



Tutorial

# How to Get Started with the IronPDF C# PDF Library

## HTML String to PDF

```
1. using IronPdf;
2. namespace IronPDFSample
3. {
4.     class Program
5.     {
6.         static void Main()
7.         {
8.             var Renderer = new HtmlToPdf();
```

# IronPDF C# PDF Library

Interact with the tutorial: <https://ironpdf.com/docs/>

Share the tutorial: [✉](#) [f](#) [@](#) [in](#) [t](#)


IronPDF takes care of the difficult problem of adding PDF generation to your app, and automates turning formatted documents into a PDF.

- Convert web forms, local HTML pages, and other web pages to PDF with .NET
- Allow users to download documents, send them by email, or store them in the cloud.
- Produce invoices, quotes, reports, contracts, and other documents.
- Work with ASP.NET, ASP .NET Core, web forms, MVC, Web APIs on .NET Framework, and .NET Core.

## Table of Contents

1. Install the IronPDF C# Library to your project
2. Render HTML String to PDF
3. Convert HTML File to PDF
4. Render Existing URL to PDF
5. ASP.NET Web Forms to PDF
6. Route ASP MVC View to PDF
7. Add Headers and Footers
8. Encrypt PDFs with a Password
9. Merge and Split PDF Documents
10. Extract Images from PDF Documents
11. Enable JavaScript
12. Use OCR Scanning
13. Use More Printing Options
14. Download the C# PDF Cheat Sheet
15. Learn More

# 1. Install the IronPDF C# Library to your project

|  |    |   |
|--|----|---|
| <br><b>Download DLL</b><br>Manually install into your project | or | <br><b>Install with NuGet</b><br><a href="https://nuget.org/packages/IronPdf/">nuget.org/packages/IronPdf/</a> |
|--|----|---|

## 1.1. Install with NuGet Package Manger

Install IronPDF in Visual Studio or at the command line with the NuGet Package Manager. In Visual Studio, navigate to the console with:

- {page} for the current page number
- {total-pages} for the total number of pages in the PDF
- {url} for the URL of the rendered PDF if rendered from a web page
- {date} for today's date

```
PM > Install-Package IronPdf
```

And check out [IronPDF on NuGet](#) for more about version updates and installation.

## 1.2. Directly Download the DLL

Alternatively, you can [directly download the DLL](#).

Remember to add this statement to the top of any **cs** class file using IronPDF:

```
using IronPdf;
```

## 1.3. Install and Deploy the Library

For more details, check the [guide on how to install and deploy the IronPDF C# Library](#).

## 2. Render HTML String to PDF

IronPDF can render HTML text to PDF quite easily. This example illustrates the capability. Use this option when you only need to add simple text to your PDF document.

- Create a new .NET Core console application
- Install the NuGet package
- Import the `IronPdf` namespace with the `using` keyword
- Create a new `HtmlToPdf` renderer
- Call `RenderHtmlAsPdf` and then `SaveAs` on the result.

```
1. using IronPdf;
2.
3. namespace IronPDFSample
4. {
5.     class Program
6.     {
7.         static void Main()
8.         {
9.             var Renderer = new HtmlToPdf();
10.            Renderer.RenderHtmlAsPdf("<h1>Hello World</h1>").SaveAs("html-string.pdf");
11.        }
12.    }
13. }
```

### 3. Convert HTML File to PDF

You can render HTML files with images, CSS, forms, hyperlinks, and JavaScript as a PDF document. Use this method for scenarios where you have access to the source document locally.

This example calls `RenderHtmlAsPdf`, which returns a variable called PDF.

Call `SaveAs` to save the output to a PDF file.

The sample assumes that there is a HTML file in the folder Assets.

```
1. using IronPdf;
2.
3. namespace IronPDFSample
4. {
5.     class Program
6.     {
7.         static void Main()
8.         {
9.             // Create a PDF from an existing HTML using C#
10.            var Renderer = new HtmlToPdf();
11.            var PDF = Renderer.RenderHTMLFileAsPdf("Assets/MyHTML.html");
12.            PDF.SaveAs("MyPdf.pdf");
13.        }
14.    }
15. }
```