

University of the Cordilleras  
Communication in the Workplace  
CC11-1L<sub>A</sub>

# Using Typst to create a UC Schedule Graphic:

*A Technical Writing Document*

*Presented to the Faculty of the  
College of Information Technology and Computer Science  
University of the Cordilleras*

*Submitted by:  
Del Castillo, Carlos Romeo C. III*

*Submitted to:  
Sir Paul Terese Catala  
28 Jul. 2025*



# Table of Contents

<b>1 Rationale .....</b>	<b>3</b>
<b>2 Introduction to the Emerging Technology .....</b>	<b>3</b>
<b>3 How to use Typst and Romeoscheduler to create a UC Schedule Graphic .....</b>	<b>4</b>
3.1 Step 1: Navigate to Typst's local packages folder.....	5
3.2 Step 2: Clone Romeoscheduler to the local packages folder.....	5
3.3 Step 3: Create the Typst file.....	5
3.4 Step 4: Import the Romeoscheduler plugin.....	5
3.5 Step 5: Create the YAML configuration file.....	5
3.6 Step 6: Configure the settings for the schedule graphic.....	5
3.7 Step 7: Input Subjects.....	7
3.8 Step 8: Import the YAML configuration.....	9
3.9 Step 9: Compile the Typst file into an image.....	9
<b>4 Acknowledgements .....</b>	<b>9</b>
<b>■ References .....</b>	<b>9</b>
<b>A Sample YAML Configuration File .....</b>	<b>11</b>

## 1 Rationale

The course reservation interface of SchoolAutomate, used by students of the University of the Cordilleras before the beginning every term to reserve their courses, does not currently have a way to visualise course schedules. The most students get in the field of visualisation is a piece of text detailing the days and the times of each course.

Because of this, students report unknowingly getting themselves into unfavourable schedules (in extreme cases, having as much as seven periods (*9 hours and 20 minutes*) without any breaks) due to the lack of a visual aid when reserving schedules.

Online tools for visualising schedules [1], [2], [3] either lack adjustment to the University's eighty-minute fixed periods, lack customisation options expected for a visual aid tool, require signup with an email account, or all of the above. A free, open-source, no-login, customisable solution to visualising University of the Cordilleras schedules is therefore needed to help students plan their course enrolments and avoid unfavourable schedules.

## 2 Introduction to the Emerging Technology

*Typst* [4] /taipst/ is an open-source digital typesetting system first developed in 2019 by Martin Haug and Laurenz Mädje, students at the Technische Universität Berlin. It was made primarily out of frustrations with LaTeX, the prime typesetting software for papers in academia [5], [6]. It boasts a comparatively simpler markup system inspired by Markdown [7] for basic elements, a Turing-complete scripting system, integrated functions for advanced math and bibliography management, and millisecond-level document compile times.

Users can edit *Typst* files online in an online editor [8] or offline, through the installation of the *Typst* command-line binary either through the project's Github releases page [9] or through the package manager of their choosing [4]. As a result of its powerful scripting system and simple markup, users have found applications for *Typst* inside and outside of academia, with user-submitted packages available [10] for things such as academic paper templates, curriculum vitae (CV) templates, school notes templates, and plugins for data visualisation (`lilaq`), diagram (`fletcher`) and graphic drawing (`cetz`), computer program execution visualisation (`stack-pointer`), QR code generation (`tiaoma`), and others. In addition, *Typst* was also used to typeset this very document.

Given the possible and proven applications of *Typst*'s scripting-powered typesetting system in making documents and graphics, it emerges as a serious choice among typesetting systems, especially for purposes that require fast compilation times and easy-to-understand configuration.

