# Dompdf

[Build Status](https://github.com/dompdf/dompdf/actions/workflows/test.yml) [PHP Versions Supported](https://packagist.org/packages/dompdf/dompdf) [Latest Release](https://packagist.org/packages/dompdf/dompdf) [Total Downloads](https://packagist.org/packages/dompdf/dompdf) [License](https://packagist.org/packages/dompdf/dompdf)

**Dompdf is an HTML to PDF converter**

At its heart, dompdf is (mostly) a [CSS 2.1](http://www.w3.org/TR/CSS2/) compliant HTML layout and rendering engine written in PHP. It is a style-driven renderer: it will download and read external stylesheets, inline style tags, and the style attributes of individual HTML elements. It also supports most presentational HTML attributes.

*This document applies to the latest stable code which may not reflect the current release. For released code please* [*navigate to the appropriate tag*](https://github.com/dompdf/dompdf/tags)*.*

**Check out the** [**demo**](http://eclecticgeek.com/dompdf/debug.php) **and ask any question on** [**StackOverflow**](https://stackoverflow.com/questions/tagged/dompdf) **or in** [**Discussions**](https://github.com/dompdf/dompdf/discussions)**.**

Follow us on [Twitter](http://www.twitter.com/dompdf).

## Features

* Handles most CSS 2.1 and a few CSS3 properties, including @import, @media & @page rules
* Supports most presentational HTML 4.0 attributes
* Supports external stylesheets, either local or through http/ftp (via fopen-wrappers)
* Supports complex tables, including row & column spans, separate & collapsed border models, individual cell styling
* Image support (gif, png (8, 24 and 32 bit with alpha channel), bmp & jpeg)
* No dependencies on external PDF libraries, thanks to the R&OS PDF class
* Inline PHP support
* Basic SVG support (see "Limitations" below)

## Requirements

* PHP version 7.1 or higher
* DOM extension
* MBString extension
* php-font-lib
* php-svg-lib

Note that some required dependencies may have further dependencies (notably php-svg-lib requires sabberworm/php-css-parser).

### Recommendations

* GD (for image processing)
  + Additionally, the IMagick or GMagick extension improves image processing performance for certain image types
* OPcache (OPcache, XCache, APC, etc.): improves performance

Visit the wiki for more information: <https://github.com/dompdf/dompdf/wiki/Requirements>

## About Fonts & Character Encoding

PDF documents internally support the following fonts: Helvetica, Times-Roman, Courier, Zapf-Dingbats, & Symbol. These fonts only support Windows ANSI encoding. In order for a PDF to display characters that are not available in Windows ANSI, you must supply an external font. Dompdf will embed any referenced font in the PDF so long as it has been pre-loaded or is accessible to dompdf and reference in CSS @font-face rules. See the [font overview](https://github.com/dompdf/dompdf/wiki/About-Fonts-and-Character-Encoding) for more information on how to use fonts.

The [DejaVu TrueType fonts](https://dejavu-fonts.github.io/) have been pre-installed to give dompdf decent Unicode character coverage by default. To use the DejaVu fonts reference the font in your stylesheet, e.g. body { font-family: DejaVu Sans; } (for DejaVu Sans). The following DejaVu 2.34 fonts are available: DejaVu Sans, DejaVu Serif, and DejaVu Sans Mono.

## Easy Installation

### Install with composer

To install with [Composer](https://getcomposer.org/), simply require the latest version of this package.

composer require dompdf/dompdf

Make sure that the autoload file from Composer is loaded.

