Authoring presentations with Quarto

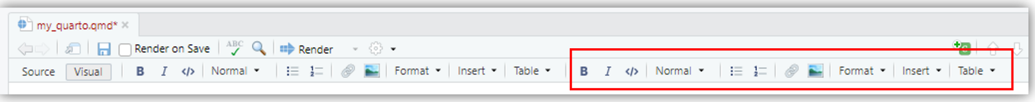
1 Quarto Presentations

Another great feature of Quarto is that it allows you to create presentations and supports a variety of formats such as html, power point and beamer. Today we will focus on creating an html presentation (which can be also printed as a pdf).

Creating an html presentation is as easy as setting the output format to `revealjs`:

```
---

title: "My title"

author: "Jafet Belmont"

format: revealjs

---
```

This will create a title slide that includes the provided `title` and `author`. (you can remove any of these options if you the author or title page to be removed). Now we will switch to the `visual` mode. You will notice that a new set of options have now appeared:



This new tab will allow us to interact with the different features of our presentation. Lets begin by creating some slides. Slides in Quarto are delineated using level 1 and 2 headings:

```
---

title: "My title"

author: "Jafet Belmont"

format: revealjs

---

# Topic 1

## Slide 1.1

# Topic 2
```

```
## Slide 2.1
```

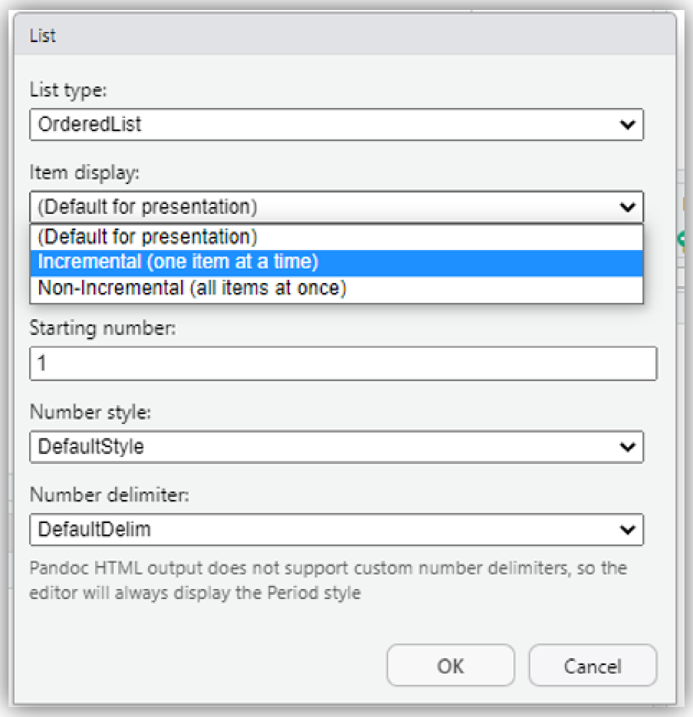
In the example above we use level 1 headings for creating a new sections and level 2 heading for defining the new slides. Alternatively we can use horizontal rules to create the slides as follows (on `visual` mode click on `Insert` `Horizontal Rule`):

```
---

title: "My title"

author: "Jafet Belmont"

format: revealjs

---

  Content of slide 1

---

  Content of slide 2

---
```

You can add bullet and numbered lists to each slide as follows (on `visual` mode click on ☰ or 12☰ for adding bullet or numbered lists respectively):

```
---

title: "My title"

author: "Jafet Belmont"

format: revealjs

---

## Slide 1

1.  Item 1

2.  Item 2

## Slide 2

-   Bullet 1

    -   Bullet 1.1

    -   Bullet 1.2
-   Bulet 2

    -   Bullet 2.1

    -   Bullet 2.2
```

Notice that indentation allow us to create a hierarchical structure within our lists of items. You can modify some of the option of the list by clicking the ··· button on the right side of it. This will open a pop-up window where you can modify some of its attributes. For example, you can ask Quarto to display each item one by one as you move forward through the slides:



List

List type: OrderedList

Item display: (Default for presentation)

(Default for presentation)
Incremental (one item at a time)
Non-Incremental (all items at once)

