Advanced Styling

Supposed I'm assigned to base a conference paper on my report. Here are the guidelines that I should comply:

1. The font should be an 11pt serif font
2. The title should be in 17 pt and bold
3. The paper contains a single-column abstract and two-column main text
4. The abstract should be centered
5. The main text should be justified (both edges of each line are aligned)
6. First level section headings should be 13pt, centered, and rendered in small capitals
7. Second level headings are run-ins, italicized and have the same size as the body text
8. Finally, the pages should be US letter sized, numbered in the center of the footer and the top right corner of each page should contain the title of the paper

Writing the right set rules

```
// the pages should be US letter sized, numbered in
// the center of the footer and the top right corner
// of each page should contain the title of the paper
#set page(
  paper: "us-letter",
  header: align(right)[
  A fluid dynamic model for
  glacier flow
  ],
  numbering:"1",
  // "(1/1)" to should current and total pages
  )


// The main text should be justified
#set par(justify: true)


// The font should be an 11pt serif font
#set text(
  font: "Libertinus Serif",
  size: 11pt,
  )
```

Creating a title and abstract

```
  // The title should be in 17 pt and bold
  #align(center, text(17pt)[
    *A fluid dynamic model for glacier flow*
  ])
  #grid(
    columns: (1fr, 1fr),
    align(center)[
      Therese Tungsten \
      Artos Institute \
      #link("mailto:tung@artos.edu")
    ],
    align(center)[
      Dr. John Doe \
      Artos Institute \
      #link("mailto:doe@artos.edu")
    ]
  )
```

```
  // The paper contains a single-column abstract
  // The abstract should be centered
  #align(center)[
    #set par(justify: false) //doesn't affect the
remainder of the document
    *Abstract* \
    #lorem(80)
  ]
```

Tip: use variables

By saving the paper title in a variable, so that we do not have to type it twice.

```
#let title = [
  A fluid dynamic model for glacier flow
]

#set page(
  header: align(
    right + horizon,
    title,
  ),
)

#align(
  center,
  text(17pt)[
    *#title*
  ],
)
```
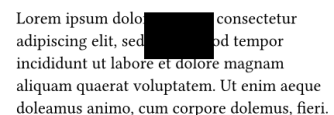
Adding columns

To switch our paper to a two-column layout, simply amend our `page` set rule with the `columns` argument. However, by adding `columns: 2` to the argument list, we have wrapped the whole document in two column.

To keep title, authors and abstract spanning the whole page, we can wrap them in a function called `place`. It expects an alignment and the content it should place. Using the `named scope` argument, we can decide if the items should be placed relative to the current column or its parent.

What's more, if no other arguments are provided, `place` takes its content out of the flow of the document, and positions it over the other content without affecting the layout of other content in its container. For example:

```
#place(
  top + center,
  rect(fill: black),
)
#lorem(30)
```

Lorem ipsum dolo⬛⬛⬛consectetur adipiscing elit, sed⬛⬛⬛od tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri.

Since the text below the block acts like as if there was nothing, and we don't want title and others cover the main text, we pass the argument `float: true` to `place`. Also, we remove `#set align(center)`s inside it since they inheit the center alignment from the placement.

```
#set page(
  paper: "us-letter",
  header: align(
    right + horizon,
    title
  ),
  numbering: "1",
  columns: 2,
)

#place(
  top + center,
  float: true,
  // be placed relative to its parent (page)
  scope: "parent",
  // provide the space between it and the body
  clearance: 2em,
)[
  ...

  #par(justify: false)[
    *Abstract* \
    #lorem(80)
  ]
]

= Introduction
#lorem(300)

// here is a second level headings,
// the example on official website seems wrong
== Related Work
#lorem(200)
```
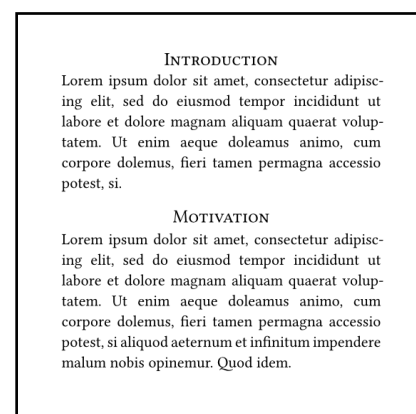
Adding headings

By writing our own heading show rule, we can style our headings.

```
// First level section headings should be 13pt,
// centered, and rendered in small capitals
#show heading: it => [
  #set align(center)
  #set text(13pt, weight: "regular")
  #block(smallcaps(it.body))
]
```

Now, there's only one rule left. To make second level headings looks different from the firsts, we use a `where` selector on our set rule. It's a method we can call on headings (and other elements) that allows us to filter them by their level. We can use it to differentiate between section and subsection headings:

```
#show heading.where(
  level: 1
): it => block(width: 100%)[
  #set align(center)
  #set text(13pt, weight: "regular")
  #smallcaps(it.body)
]

#show heading.where(
  level: 2
): it => text(
  size: 11pt,
  weight: "regular",
  style: "italic",
  it.body + [.],
)
```


Review

Now, all guidelines are done perfectly. We learn to use functions and the `where` method, and custom heading styles. The paper is a great success! Here is the final work: