

# Mastering PDF

Create, read and edit  
PDF files in your  
PHP applications

with  
PHP

Piotr Horzycki

# Mastering PDF with PHP

Create, read and edit PDF files in your PHP applications

Piotr Horzycki

This book is for sale at <http://leanpub.com/mastering-pdf-with-php>

This version was published on 2023-10-24



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

© 2021 - 2023 Piotr Horzycki

# **Tweet This Book!**

Please help Piotr Horzycki by spreading the word about this book on [Twitter](#)!

The suggested hashtag for this book is [#MasteringPDFwithPHP](#).

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

[#MasteringPDFwithPHP](#)

# Contents

<b>Introduction</b>	<b>1</b>
Who is this book for?	1
How to read this book?	1
About the author	2
<b>What is Portable Document Format for?</b>	<b>3</b>
Quick look at the history	3
Main features	3
Which version to use?	3
Differences between HTML and PDF	3
Is PDF a right choice for me?	3
<b>The structure of a PDF file</b>	<b>4</b>
Main goals	4
Four parts of a file	4
Write your own PDF in Notepad	4
Compression	5
Summary	5
<b>An example document: an invoice</b>	<b>6</b>
An example invoice	6
Basic HTML structure	6
Body structure	6
Creating a stylesheet	7
Validating document syntax	7
Summary	7
<b>Choosing a tool to generate PDF files</b>	<b>8</b>
Two ways to create a PDF file	8
Native PHP libraries	8
External conversion tools	9
Summary	10
<b>Presenting a document to a user</b>	<b>11</b>
Displaying a PDF in a frame	11

## CONTENTS

Browser window . . . . .	11
Downloading a PDF . . . . .	11
More about filenames . . . . .	11
Sending a PDF over email . . . . .	12
Saving a file . . . . .	12
Display preferences . . . . .	12
Summary . . . . .	13
<b>Understanding fonts . . . . .</b>	<b>14</b>
How fonts work? . . . . .	14
Font file formats in the PDF . . . . .	15
Picking a proper font . . . . .	15
Selecting a font in CSS . . . . .	17
Text transformations . . . . .	22
ANSI or Mac versus Unicode . . . . .	24
Summary . . . . .	26
<b>Metadata . . . . .</b>	<b>27</b>
How it looks inside . . . . .	27
Entering metadata with TCPDF . . . . .	27
mPDF example . . . . .	27
Dompdf example . . . . .	27
WeasyPrint example . . . . .	27
Summary . . . . .	27
<b>Encryption . . . . .</b>	<b>28</b>
Choosing the cipher . . . . .	28
User permissions . . . . .	28
Encryption with TCPDF . . . . .	28
Encryption with mPDF . . . . .	28
Encryption with Dompdf . . . . .	28
Encryption with FPDF . . . . .	28
Encrypting an existing file with command line tools . . . . .	29
Summary . . . . .	29
<b>Calling external programs from a PHP script . . . . .</b>	<b>30</b>
Input, output and exit codes . . . . .	30
Basic execution from a PHP script . . . . .	30
Escaping arguments . . . . .	30
Opening and controlling a process . . . . .	30
The most convenient solution: The Process Component . . . . .	30
Reporting progress of time-consuming tasks . . . . .	31
Queueing tasks . . . . .	31
Summary . . . . .	31

## CONTENTS

<b>Styling pages with CSS</b> . . . . .	<b>32</b>
Page dimensions and margins . . . . .	32
Adjusting individual pages . . . . .	32
Page-margin boxes and generated content . . . . .	32
Bleed area and crop marks . . . . .	32
Controlling page breaks . . . . .	32
Media queries . . . . .	33
Summary . . . . .	33
<b>Reading and testing PDF files</b> . . . . .	<b>34</b>
Metadata and text extraction . . . . .	34
Taking a screenshot . . . . .	34
Summary . . . . .	35
<b>Editing PDF files</b> . . . . .	<b>36</b>
Importing a document to FPDF or TCPDF . . . . .	36
The PDF Toolkit . . . . .	36
Manipulating files with pdfjam . . . . .	37
Summary . . . . .	37

# Introduction

Welcome to the world of dynamically built PDF documents!