The screenshot shows a Typst editor interface. On the left, the code editor displays a Typst document with various sections and code snippets. On the right, a preview window shows a table of the first 8 Fibonacci numbers. The code includes sections for setting page properties, defining a Fibonacci sequence, and generating a table of values.

```

1 #set page(width: 10cm, height: auto)
2 #set heading(numbering: "1.")
3
4 = Fibonacci sequence
5 The Fibonacci sequence is defined through the recurrence relation
$F_n = F_{n-1} + F_{n-2}$. It can also be expressed in closed
form:
6
7 $ F_n = round(1 / sqrt(5) phi.alt^n), quad
8 | phi.alt = (1 + sqrt(5)) / 2 $
9
10 #let count = 8
11 #let nums = range(1, count + 1)
12 #let fib(n) =
13 | if n ≤ 2 { 1 } else { fib(n - 1) + fib(n - 2) }
14 }
15
16 The first #count numbers of the sequence are:
17 #align(center, table(
18 | columns: count,
19 | ..nums.map(n => $F_#n$),
20 | ..nums.map(n => str(fib(n))),
21 ))

```

$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$	$F_7$	$F_8$
1	1	2	3	5	8	13	21

Figure 1: An example Typst document taken from the project's Github page, demonstrating the language's markup, math typesetting, and scripting capabilities

### 3 How to use Typst and Romeoscheduler to create a UC Schedule Graphic

To fill the need for a stable, customisable way of visualising student schedules, and to demonstrate the capabilities of Typst in graphic document creation, the author has developed a plugin, with the temporary name « *Romeoscheduler* » [11], using pure Typst (with configuration done using a YAML file) that can produce high-quality schedule graphics for students of the University of the Cordilleras and for all other purposes that require fixed-length periods.

To use Romeoscheduler, the user will require:

- An installation of the **Typst command-line binary**, available at <https://github.com/typst/typst/releases/>,
- An installation of the **Git command-line interface (CLI)**,
- A **terminal emulator** such as Windows' Command Prompt,
- A **text editor**,
- The **Material Symbols font**, available at <https://github.com/google/material-design-icons/tree/master/variablefont> (for icons), and
- Ideally a thin font such as Iosevka (<https://github.com/be5invis/Iosevka/>) for information density.

A sample configuration and output compiled and built using these steps is available at [Appendix A](#).

## 3.1 Step 1: Navigate to Typst's local packages folder.

Typst can load local packages as long as they are installed properly.

Open a command prompt window and navigate to the following directory depending on your operating system by running the `cd` command:

- On Windows: `cd %APPDATA%/typst/packages/local`
- On MacOS: `cd ~/Library/Application Support/typst/packages/local`
- On Linux: `cd ~/.local/share/typst/packages/local`

*Note: If the directory does not already exist, you have to create it first:*

- On Windows: `mkdir %APPDATA%/typst/packages/local`
- On MacOS: `mkdir -p ~/Library/Application Support/typst/packages/local`
- On Linux: `mkdir -p ~/.local/share/typst/packages/local`

## 3.2 Step 2: Clone RomeoScheduler to the local packages folder.

- After navigating to the folder in the command prompt, download the RomeoScheduler plugin to the folder by running the command:

```
git clone https://github.com/carocledc3/romeoscheduler.git
```

## 3.3 Step 3: Create the Typst file.

Now that the plugin has been downloaded to the correct folder, it is ready to be used. Create a Typst file anywhere by creating a plain text file with the file extension set to `.typ`.

## 3.4 Step 4: Import the RomeoScheduler plugin.

Open the newly created Typst file with a text editor of choice and insert the following line:

```
#import "@local/romeoscheduler:0.0.1": scheduler
```

## 3.5 Step 5: Create the YAML configuration file.

In the same folder as the Typst file, create a plain text file with the extension `.yaml`. This will serve as the configuration file for the schedule graphic that Typst will render. Each particular schedule graphic is defined by its configuration file.

## 3.6 Step 6: Configure the settings for the schedule graphic.

Open the `.yaml` file in a text editor. Input the following text:

```
parameters: {  
}
```

Inside the brackets one can configure different settings for how Romeoscheduler renders the graphic. These are inserted as key-value pairs of the setting name and its value. Entries themselves are separated by commas. As an example, the default configuration is shown below:

```
parameters: {  
    title: Schedule,  
    period-length: 80,  
    day-start: 450,  
    height: 8,  
    days: 6,  
    offset: 0,  
    font-size: 16,  
    page-width: 16,  
    page-height: 26,  
    font: RomeosevkaQP,  
    font-size: 16,  
    militarytime: false,  
}
```

If any of these settings are not specified, the above default values will be used instead. The available settings are as follows:

- **title**: A string (sequence of characters). The title of the graphic that will appear in large font at the top of it. **Default: Schedule**
- **period-length**: An integer. The length of each period in the graphic, in minutes. **Default: 80**
- **day-start**: An integer. The start time of the first period, in minutes from midnight, from where all other times in the schedule will be based. **Default: 450** (corresponding to 7:30 am)
- **height**: An integer. How many periods the schedule graphic will have. **Default: 8**
- **days**: An integer. How many days, with Monday as the first, the graphic will include. **Default: 6**
- **offset**: How many periods downward the schedule graphic will be shifted by. By default, the first period in the graphic is Period 1, but if **offset** is 1, then the first period graphic is instead the next one (Period 2), and so on. **Default: 0**
- **page-width**: The width of the schedule graphic in inches. **Default: 16**
- **page-height**: The height of the schedule graphic in inches. **Default: 16**

- `font`: A string. The name of the font family to be used in the graphic. **Default: Iosevka**
- `font-size`: A number. The base font size in points. **Default: 16**
- `military-time`: A boolean (true/false). Whether to use 24-hour format in the schedule's times. **Default: false**

## 3.7 Step 7: Input Subjects.

In the same file, outside the brackets of the `parameters`, create another entry with brackets called `subjects`, like so:

```
subjects: {
}
```

Inside the `subjects` entry, subjects can be inputted as sub-entries with unique identifiers, ultimately with the subject entries themselves and their settings separated by commas. Settings per subject can also be configured to adjust the subject's appearance and to adjust where it appears on the schedule grid. An example for one subject is provided below:

```
subjects: {
  cc10: {
    schedule: 9th,
    lab-schedule: 8thr,
    code: CC10,
    name: Introduction to Networks,
    section: 1J,
    teacher: Jhunfel Dela Cruz,
    lab-teacher: Jhunfel Dela Cruz,
    room: S311,
    lab-room: S312,
    colour: '#02adcc',
    textcolour: '#ebfbff',
    icon: wifi,
  },
}
```

The example above shows a subject with the identifier `cc10`, and all the settings available for a subject entry.

All the possible settings for a subject entry are as follows. All subject options are optional except for `schedule`. If any of these settings are not specified, the default values will be used instead:

- `schedule`: **Required.** A string. A shorthand notation for the subject's (lecture) schedule with a specific format:
  - ▶ The string should begin with a number corresponding to the period the subject occurs on. For the default settings of the University of the Cordilleras, where the school day starts at 07:30 and periods are 80 minutes each:
    - Period 1 corresponds to the period 7:30am → 08:50am,
    - 2 to 8:50am → 10:10am,
    - 3 to 10:10am → 11:30am,
    - 4 to 11:30am → 12:50pm,
    - 5 to 12:50pm → 2:10pm,
    - 6 to 2:10pm → 3:30pm,
    - 7 to 3:30pm → 4:50pm,
    - 8 to 4:50pm → 6:10pm, and
    - 9 to 6:10pm → 7:30pm.
  - ▶ The number should be followed by letters corresponding to which days the subject occurs on. Allowed characters are:
    - m for Monday, t for Tuesday, w for Wednesday, h for Thursday, f for Friday, r for Saturday, and s for Sunday.
  - ▶ In the example, the subject's `schedule` code is 9th, which says that the schedule's (lecture) period is on **Period 9, Tuesdays and Thursdays**.
- `lab-schedule`: A string. A shorthand notation for the subject's laboratory schedule, if any. The notation is the same as in the `schedule` setting.
- `code`: A string. The subject's shorthand code as it appears in the University's system. **Default:** CCxx
- `name`: A string. The subject's full name. **Default:** Sample Subject Name
- `section`: A string. The specific section of the subject. **Default:** 1x
- `teacher`: A string. The instructor for the (*lecture portion of the*) subject. **Default:** Dr. Gregory House
- `lab-teacher`: A string. Same as `teacher` but for the laboratory portion, if any.
- `room`: A string. The name of the room the (*lecture portion of the*) subject happens in. **Default:** Room
- `lab-room`: A string. Same as `room` but for the laboratory portion, if any.