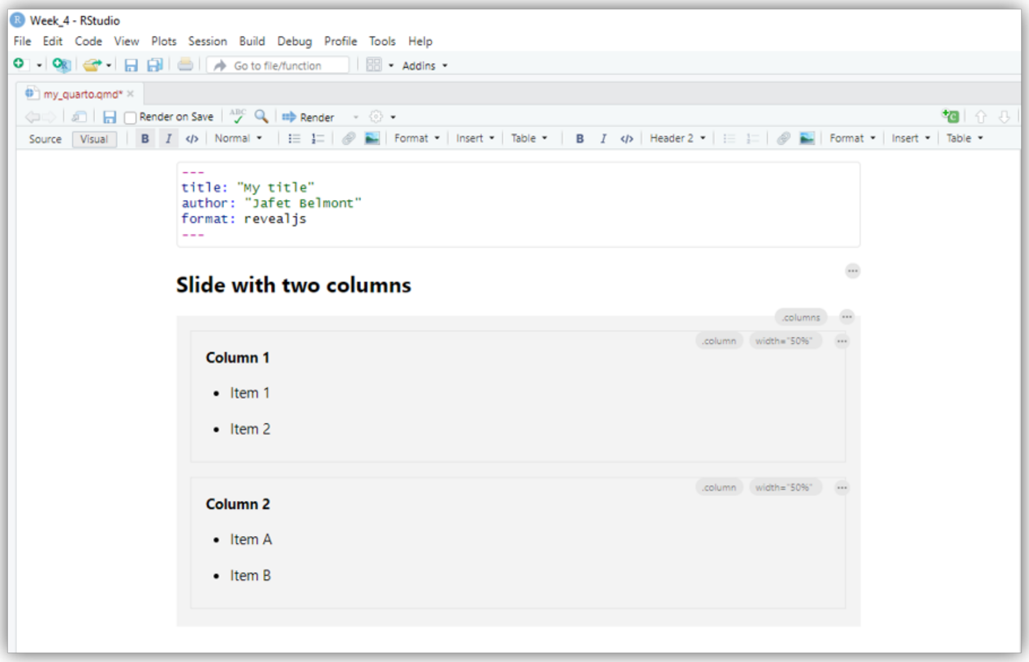
Starting number: 1

Number style: DefaultStyle

Number delimiter: DefaultDelim

Pandoc HTML output does not support custom number delimiters, so the editor will always display the Period style

If you want to arrange the content of you slide into different columns you can insert a multiple column output. To do this switch to `visual` mode and click on `Insert` `Slide Columns`). You can then fill each column:



```
---
title: "My title"
author: "Jafet Belmont"
format: revealjs
---
```

Slide with two columns

Column 1

- Item 1
- Item 2

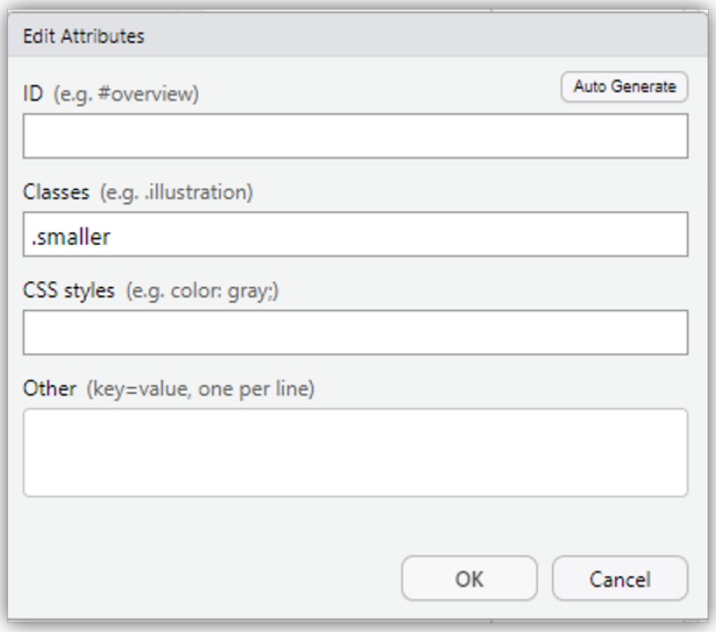
Column 2

- Item A
- Item B

Sometimes the text size in your slide might be too large to fit in your slide and you would like to make it smaller. You can then use the .smaller class to use a smaller typeface so that more text fits on the slide.

```
---

title: "My title"

author: "Jafet Belmont"

format: revealjs

---

## Slide with small text {.smaller}

1.  Item 1

2.  Item 2
```

If you are on the visual model, click on the slide settings ⋯ and then type `.smaller` on the class box:


Edit Attributes

ID (e.g. #overview)

Classes (e.g. .illustration)

.smaller

CSS styles (e.g. color: gray;)

Other (key=value, one per line)

Note

Note that ideally you slides should not contain too much text and thus this is just a quick way around if you feel that you need a bit of extra space in your slide.

R code can be embedded in any slides by including a R chunk in the same way as we did before. However, sometimes we would like to arrange the code output to be display in a "tidy" manner within the slide. To achieve this we can use the `output-location:` argument which has the following options:

- `column`: Display output in a column adjacent to the code
- `column-fragment`: Display output in a column adjacent to the code and delay showing it until its explicitly stepped through by advancing the slides.
- `fragment`: Delay the output display until it is explicitly stepped through by advancing the slides.
- `slide`: Display output on the subsequent slide.

  For example, the source code for a chunk within a slide containing an R code would look something like this:

```
#| output-location: column-fragment

ggplot(evals.scores, aes(x = bty_avg, y = score)) +

  geom_point() +

  labs(x = "Beauty Score", y = "Teaching Score") +

  geom_smooth(method = "lm", se = FALSE)
```