In today's web applications, the Portable Document Format is commonly used to send invoices, reports, tickets, e-books, printed materials and many other files. Chances are that you are already working in a project that requires processing PDF files for users.

I have over ten years of experience with the PDF and now I'm going to share everything, so that you can save time, avoid common pitfalls and learn more!



This book is a work-in-progress and I'll be happy to hear your opinions!

## Who is this book for?

If you're reading this, you're most likely a PHP developer looking for help with processing PDF files. You probably need to dynamically create documents and you don't know where to start. Perhaps you also need to transform existing documents to text or images, or do any other sort of editing.

You might already have some experience with PDF processing, but you're searching for a more in-depth explanation. You're struggling with different tools and you're not sure which one is perfect for the job.

You should have basic programming skills with [PHP](https://www.php.net/)<sup>1</sup>. Experience with using [Composer](https://getcomposer.org/)<sup>2</sup> is highly recommended as this is the primary tool to load external libraries into your project. Also, basic knowledge of the Unix shell will be useful.

This book is not tied to any particular framework, however we might use some components coming from popular frameworks. Users of all frameworks are welcome here, just like people who code plain PHP.

## How to read this book?

[Chapters 2](#) and [3](#) provide explanations about the Portable Document Format itself. First you'll learn about the features offered by subsequent versions of the format. You'll know the differences between

---

<sup>1</sup><https://www.php.net/>

<sup>2</sup><https://getcomposer.org/>

PDF versions. This will let you speak more freely with your business clients about possibilities of the PDF. You'll also know what to expect from the tools you will be using.

[Chapter 3](#) is an in-depth look into the **internal structure of a PDF file**. For most basic tasks you won't need to know it. However, some advanced processes require basic understanding of how a PDF is made. You will see why it can be so difficult to extract meaningful data from a PDF by a program.

Later, we will look at different tools to generate PDF documents. You'll know the pros and cons of several solutions, both native libraries and binary tools. You can **treat chapter 5 as a cheat sheet** to quickly choose the best tool for your needs, both feature and performance-wise.

In [chapter 6](#) you will learn about different ways to share a document with a user. Reading inside a browser window, downloading, emailing, cloud uploads - every business case may require something else. Know your options.

I'm also going to describe advanced topics like CSS Paged Media and Generated Content. These specifications allow you to create professional paged documents and books with a simple HTML and CSS syntax.

This book ends with several guides about processing existing documents. You'll learn how to extract text, metadata and other objects from a PDF. You'll also perform splitting, joining, watermarking and other editing tasks.

## About the author

I'm a software developer mostly with a PHP and Java background. I started my professional career in 2008, just before a global economic crisis hit. I survived and then learnt a lot about different industries I worked for: media, advertising, education and finance.

I've been sharing my programming knowledge since I was a teenager. Back in 2002 I started my adventure with a Polish e-zine, when dial-up connections were still the main way of browsing the internet. I often stopped my parents from making phone calls when I wanted to surf the Web. Sometimes, they had to pay terrifying bills for my online time.

The internet kept growing rapidly. Unfortunately I had to stay a bit in the back and finish my formal education. Then I took an ordinary job and as a 19-year-old young man, I did everything my bosses wanted me to do.

When I gathered enough professional experience, I decided to share it. I started my [Twitter profile](#)<sup>3</sup> and [my blog](#)<sup>4</sup>. I gave talks at meetups and conferences. I'm still doing it today and I hope you will all benefit from my work!

---

<sup>3</sup><https://twitter.com/peterdevpl>

<sup>4</sup><https://peterdev.pl/>



# What is Portable Document Format for?

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Quick look at the history

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Main features

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Which version to use?

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Differences between HTML and PDF

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Is PDF a right choice for me?

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

# The structure of a PDF file

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Main goals

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Four parts of a file

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Write your own PDF in Notepad

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Header

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Catalog

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Page tree

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Page definition

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Page contents stream

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Cross-reference table

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Trailer

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Compression

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

# An example document: an invoice

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## An example invoice

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Basic HTML structure

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Body structure

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Writing header

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Main section

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Adding footer

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Creating a stylesheet

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Page size and margin

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Body and header styles

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Main contents

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Footer

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## CSS preprocessors for advanced styles

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Validating document syntax

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

# Choosing a tool to generate PDF files

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Two ways to create a PDF file

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Converting other formats to PDF

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Native PHP libraries

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## First steps with FPDF

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## TCPDF

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## mPDF

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## **Dompdf**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## **typeset.sh**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## **External conversion tools**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## **wkhtmltopdf**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## **Headless Chrome**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## **Paged.js and Vivliostyle**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## **WeasyPrint**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## **PrinceXML**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## **LaTeX or PostScript to PDF**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## **Markdown to PDF**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## **ReStructured Text to PDF**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## **Summary**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.



# Presenting a document to a user

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Displaying a PDF in a frame

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Browser window

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Using PHP libraries

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Using command-line tools

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Downloading a PDF

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## More about filenames

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## **Sending a PDF over email**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## **Saving a file**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## **Display preferences**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## **Page layout and mode**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## **Viewer preferences**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## **Open action**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## **Setting TCPDF**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## **Setting mPDF**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Setting Dompdf

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

# Understanding fonts

Most documents are based on text. To create documents you have to understand how fonts work and how to use them properly. Typography is an art itself and I'm sure you love nicely printed books and brochures. Would be nice if you could make such things too, right?

I've engineered dynamic documents used by thousands of teachers, students, parents, customers and so on. I had to adhere to the highest standards possible because those documents communicated very important things: grades, prices, business deals, formalities. The mere text on paper impacted people's lives.

I learned about serif and sans-serif typefaces, about varying the space between different letter combinations, about what makes great typography great. It was beautiful. Historical. Artistically subtle in a way that science can't capture. And I found it fascinating.  
–Steve Jobs

## How fonts work?

To build a piece of text you need characters that will make letters and words. Before we start, there are some terms you should familiarize with.

A *character set* defines mappings between numeric codes and characters: letters, digits, symbols, and so on. For example, in the [ASCII table](https://www.asciitable.com/)<sup>5</sup>, the decimal number 65 represents a Latin letter A. This is an abstract representation; we still don't know how this letter should be drawn on screen or printed.

An *encoding* specifies how the character codes will be represented as bytes. For ANSI this is simple: a byte value 65 (decimal) is equal to ASCII code 65, which represents capital letter A. However, if a character set exceeds 256 possible values of a single byte, we dive into the world of multi-byte encodings. The most popular ones are UTF-8, UTF-16, UTF-32, UCS-2 and UCS-4 for the [Unicode standard](https://unicode.org/).<sup>6</sup>

There are hundreds of thousands of characters in the world, including many alphabets, scripts, and even emojis. Until 1990s, computers used different sets of 256 characters, enough to fit in one byte. This reduced the ability to interchange documents; the same byte sent from an English to a Russian computer would be represented as a different character. The Unicode standard was made as an effort to collect all the characters in the world in one set.

Although Unicode has been used for years now, you still need to tell the conversion tool that you're using Unicode, preferably with the `<meta charset="utf-8">` tag we mentioned in chapter 4. Some tools might still default to old encodings like ISO-8859-1.

---

<sup>5</sup><https://www.asciitable.com/>

<sup>6</sup><https://unicode.org/>

A *font* is a set of *glyphs* - readable characters and other symbols that represent a character set. A font data file contains either bitmaps or vectors that make up all the character shapes.

*Typesetting* is a broad topic. To please the aesthetic feelings of readers, font systems have to provide many features. For example, it's common to use *ligatures*: symbols that represent pairs of letters like ff and fi. Font files can also provide *hinting* to optimize character rendering on computer screens, especially at small font sizes. Another feature is  *Kerning* which defines spacing between pairs of letters, like “V” and “A”, to maintain good visual proportions.

Having these basics described, we can start using fonts and typing text the way we want to!

## Font file formats in the PDF

A PDF file defines which fonts are used in certain parts of the document. To draw a page properly, these fonts have to be either installed in the operating system, built into the PDF reader or provided along with the document.

The most common font file formats are [OpenType](https://en.wikipedia.org/wiki/OpenType)<sup>7</sup>, [TrueType](https://en.wikipedia.org/wiki/TrueType)<sup>8</sup> and [Type 1](https://en.wikipedia.org/wiki/PostScript_fonts#Type_1)<sup>9</sup>. They differ in features and the way of describing shapes. All of them can be used in a PDF document, but it is a font interpreter's job to understand a font file format.

