Esto es un titulo

Programmatic generation of PDF files is a frequent task when developing applications that can export reports, bills, or questionnaires. In this article, we will consider three common tools for creating PDFs, including their installation and converting principles.

PDF stands for Portable Document Format and it was originally developed by Adobe, though, it has now become an open standard for text files. Creating a single PDF file from a Microsoft Word document can be easily done through the Word's menu, print dialogue in Linux or MacOS, or Adobe Acrobat Reader. At the same time, when you need to generate tens, hundreds, or even thousands of PDF files, it is better to automate this task. For generating PDFs with Python, we have chosen the following solutions: Xhtml2pdf, Weasyprint, and Unoconv.

How to Generate PDF in Python with Xhtml2pdf

The main drawback of all HTML to PDF converters is that the latter has numerous aspects that are absent in HTML, including, for example, page size. Xhtml2pdf deals with this problem by adding specific markup tags that allow solving various tasks, such as converting headers and footers on all pages. Xhtml2pdf is a CSS/HTML to PDF generator/converter and Python library that can be used in any Python framework such as Django.

In fact, to create PDFs, the tool uses ReportLab, a common open source Python library for generating PDF files. ReportLab uses an XML-based markup language called Requirements Modelling Language (RML). Therefore, you can think of Xhtml2pdf as another markup language for the ReportLab library.

Installation

You can easily install Xthml2pdf using a common command:

pip install xhtml2pdf

Generating PDFs

Xhtml2pdf usage example: