Creating PDFs using HTML

Using modern Web technologies

François Fernandès
Senior Solution Architect
@tellme_francois
github.com/fernanfs
digital frontiers

François Fernandès

Senior Solution Architect

francois.fernandes@digitalfrontiers.de
@tellme_francois
github.com/fernanfs

digital frontiers

François Fernandès

Member of the Board

francois.fernandes@pdfa.org

PDF association

Why did we need such a tooling?

- Profile documents were created using Google Docs
- A nice looking layout was not the highest priority for some very skilled technical colleagues
- Updating took relatively long

manual … able
layouting … PDFs

I'm dying inside


QUALIFIKATIONSPROFIL
FRANÇOIS FERNANDES
SENIOR SOLUTION ARCHITECT

Einsatzschwerpunkte
Consulting
Mentoring
Analyse
Programmieren
Agile Vorgehensweisen
Cloud Konzepte
Big Data
Verteilte Systeme
IT Automatisierung
Software Architektur
Enterprise Architektur

Sprachkenntnisse
Deutsch (Muttersprache)
Englisch fließend in Wort und Schrift
Französisch fließend

Schwerpunkte und Lebenslauf

Die Tätigkeitsschwerpunkte von Herrn Fernandès liegen in der Konzeption von modernen Cloud-fähigen Unternehmensanwendungen und in der Beratung für die digitale Transformation.

Besondere Erfahrungen in:

- Architektur leichtgewichtiger, verteilter Systeme
- Technologien, die diese Systeme unterstützen
- Dokumentenmanagement und Enterprise Content Management
- Mentoring von Menschen in der Softwareentwicklung

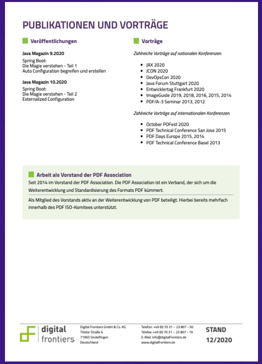
Herr Fernandès ist Senior Solution Architect bei der Digital Frontiers GmbH & Co. KG.

Stationen seines Lebenslaufs davor:

- 14 Jahre levigo solutions GmbH (zuletzt Document Technology Evangelist)

Herr Fernandès hat Wirtschaftsinformatik an der Fachhochschule Reutlingen studiert.

digital frontiers
Digital Frontiers GmbH & Co. KG
Tilsiter Straße 6
71065 Sindelfingen
Deutschland
Telefon: +49 (0) 70 31 – 23 807 – 50
Telefax: +49 (0) 70 31 – 23 807 – 19
E-Mail: info@digitalfrontiers.de
www.digitalfrontiers.de
STAND 12/2020



Objectives for a solution

- Document generation is fast and reliable
- Editing the content doesn’t interfere with the layout
- Live preview of any changes to the content as well as the layout

Why choosing HTML?

- HTML has become the de-facto standard in document markup
- The whole ecosystem around HTML is evolving rapidly
- It is relatively easy to read and write (especially for technical people)
- Editing support is widely available

No, LaTeX is not a solution!
(at least for me)

Toolchain

- Consists of commonly used and proven tools
- yarn
  to manage dependencies and execution
- sass
  for better CSS
- gulp
  to orchestrate the build
- Puppeteer
  controlling Chrome/Chromium instances
- Chromium/Google Chrome
  used for the rendering
- Browsersync
  live reload during design and development

Document Generation

yarn

Gulp

`yarn generate-pdf`

Sass

generate CSS

Puppeteer

generate the PDF document

Live Preview

yarn

Gulp

`yarn serve`

watch for changes

Sass

generate CSS

Browsersync

reload changes in Browser

Generating the PDF Document

```
const browser = await p.launch();
const page = await browser.newPage();
await page.goto(
    "file:" + process.cwd() + "/src/document.html",
    {
      waitUntil: 'networkidle0',
      timeout: 0
    });
await page.pdf({
  format: "a4",
  preferCSSPageSize: true,
  displayHeaderFooter: false,
  path: process.cwd() + "/document.pdf"
});
await browser.close();
```

1. Launch the Browser

Generating the PDF Document

```
const browser = await p.launch();
const page = await browser.newPage();
await page.goto(
    "file:" + process.cwd() + "/src/document.html",
    {
      waitUntil: 'networkidle0',
      timeout: 0
    });
await page.pdf({
  format: "a4",
  preferCSSPageSize: true,
  displayHeaderFooter: false,
  path: process.cwd() + "/document.pdf"
});
await browser.close();
```