Fonts designed during different times, by designers having their own preferences and capabilities, would use different formats. It may be impossible to convert between font file formats without data loss.

Document generators often embed a font subset in a document. It's not necessary to attach the whole font file that includes thousands of glyphs. A program chooses only a subset that is enough to represent the document contents.

If a font is not provided by the document, it can be substituted by a reader to one of the standard Type 1 fonts, including Times-Roman, Helvetica, Courier and Symbol. You can also explicitly refer to these standard fonts in a document, just like we did in the [chapter 3](#). However, these aren't Unicode fonts. If you don't make sure a Unicode font is embedded in a document, some text contents might be lost on an end user's machine.

## Picking a proper font

To use a custom font, first you have to choose one that covers all characters you need in your document or its part. This should be common sense, but sometimes we (or the client) forgets about it.

---

<sup>7</sup><https://en.wikipedia.org/wiki/OpenType>

<sup>8</sup><https://en.wikipedia.org/wiki/TrueType>

<sup>9</sup>[https://en.wikipedia.org/wiki/PostScript\\_fonts#Type\\_1](https://en.wikipedia.org/wiki/PostScript_fonts#Type_1)

For example if you pick a fancy header font and your language includes non-Latin characters, check if the font contains glyphs for them! Either use a website that allows testing fonts or download the font files and try them in some text editor or graphics program.

Also remember that even if you use English, you might have users from other countries. They won't like seeing their names misspelled. For example, Mr. Lech Wałęsa<sup>10</sup> would be disappointed to see his name printed as Lech Wa??sa or Lech Walesa.



Don't be afraid to say "no" if you're asked to use a font that you know is not capable of displaying real text in a PDF! Designers will use generic examples like "John Smith" to show how beautiful and clever their designs are, but reality often exceeds one's imagination. If the real documents are rendered improperly, everyone will blame you, not the designer!

Type here to preview text

Zażółć gęślą jaźń

Thin 100

Zażółć gęślą jaźń

Thin 100 italic

Zażółć gęślą jaźń

Light 300

Zażółć gęślą jaźń

Checking a Polish font "Lato" ("summer") on Google Fonts. This popular test text was made up to ensure the presence of all Polish diacritics. You might as well verify characters from other alphabets.

As you can see in the example above, one font family can have multiple styles and weights. The shapes for "light" and "bold" or "normal" and "italic" are different. A font is not automatically converted from one version to another; it is up to the font designer to provide proper files that comply to the designer's vision.

Usually, there are no "one-size-fits-all" solutions. Some fonts do not have an "italic" or "bold italic" versions on purpose. Some fonts contain only uppercase letters (capitals). Other fonts, like fancy handwriting-like ones, are not readable in small sizes. One font can be designed explicitly for small

<sup>10</sup>"A Polish statesman, dissident, and Nobel Peace Prize laureate, who served as the first democratically elected president of Poland from 1990 to 1995." (Wikipedia)

text and another can be dedicated for big titles. Moreover, what looks good on a screen might not be aesthetically pleasing on paper.

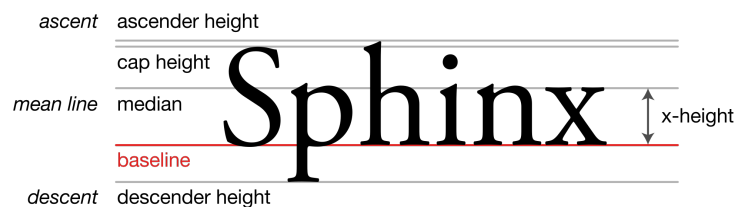


Be careful with using Google Fonts, a popular web font provider. By default, it provides font files compressed with the [WOFF2 format](#)<sup>11</sup> which is not supported by PDF.

Fortunately, Google offers a “Download family” function which gives you a complete ZIP archive with TrueType files. You can also use additional tools like [FontForge](#)<sup>12</sup> to decompress WOFF2 files.

Æ	Ç	È	É	Ê	Ë	Ì	Í
Î	Ï	Ð	Ñ	Ò	Ó	Ô	Õ
Ö	×	Ø	Ù	Ú	Û	Ü	Ý
Þ	ß	à	á	â	ã	ä	å

Browsing glyphs of the DejaVu Sans font using a site called FontDrop!. This file contains over 6 thousand glyphs and covers more than 150 languages.



