Quick header example

Chapter 1

This is a collection of 4 Quarto shortcodes for easily inserting header text. The intended use case is for HTML documents that will then be converted to PDF using a tool like pagedjs or weasyprint, but they also work in other settings.

The main shortcode is `{{< quick-header "text" >}}`, where `"text"` is the text to be used in the header. The `quick-header` shortcode can take an optional argument of `side` with values “left”, “right”, or “center”. If not specified (or specified with something other than those 3 correct options) it defaults to “left”. There is also an optional argument `class`, which will add additional classes to the HTML element created; this can be customized with your own CSS, but an example is included for the class `.uppercase`.

The other 3 shortcodes are shorthands for `quick-header`: `{{< hdr-left >}}`, `{{< hdr-right >}}`, and `{{< hdr-center >}}` create headers with `side` set to left, right, or center, respectively.

If the HTML document is printed or converted to PDF with something like pagedjs that relies on CSS print-media settings, these headers will be moved into the page margins using the enclosed CSS stylesheet.

If the HTML is *not* printed, the header elements will just appear where they’ve been placed. If, however, you wrap the headers in a block with the class `.headers`, like above, they will be aligned and evenly spaced in a row, placed in the correct order (left, then center, then right) even if, like in this document, they were created out of order.

If you’re rendering just to HTML, the necessary CSS will be included automatically. If you’re making a PDF, either by setting `pdf-engine` in the yaml or by calling `pagedjs-cli` or something similar on the command line, you’ll need to point to the CSS file as shown in the yaml above.

There is also a very simple filter called `header-page` that creates the requisite bit of CSS to place headers in the margins. In the CSS file, `section.level2` blocks, such as anything appearing after a `##` line, have the page type of `chapter`; this carries over from working with RMarkdown and pagedown. If `header-page` is included in your document, you can pass a meta property `chapter-header-only` in your YAML; if true, *only* chapter pages will have headers. This defaults to false, so *all* pages have headers.

Chapter 2

Here’s some more text to verify that headers get placed on the next page.

Nostrud ut in sunt labore est laborum do nisi laboris magna consequat incididunt ullamco aliquip. Occaecat nulla fugiat anim sint sit commodo labore eiusmod Lorem do commodo. Elit duis dolore aliqua reprehenderit anim incididunt. Esse aliquip sunt elit amet reprehenderit excepteur reprehenderit dolor tempor.

Elit nostrud deserunt tempor eu velit. Occaecat aute sunt reprehenderit velit commodo consequat veniam sit elit est ea aliqua. Adipisicing nisi anim ad do nisi ea irure. Est aliqua mollit ad reprehenderit duis laborum dolore fugiat deserunt commodo esse nulla. Do amet minim consectetur do pariatur dolore amet ut. Consequat labore eu voluptate cupidatat sit ex.

Proident pariatur laborum mollit tempor aliquip. Culpa eu et sint sit ad consectetur elit. Cillum anim veniam duis minim sit officia laborum ad commodo aliquip in esse incididunt.

Sunt quis dolore ea laborum excepteur proident tempor dolor commodo velit ipsum cupidatat dolore mollit. Adipisicing nisi ut velit enim ipsum eiusmod pariatur est. Reprehenderit dolor ut incididunt est cupidatat reprehenderit cillum id irure id officia.