// somewhere early in your project's loading, require the Composer autoloader

// see: http://getcomposer.org/doc/00-intro.md

require 'vendor/autoload.php';

### Download and install

Download a packaged archive of dompdf and extract it into the directory where dompdf will reside

* You can download stable copies of dompdf from <https://github.com/dompdf/dompdf/releases>
* Or download a nightly (the latest, unreleased code) from <http://eclecticgeek.com/dompdf>

Use the packaged release autoloader to load dompdf, libraries, and helper functions in your PHP:

// include autoloader

require\_once 'dompdf/autoload.inc.php';

Note: packaged releases are named according using semantic versioning (*dompdf\_MAJOR-MINOR-PATCH.zip*). So the 1.0.0 release would be dompdf\_1-0-0.zip. This is the only download that includes the autoloader for Dompdf and all its dependencies.

### Install with git

From the command line, switch to the directory where dompdf will reside and run the following commands:

git clone https://github.com/dompdf/dompdf.git

cd dompdf/lib

git clone https://github.com/PhenX/php-font-lib.git php-font-lib

cd php-font-lib

git checkout 0.5.1

cd ..

git clone https://github.com/PhenX/php-svg-lib.git php-svg-lib

cd php-svg-lib

git checkout v0.3.2

cd ..

git clone https://github.com/sabberworm/PHP-CSS-Parser.git php-css-parser

cd php-css-parser

git checkout 8.1.0

Require dompdf and it's dependencies in your PHP. For details see the [autoloader in the utils project](https://github.com/dompdf/utils/blob/master/autoload.inc.php).

## Framework Integration

* For Symfony: [nucleos/dompdf-bundle](https://github.com/nucleos/NucleosDompdfBundle)
* For Laravel: [barryvdh/laravel-dompdf](https://github.com/barryvdh/laravel-dompdf)
* For Redaxo: [PdfOut](https://github.com/FriendsOfREDAXO/pdfout)

## Quick Start

Just pass your HTML in to dompdf and stream the output:

// reference the Dompdf namespace

use Dompdf\Dompdf;

// instantiate and use the dompdf class

$dompdf = new Dompdf();

$dompdf->loadHtml('hello world');

// (Optional) Setup the paper size and orientation

$dompdf->setPaper('A4', 'landscape');

// Render the HTML as PDF

$dompdf->render();

// Output the generated PDF to Browser

$dompdf->stream();

### Setting Options

Set options during dompdf instantiation:

use Dompdf\Dompdf;

use Dompdf\Options;

$options = new Options();

$options->set('defaultFont', 'Courier');

$dompdf = new Dompdf($options);

or at run time

use Dompdf\Dompdf;

$dompdf = new Dompdf();

$options = $dompdf->getOptions();

$options->setDefaultFont('Courier');

$dompdf->setOptions($options);

See [Dompdf\Options](http://src/Options.php) for a list of available options.

### Resource Reference Requirements

In order to protect potentially sensitive information Dompdf imposes restrictions on files referenced from the local file system or the web.

Files accessed through web-based protocols have the following requirements:

* The Dompdf option "isRemoteEnabled" must be set to "true"
* PHP must either have the curl extension enabled or the allow\_url\_fopen setting set to true

Files accessed through the local file system have the following requirement:

* The file must fall within the path(s) specified for the Dompdf "chroot" option

## Limitations (Known Issues)

* Table cells are not pageable, meaning a table row must fit on a single page: See <https://github.com/dompdf/dompdf/issues/98>
* Elements are rendered on the active page when they are parsed.
* Embedding "raw" SVG's (<svg><path...></svg>) isn't working yet: See <https://github.com/dompdf/dompdf/issues/320>  
  Workaround: Either link to an external SVG file, or use a DataURI like this:

$html = '<img src="data:image/svg+xml;base64,' . base64\_encode($svg) . '">';

* Does not support CSS flexbox: See <https://github.com/dompdf/dompdf/issues/971>
* Does not support CSS Grid: See <https://github.com/dompdf/dompdf/issues/2988>
* A single Dompdf instance should not be used to render more than one HTML document because persisted parsing and rendering artifacts can impact future renders.

[Donate button](http://goo.gl/DSvWf)

*If you find this project useful, please consider making a donation. Any funds donated will be used to help further development on this project.)*