Letter metrics in a typical font. (Max Naylor/Wikipedia)

## Selecting a font in CSS

Let’s remind ourselves how to pick a font in CSS. The most basic syntax looks like this:

```
1 body {
2   font-family: Verdana, Arial, sans-serif;
3 }
```

The example above means that we prefer the Verdana font, but in case if it’s not available we recommend substituting it either with Arial or any sans-serif font. We depend only on fonts available in a certain system. Every OS has a basic set of fonts, but you can also install your own.

<sup>11</sup><https://www.w3.org/TR/WOFF2/>

<sup>12</sup><https://fontforge.org/>

You might want to use a font from the web in your document without installing it globally in the operating system. In the example below, we import a font file and assign a local name `Lato`. We declare this is a normal (not italic) font of a regular weight:

```
1 @font-face {  
2   font-family: 'Lato';  
3   font-style: normal;  
4   font-weight: normal;  
5   src: url('https://example.org/lato.ttf') format('truetype');  
6 }  
7  
8 body {  
9   font-family: Lato;  
10 }
```



The `font-weight` property accepts either simple `normal` and `bold` values, or one of the numeric weights, from the lightest to the boldest: 100, 200, 300, 400, 500, 600, 700, 800 and 900. A normal weight is considered to be equal to 400, and bold equals to 700. Using numeric weights allow additional options like “extra light” or “black” (super bold).

The `@font-face` syntax works fine with any Chromium-based tools, and also WeasyPrint and Prince. Other tools make selecting a font a bit harder.

## Providing a font to wkhtmltopdf

For security reasons, `wkhtmltopdf` blocks any access to remote font files. It cannot even read a font file from a local drive.

To pick a custom font, we will use a data URL trick. First we have to encode the font file with Base64<sup>13</sup>. We can use either the PHP function `base64_encode()`, the Linux console command `base64` or any Base64 encoder available online.

Then we copy the encoded file contents and paste into the CSS:

---

<sup>13</sup>Base64 is a method of encoding binary 8-bit data with printable characters that use only a 6-bit range. This allows embedding binary contents in emails, or other text media - like the CSS here. A Base64-encoded file is bigger by one-third than the original.



```

1  @font-face {
2      font-family: 'CaslonItalic';
3      src: url(data:font/truetype;charset=utf-8;base64,PASTE_IT_HERE) format("truetype");
4  }
5
6  body {
7      font-family: CaslonItalic;
8  }

```

Because an encoded font file can be very long, it's more convenient to move the @font-face declaration to a separate CSS file and then use @include to attach it to the main stylesheet. You can decide if you want to include that encoded file in your repository, or generate it on-demand in some build script.

## Providing a font to Dompdf

The Dompdf PHP library has its internal font metrics engine which incorporates local caching. The mechanism is cumbersome because you have to manually register the font before using it.

This can be done with a `load_font.php` script which is available in the [dompdf/utls](#)<sup>14</sup> package. Since it would require to copy another repo to the `vendor/dompdf/dompdf` directory, I don't really like this method.

Another way is to extend your PDF rendering code. During the first round, Dompdf will create cache files in the `vendor/dompdf/dompdf/lib/fonts` directory - which means your script must have write access there. Next time, those cached resources will be used to embed the font in a PDF:

```

1  use Dompdf\Dompdf;
2  use Dompdf\Options;
3
4  $fontDirectory = '/home/someuser/fonts';
5
6  $options = new Options();
7  $options->setChroot($fontDirectory);
8
9  $pdf = new Dompdf($options);
10 $pdf->getFontMetrics()->registerFont(
11     ['family' => 'CaslonItalic', 'style' => 'italic', 'weight' => 'normal'],
12     $fontDirectory . '/CaslonItalic.ttf'
13 );
14 $pdf->loadHtml($html);
15 $pdf->render();
16 file_put_contents('output.pdf', $pdf->output());

```

---

<sup>14</sup><https://github.com/dompdf/utls>

The `setChroot()` call is necessary for security purposes, so that Dompdf won't access any system files.

