Pages, headers and footers

The paper

The paper is Format (A4 by default; Letter, Legal, Tabloid, Ledger, A0 to A6), or Width and Height as CSS lengths, turned with Landscape. A page's own `@page { size: ... }` wins by default (PreferCssPageSize). See paper sizes.

Margins

Margin sets them as CSS lengths (`HtmlMargins.All("15mm")`, or each side). Without it, the page's `@page` margins apply, else Chrome's default margins. See margins. Scale lays the page out as on larger or smaller paper (0.1 to 2): see scale.

Page breaks

CSS decides where pages break: `break-before`, `break-after` and `break-inside: avoid` on elements, `orphans` and `widows` on paragraphs; a table's header rows are printed again on each page it runs over. PageRanges keeps some pages only ("2-3, 5"): see choosing pages.

Headers, footers and page numbers

The page's own CSS can print them in the margins, with `@page` margin boxes; `counter(page)` and `counter(pages)` give the numbers:

```
@page {
  size: A4;
  margin: 18mm 15mm 20mm;
  @bottom-right { content: "Page " counter(page) " of " counter(pages); font: 9pt
Arial; color: #6b7280; }
  @top-left { content: "Quarterly report"; font: 9pt Arial; }
}
```

Or Chrome's header and footer templates, as Puppeteer prints them (DisplayHeaderFooter, HeaderTemplate, FooterTemplate): HTML printed in the margins of each page, its elements of the classes `pageNumber`, `totalPages`, `title`, `url` and `date` filled in. A template that sets no font size prints in Chrome's tiny default size: set one. Give the margins room for them: see the example.

```
var options = new HtmlPdfOptions
{
    DisplayHeaderFooter = true,
    Margin = new HtmlMargins("22mm", "12mm", "20mm", "12mm"),
    FooterTemplate = "<div style='font-size:9px; width:100%; text-align:right;
padding:0 12mm'>Page <span class='pageNumber'></span> of <span class='totalPages'>
</span></div>",
};
```