pdfHTML

an iText 7 add-on

pdfHTML: Convert HMTL/ CSS to PDF

What is pdfHTML?

pdfHTML is an add-on for the iText 7 platform that transforms HTML and accompanying CSS into a PDF document. pdfHTML allows you to automate PDF generation for documents such as internal reports, tickets, invoices, and more.

Using pdfHTML, you can create beautiful and functional PDFs without having to learn complex PDF syntax or the intricacies of the iText 7 platform. Any designer with HTML and CSS skills can create the template that will be used as the starting point for the conversion by the pdfHTML add-on. This will then generate a visually equivalent PDF document, with all semantic, structural and accessibility information preserved (if required).

Why pdfHTML?

Technologies such as HTML and CSS have become commonplace standards for web page creation. Meanwhile, many companies rely on the PDF format to manage their various document workflows, internal documentation and archiving.
PDF has many great benefits such as reliable formatting across devices and portability, but the format is not easily editable.

Many developers and designers choose to edit documents in HTML and then convert their final version into PDF. This process is painstaking for most people and is why we have developed pdfHTML to help.
pdfHTML, an add-on to iText 7, leverages the widespread knowledge of the HTML format and existing skills of development resources in converting HTML to PDF.

Similarities and differences between HMTL/CSS and PDF

In HTML, content is wrapped in HTML tags. Each tag corresponds to a conceptual element (e.g. paragraph, table), and many tags can be nested. Styling and visual representation for HTML content is provided by the use of Cascading Style Sheets (CSS) declarations. A web browser parses and interprets the HTML file and accompanying CSS to create a visual representation of the content, calculating the rendering and layout on the fly and visualizing the various contents according to their CSS declarations and the renderer's own settings.

A PDF document is not inherently structured and semantic. Its content consists of a set of instructions that result in painting objects at absolute positions on a large canvas. PDF does offer an additional layer of functionality to store semantic and structural information.
To support features like proper content extraction, repurposing of content, search indexing and accessibility, it's crucial to augment the visual-only representation of PDF documents with such additional information.

Since HTML documents inherently contain semantic and structural information too, they are an excellent source to convert to rich, smart PDF documents. This is where pdfHTML can help.

How does pdfHTML work?

On a conceptual level, pdfHTML maps HTML tags to iText 7 layout objects, and CSS property declarations to iText layout properties. On a practical level, this process happens through the use of TagWorkers (classes responsible for processing an HTML tag) and CssAppliers (classes responsible for the processing of CSS styles and any declarations for a HTML tag). Each HTML tag is mapped to a TagWorker and CssApplier, and those classes contain the necessary logic to process the tag, selecting the iText layout object it corresponds to and applying any necessary CSS.

When processing the HTML DOM, pdfHTML walks through the tree in a depth-first manner, starting the translation on a tag, and then recursively processing all its children, ending the processing when all its children have been processed.

TagWorker and CssApplier explained

Mapping HTML tags to iText 7 layout objects, and CSS property declarations to iText layout properties, is a process which takes place through the use of TagWorkers and CssAppliers. pdfHTML comes with default implementations, which can be customized and extended. This allows custom processing according to your business logic and processing of custom tags and properties to fit your workflow.

TagWorker

A TagWorker is a class responsible for processing an HTML tag. This processing includes:

- Resolving any resources required by the tag and its content;
- Translating the tags' content into an iText layout element;
- Resolving any (non-style and style) attributes through a CssApplier.

CssApplier

A CssApplier is a class responsible for the processing of CSS styles and any declarations for an HTML tag. The implementation of this CssApplier contains all the necessary logic to resolve and apply the style declarations to the iText layout element that is associated with the CssApplier.

In practice

A basic example will show the use of pdfHTML. For this, we will use the following HTML and CSS.

```
<html>
    <head>
        <link rel="stylesheet"
        type="text/css"
        href="simple.css"/>
    </head>
    <body>
        <p>iText pdfHTML</p>
        <div>Converting
        HTML to PDF with
        (Cascading) Style
        (Sheets)!
        </div>
    </body>
</html>
```

HTML code

```
p{
    font-style:italic;
}
div{
    color:red;
    border-style:solid;
    border-width:2pt;
    border-color:blue;
}
```

Simple css

The output will be directly written to a PDF file, using the following code:

```
ConverterProperties converterProperties = new ConverterProperties();
HtmlConverter.convertToPdf(new FileInputStream(htmlSource),
new FileOutputStream(pdfDest), converterProperties);
```

The resulting PDF will look as follows:

iText pdfHTML

Converting HTML to PDF with (Cascading) Style (Sheets)!

Figure 3: pdfHTML output: pdfHTML 6

Comparison to XMLWorker

Compared to iText's previous HTML/CSS conversion, XML Worker in combination with iText 5, pdfHTML shows several advantages:

- The pdfHTML add-on makes full use of the enhanced and new capabilities of iText 7.
- pdfHTML supports more HTML tags and CSS features such as floating and fixed positioning, and @media rules and queries.
- It's also easier to extend for custom tags.
- It integrates seamlessly with other iText functionalities such as barcodes, PDF/A and PDF/UA output, and advanced typography features of pdfCalligraph.
- pdfHTML has more robust handling of imperfect or invalid HTML input.

Conclusion

Today HTML/CSS users need to be able to share their documents and files in the wider document workflow, which is often PDF based. This is now possible without our HTML/CSS users acquiring an in depth knowledge of PDF syntax. With the iText 7 Platform, assisted by the pdfHTML add-on, this process of HTML/CSS conversion to PDF can be easily automated, saving our designers headache, time and money.