Note that when adding a font file you must specify its corresponding style and weight.

## Setting a custom font in mPDF

mPDF has a decent documentation which explains a lot of nuances related to [international font handling](https://mpdf.github.io/fonts-languages/fonts-in-mpdf-7-x.html).<sup>15</sup>

To use your own font you have to register it. There is one major drawback: you have to invent a font family name that's all lowercase and without any spaces nor other special characters. So instead of `font-family: 'DejaVu Sans'` you have to enter `font-family: dejavusans`.

You can register as many font directories as you need. Moreover, you'll need a temporary directory to store font cache. By default it's `vendor/mpdf/mpdf/tmp/mpdf/ttfontdata` and the script must have write permissions for that. Fortunately you can set another cache path:

```
1  use Mpdf\Config\ConfigVariables;
2  use Mpdf\Config\FontVariables;
3  use Mpdf\Mpdf;
4
5  $fontDirectory = '/home/someuser/fonts';
6
7  $defaultConfig = (new ConfigVariables())->getDefaults();
8  $fontDirs = $defaultConfig['fontDir'];
9
10 $defaultFontConfig = (new FontVariables())->getDefaults();
11 $fontData = $defaultFontConfig['fontdata'];
12
13 $mpdf = new Mpdf([
14     'fontDir' => \array_merge($fontDirs, [
15         $fontDirectory,
16     ]),
17     'fontdata' => $fontData + [
18         'caslon' => [
19             'I' => 'CaslonItalic.ttf',
20         ],
21     ],
22     'tempDir' => $fontDirectory . '/tmp',
23 ]);
24 $mpdf->WriteHTML($html);
25 $mpdf->Output('output.pdf', 'F');
```

---

<sup>15</sup><https://mpdf.github.io/fonts-languages/fonts-in-mpdf-7-x.html>

When registering font files, you have to declare their style with R, B, I and BI identifiers, corresponding to “regular”, “bold”, “italic” and “bold italic” styles, respectively.

## Custom fonts in TCPDF

TCPDF follows a similar font registration pattern to the previous two libraries. You can do it in two ways - either in the command line, or directly in PHP code.

Thanks to the command line you can embed the conversion commands in some Continuous Delivery pipeline that builds your application. Instead of committing the temporary font files, you can rebuild them every time with a simple command like this (assuming you’re using Composer):

```
1 php ./vendor/tecnickcom/tcpdf/tools/tcpdf_addfont.php -b -f 32 -o /home/someuser/fon\
2 ts/tmp/ -i CaslonItalic.ttf
```

If you don’t use the command line, you can still do the same conversion thing in PHP using the TCPDF\_FONTS class:

```
1 $fontDirectory = '/home/someuser/fonts/';
2
3 // The trailing slash is mandatory here
4 $tempDirectory = $fontDirectory . 'tmp/';
5
6 $fontname = TCPDF_FONTS::addTTFfont($fontDirectory . 'CaslonItalic.ttf', 'TrueTypeUn\
7 icode', '', 32, $tempDirectory);
8
9 $pdf = new TCPDF('P', 'mm', 'LETTER');
10 $pdf->AddPage();
11 $pdf->AddFont($fontname, 'I', $tempDirectory . $fontname . '.php');
12 $pdf->writeHTML($html);
13 file_put_contents('output.pdf', $pdf->Output('', 'S'));
```

The `addTTFfont()` method parses the original font file and creates three temporary files in the directory of your choice. Obviously, the script must have write access to that path. The return value holds a font file name which is usually a lowercase string. With `AddFont()` method you register the PHP font definition file created earlier.

Now you can use the font inside the document like this (remember about the lowercase font family name):

```
1 body {  
2   font-family: 'caslon';  
3   font-size: 72pt;  
4   font-style: italic;  
5 }
```

Instead of using CSS, you can also set the current font with PHP:

```
1 $pdf->SetFont($fontname, 'I', 72);
```

The mysterious number 32 which appears both in the command line call and the `addTTFfont()` method is the *font descriptor flag* from the PDF specification. Fixed and italic fonts are usually autodetected, but for other types you have to specify an exact flag value:

Font descriptor flag	Meaning
1	fixed font
4	symbol font
8	script (handwriting)
32	non-symbol (standard) font
64	italic font
65,536	all caps (no lowercase letters)
131,072	small caps



TCPDF does not support OpenType nor WOFF2 fonts.

