



WeasyPrint converts HTML/CSS documents to PDF

WeasyPrint is a visual rendering engine for HTML and CSS that can export to PDF. It aims to support web standards for printing. WeasyPrint is free software made available under a BSD license.

It is based on various libraries but *not* on a full rendering engine like WebKit or Gecko. The CSS layout engine is written in Python, designed for pagination, and meant to be easy to hack on.

- Free software: BSD licensed
- Python 2.7 or 3.4+
- Source code and issue tracker: [on Github](#)
- [Documentation](#)
 - [Install](#)
 - [Features](#)
- [Get in touch](#)

Sample output

As an example, here is the [introduction chapter](#) of the CSS 2.1 spec rendered with WeasyPrint: [CSS21-intro.pdf](#). It was obtained by running:

```
weasyprint http://www.w3.org/TR/CSS21/intro.html CSS21-intro.pdf -s
http://weasyprint.org/samples/CSS21-print.css
```

Here is an extract of [CSS21-print.css](#):

```
@page {  
  margin: 3cm 2cm; padding-left: 1.5cm;  
  @top-center {  
    content: "Introduction to CSS 2.1";  
    vertical-align: bottom;  
    border-bottom: 0.5pt solid }  
  @bottom-right {  
    content: "Page " counter(page)  
      " of " counter(pages) }  
  @left-top {  
    content: "W3C Recommendation";  
    background: #005a9c;  
    color: #fff;  
    text-align: right;  
    transform-origin: 100% 0;  
    transform: rotate(-90deg) }  
body { text-align: justify }  
h1 { bookmark-level: none }
```

Acid2

Starting with version 0.11, WeasyPrint passes the [Acid2](#) test. This test was published by the [Web Standards Project](#) as a challenge for web browser to get better support for the HTML, CSS and PNG standards. It is made of many small elements with various positioning techniques. When all the layout rules are implemented correctly, they should add up to the smiling face shown on the right.



Have a look at WeasyPrint's output [in PDF](#) and [in PNG](#). The PNG is pixel-perfect, but with PDF you may see faint lines or a [moiré](#) pattern inside the smiling head. This is because the test is made for pixel-based screens with many layers of different colors that overlap exactly. Only the top layers are supposed to be visible but due to the way that most PDF viewers use anti-aliasing, the bottom layers may bleed on the sides.