1. Launch the Browser
2. Open a new Page

Generating the PDF Document

```
const browser = await p.launch();
const page = await browser.newPage();
await page.goto(
    "file:" + process.cwd() + "/src/document.html",
    {
      waitUntil: 'networkidle0',
      timeout: 0
    });
await page.pdf({
  format: "a4",
  preferCSSPageSize: true,
  displayHeaderFooter: false,
  path: process.cwd() + "/document.pdf"
});
await browser.close();
```

1. Launch the Browser
2. Open a new Page
3. Navigate to the local HTML file

Generating the PDF Document

```
const browser = await p.launch();
const page = await browser.newPage();
await page.goto(
    "file:" + process.cwd() + "/src/document.html",
    {
      waitUntil: 'networkidle0',
      timeout: 0
    });
await page.pdf({
  format: "a4",
  preferCSSPageSize: true,
  displayHeaderFooter: false,
  path: process.cwd() + "/document.pdf"
});
await browser.close();
```

1. Launch the Browser
2. Open a new Page
3. Navigate to the local HTML file
4. Generate the PDF document

Generating the PDF Document

```
const browser = await p.launch();
const page = await browser.newPage();
await page.goto(
    "file:" + process.cwd() + "/src/document.html",
    {
      waitUntil: 'networkidle0',
      timeout: 0
    });
await page.pdf({
  format: "a4",
  preferCSSPageSize: true,
  displayHeaderFooter: false,
  path: process.cwd() + "/document.pdf"
});
await browser.close();
```

1. Launch the Browser
2. Open a new Page
3. Navigate to the local HTML file
4. Generate the PDF document
5. Close the Browser

Generating the PDF Document

```
const browser = await p.launch();
const page = await browser.newPage();
await page.goto(
    "file:" + process.cwd() + "/src/document.html",
    {
      waitUntil: 'networkidle0',
      timeout: 0
    });
await page.pdf({
  format: "a4",
  preferCSSPageSize: true,
  displayHeaderFooter: false,
  path: process.cwd() + "/document.pdf"
});
await browser.close();
```

1. Launch the Browser
2. Open a new Page
3. Navigate to the local HTML file
4. Generate the PDF document
5. Close the Browser

Example Article

Section 1

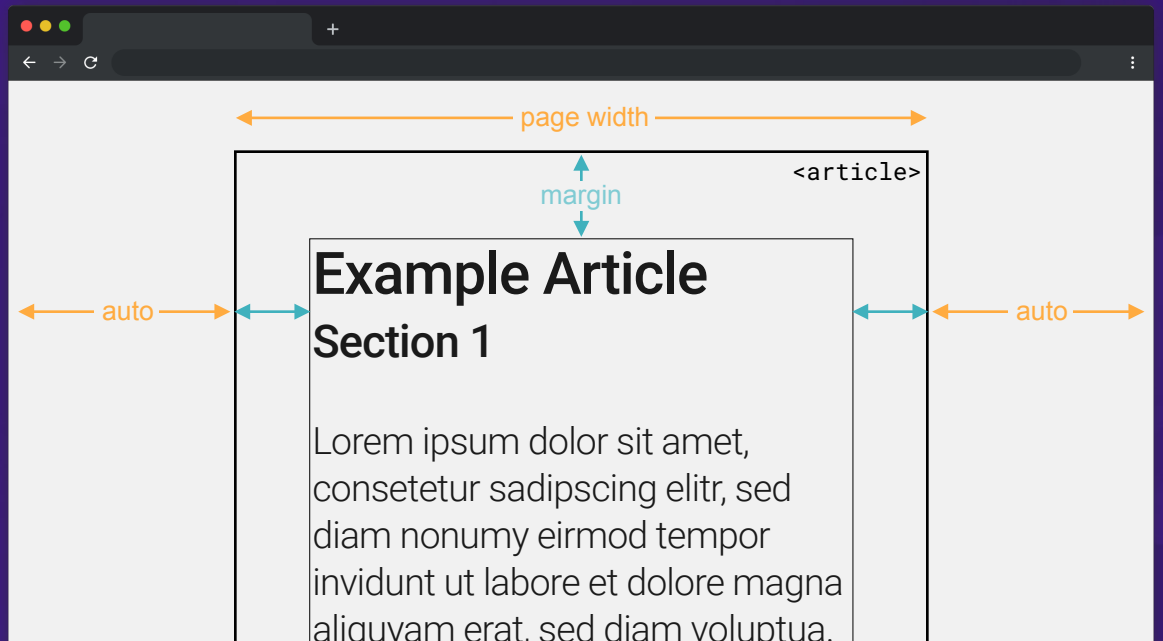
Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et ccusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea

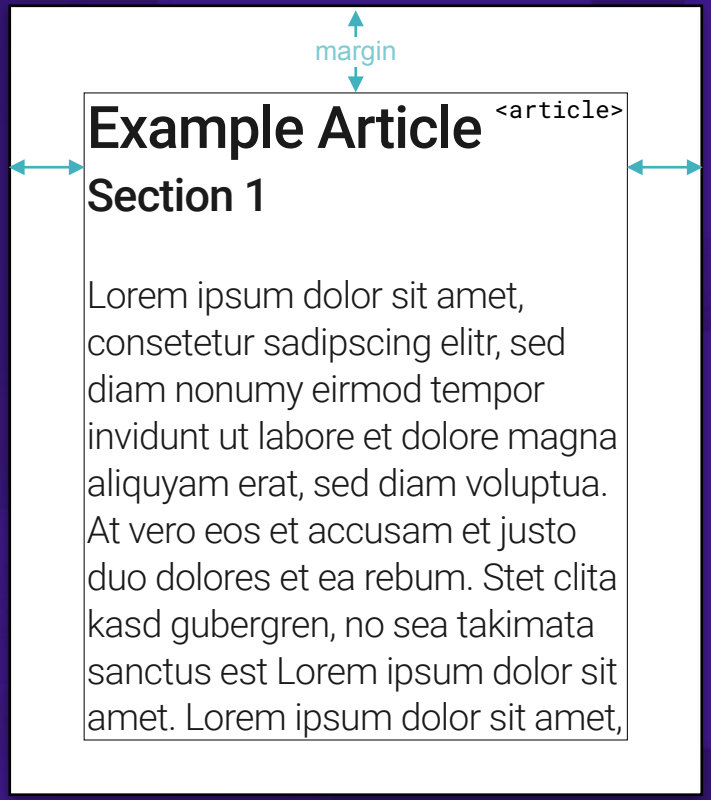
Simulating a page layout in the browser

- layout the content centered in the browser window
- use the page width

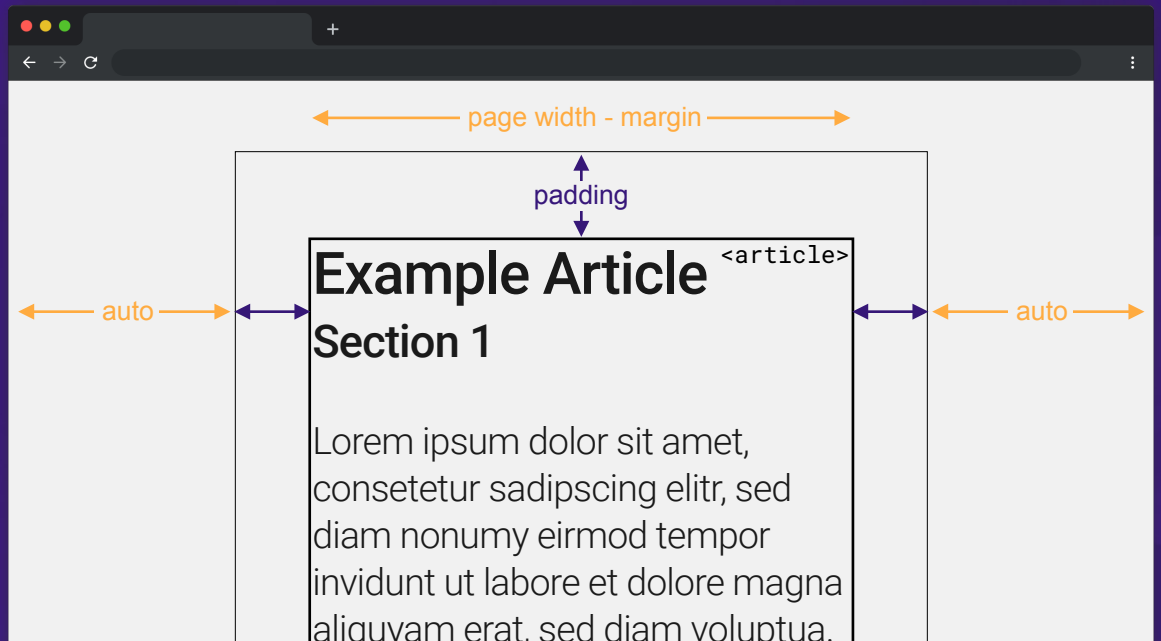
Simulating a page layout in the browser

- layout the content centered in the browser window
- use the page width
- Applying the desired margin


page width
<article>
margin
auto
auto
Example Article
Section 1
Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.


margin
Example Article
<article>
Section 1
Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet,

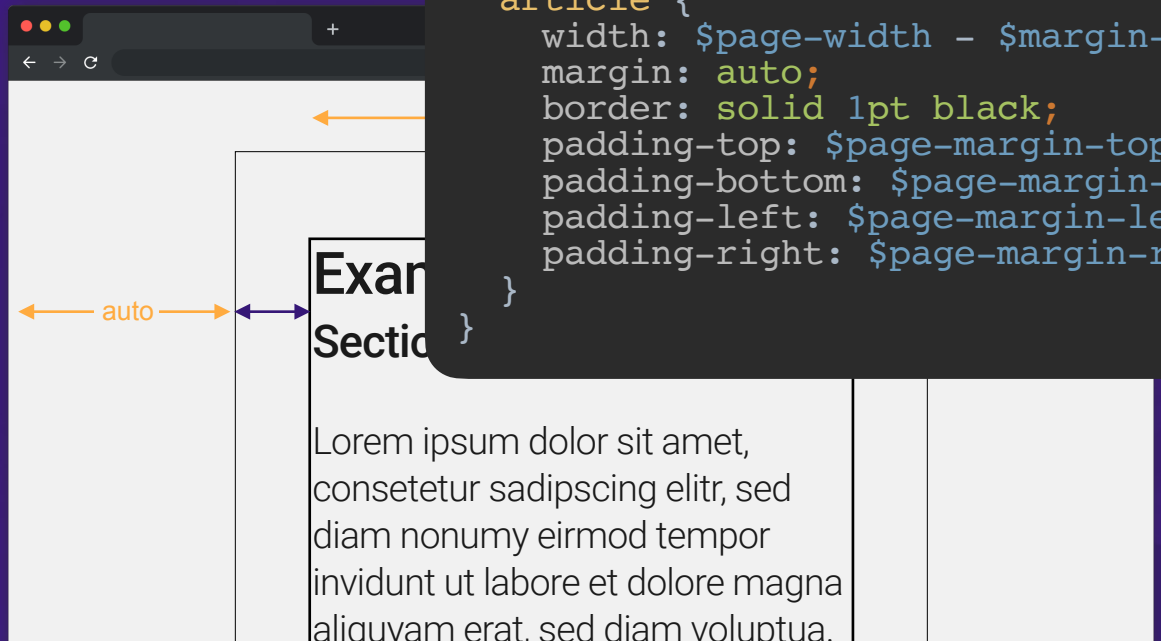
Simulating a page layout in the browser



Simulating a page layout in the browser

```
@page {
  size: $page-size;
  margin: $margin-top $margin-right $margin-bottom $margin-left;
}

@media screen {
  article {
    width: $page-width - $margin-left - $margin-right;
    margin: auto;
    border: solid 1pt black;
    padding-top: $page-margin-top;
    padding-bottom: $page-margin-bottom;
    padding-left: $page-margin-left;
    padding-right: $page-margin-right;
  }
}
```



Enough Talking!
let's see a

<Live Demo>

Limitations

- Table of Contents with page numbers can not be generated
- Header and Footer handling is very limited
- Adding content outside the page content area not possible (this could be resolved by additional element nesting)
- No tagging information is written into the document

Photo by Matt Walsh on Unsplash

What will be your next HTML based PDF?

The sample project can be found on GitHub:

https://github.com/dxfrontiers/html-to-pdf

François Fernandès

Senior Solution Architect

@tellme_francois

github.com/fernanfs

https://blog.digitalfrontiers.de

https://www.digitalfrontiers.de

digital frontiers