## Text transformations

Although some transformations like italics or bold cannot be done automatically, there are plenty of other operations we can perform on text. We'll see what is possible through CSS:

Lorem ipsum dolor sit amet

Lorem ipsum dolor sit amet

~~Lorem ipsum dolor sit amet~~

Lorem ipsum dolor sit amet

**Lorem ipsum dolor sit amet**

Four variants of the text-decoration property: underline, overline, line-through, and a special one: underline with a wave. The fifth example uses a text-shadow property.

L o r e m   i p s u m   d o l o r   s i t   a m e t

Lorem   ipsum   dolor   sit   amet

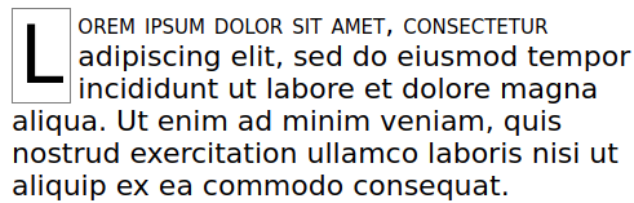
Adjusting horizontal letter and word spacing. First example has letter-spacing set to 10px. The other one has word-spacing equal to 63px.

Lorem ipsum dolor sit amet,  
consectetur adipiscing elit, sed do  
eiusmod tempor incididunt ut  
labore et dolore magna aliqua. Ut  
enim ad minim veniam, quis  
nostrud exercitation ullamco laboris  
nisi ut aliquip ex ea commodo  
consequat. Duis aute irure dolor in  
reprehenderit in voluptate velit  
esse cillum dolore eu fugiat nulla  
pariatur. Excepteur sint occaecat  
cupidatat non proident, sunt in  
culpa qui officia deserunt mollit  
anim id est laborum.

Line height adjusted to 0.6em. This way we're overriding the default line spacing for a font.

Lorem ipsum dolor sit amet

Text stretched vertically to 400% of the original font's height.



Example tricks with the paragraph's style. The first line of text is in small caps. Also, the first letter floats on the left, has a bigger size and a border.

Below there is a CSS code for the example above:

```
1 p {  
2   width: 500px;  
3 }  
4  
5 p::first-line {  
6   font-variant: small-caps;  
7 }  
8  
9 p::first-letter {  
10  border: solid gray 1px;  
11  float: left;  
12  font-size: 2.8em;  
13  margin-right: 5px;  
14  padding: 0 5px;  
15 }
```

Note that the HTML to PDF converters might not interpret all these CSS rules properly.

You can find more examples of styling text with CSS on the [Mozilla Developer's Network](https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text/Fundamentals)<sup>16</sup> and on the [CSS-Tricks](https://css-tricks.com/)<sup>17</sup> site.

## ANSI or Mac versus Unicode

If you create a standard plain text file in Notepad, it is going to use a single-byte ANSI encoding (Windows) or a Mac OS Roman encoding (classic Mac OS). If you only use Latin characters, you will be fine - but the problems arise with languages that contain letters outside the Latin alphabet.

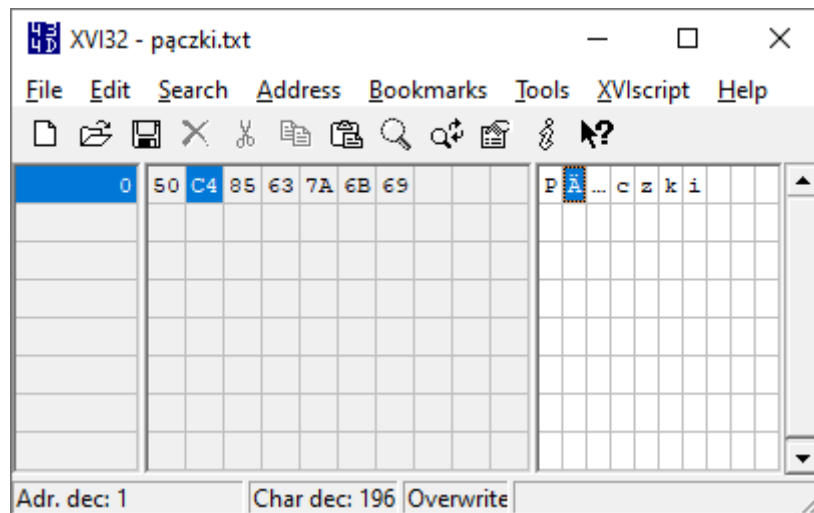
Because it is impossible to squeeze all the world's alphabets into a set of one-byte values, operating systems introduced multiple *code pages* (or *charsets*) based on the language setting. **The same code in a different code page will represent a different character.** The default page on Windows for English is CP-1252 (similar to ISO-8859-1), but for Polish there is a separate page called CP-1250

