASP.NET, Azure, MONO, Windows Store, and use the same API producing professionally looking PDF documents.

Contact us if you need any help with pdf processing, and we'll provide you with a free consultation and pdf expert's advice.