<sup>16</sup>[https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling\\_text/Fundamentals](https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text/Fundamentals)

<sup>17</sup><https://css-tricks.com/>

with different characters in it. Some old Polish websites were encoded in the ISO-8859-2 standard which is again different from all previous ones. Madness!

This is why Unicode was invented in the early 1990s, but it took many years to adopt it widely. **Today all modern websites use Unicode** and the most popular encoding is UTF-8. Characters are organized into *planes* and *blocks*. In UTF-8, the basic ASCII set is still encoded using single bytes, while codes above 255 are represented with two to four bytes.



The word “Pączki” (pronounced “ponchkee”) encoded as UTF-8. Standard ASCII characters are encoded with single bytes. The letter “ą” (small “a” with “ogonek”) has a decimal number 261 in Unicode and is saved as 0xC485 hexadecimally.

In an HTML document, the best way to ensure that the UTF-8 encoding will be understood is to add the `<meta>` tag:

```

1 <html>
2   <head>
3     <meta charset="utf-8">
4   </head>
5   <body>
6     <p>Zażółć gęślą jaźń!</p>
7   </body>
8 </html>

```



When joining texts from multiple files, be careful of the Byte Order Mark that can appear in the beginning of a text file. It can be used to indicate *endianness*, i.e. the order of bytes in multi-byte numbers. If you start seeing weird characters in the middle of your document, it might mean that a BOM wasn't interpreted properly somewhere in your processing chain. The easiest solution is to have UTF-8 in all source files and remove any BOMs.



Basic PDF fonts do not support Unicode. If you try to put some unusual characters in your document, you might end up with question marks, rectangles or other unexpected stuff. You need a proper font to represent your document contents.

## Summary

In this chapter you have learned the difference between character set, encoding and a font. You know how to pick a font and use it in an HTML document which might be later converted to PDF. You know how to ensure proper encoding of characters from different languages.

You've also learned that some conversion tools may have limited font reading capabilities. Instead of simple `@font-face` syntax, they expect additional scripting work to register font files.



# Metadata

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## How it looks inside

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Entering metadata with TCPDF

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## mPDF example

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Dompdf example

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## WeasyPrint example

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

# Encryption

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Choosing the cipher

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## User permissions

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Encryption with TCPDF

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Encryption with mPDF

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Encryption with Dompdf

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Encryption with FPDF

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Encrypting an existing file with command line tools

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

# Calling external programs from a PHP script

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Input, output and exit codes

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Basic execution from a PHP script

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Escaping arguments

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Opening and controlling a process

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## The most convenient solution: The Process Component

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Asynchronous and background processes

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Timeouts

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Reporting progress of time-consuming tasks

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Queueing tasks

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

# Styling pages with CSS

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Page dimensions and margins

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Adjusting individual pages

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Page-margin boxes and generated content

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Bleed area and crop marks

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Controlling page breaks

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Orphans and widows

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Forced page breaks

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Media queries

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

# Reading and testing PDF files

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Metadata and text extraction

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

### PdfParser

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

### Poppler

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Taking a screenshot

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

### Using ImageMagick

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

### Using GhostScript

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.



## Using Poppler

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Using Inkscape

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

# Editing PDF files

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Importing a document to FPDF or TCPDF

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## The PDF Toolkit

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Joining smaller documents into one

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Splitting a document into individual pages

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Applying a watermark or stamp

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Attachments

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Encryption

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Adding or changing metadata

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Decompressing a PDF

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Manipulating files with pdfjam

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.

## Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/mastering-pdf-with-php>.