- `colour`: A colour in hexadecimal notation. The subject's background colour on the schedule. **Default:** `#4040dd`
- `textcolour`: A colour in hexadecimal notation. The colour of the text on the subject's cell on the schedule. **Default:** `#eeeeee`
- `icon`: A string corresponding to an icon name of any of the icons at <https://fonts.google.com/icons>.

## 3.8 Step 8: Import the YAML configuration.

Once all the subjects and settings have been configured, go back to the Typst file and import the YAML configuration file into the plugin's `scheduler` function by inserting the line below:

```
#scheduler(yaml(<config_file_name>.yaml))
```

Where `<config_file_name>.yaml` is to be replaced with the actual file name of the YAML file. After this, save the changes to the Typst file.

## 3.9 Step 9: Compile the Typst file into an image.

After all files have been saved, open a command prompt window and navigate to the directory that the Typst and YAML files are in. Run the command:

```
typst compile <typst_file_name>.typ <image_name>.png
```

Where `<typst_file_name>.typ` is the filename of the Typst file and `<image_name>.png` is the filename of the schedule graphic iamge you want to export.

## 4 Acknowledgements

The author extends his gratitudes to the developers of Typst, Martin Haug and Laurenz Mädje, for developing the tool and changing his school document production workflow for the better.

No generative AI is or will ever be used in the creation of this project. The source code of the project is freely available at <https://github.com/carocledc3/romeoscheduler>.



## References

- [1] "Online Timetable Generator." [Online]. Available: <https://calculatorgrid.com/online-timetable-generator/>
- [2] "Free Class Schedule Maker | Create Perfect Course Timetables Online." [Online]. Available: <https://www.timetablemaster.com/class-schedule-maker>
- [3] "Class Schedule Maker - Create Free Timetables Online | Canva." [Online]. Available: <https://www.canva.com/create/class-schedules/>

- [4] “typst/typst: A new markup-based typesetting system that is powerful and easy to learn.” [Online]. Available: <https://github.com/typst/typst>
- [5] “CTAN: TeX.” [Online]. Available: <https://www.ctan.org/tex>
- [6] A. Gaudeul, “Do open source developers respond to competition? The LATEX case study,” *Review of Network Economics*, vol. 6, no. 2, Jan. 2007, [Online]. Available: <https://doi.org/10.2202/1446-9022.1119>
- [7] J. Gruber, “Daring Fireball: Markdown Syntax Documentation,” Daring Fireball. [Online]. Available: <http://daringfireball.net/projects/markdown/syntax#philosophy>
- [8] “Typst: Compose papers faster.” [Online]. Available: <https://typst.app/>
- [9] “Releases · typst/typst.” [Online]. Available: <https://github.com/typst/typst/releases>
- [10] “Search — Typst: Universe.” [Online]. Available: <https://typst.app/universe/search/>
- [11] C. R. Del Castillo, “carocledc3/romeoscheduler,” Github. [Online]. Available: <https://github.com/carocledc3/romeoscheduler>

## A Sample YAML Configuration File

This configuration file, when inputted into Romeoscheduler, creates a schedule graphic for the block section CITCS-1J:

```
parameters: {
    title: CITCS-1J - 3rd Term Schedule,
    height: 9,
}

subjects: {
    cc9: {
        code: CC9,
        name: Discrete Structures,
        section: 1J,
        teacher: Luis William Meing,
        room: U705,
        schedule: 2mwf,
        colour: '#0cb022',
        textcolour: '#e4ffdd',
        icon: functions,
    },
    cc10: {
        code: CC10,
        name: Introduction to Networks,
        section: 1J,
        teacher: Jhunfel Dela Cruz,
        lab-teacher: Jhunfel Dela Cruz,
        room: S311,
        lab-room: S312,
        schedule: 9th,
        lab-schedule: 8thr,
        colour: '#02adcc',
        textcolour: '#ebfbff',
        icon: wifi,
    },
    cc11: {
        code: CC11,
        name: Communication in the Workplace,
        section: 1J,
        teacher: Bretz Harllynne Moltio,
        room: U706,
        schedule: 1mwf,
        colour: '#c50d3a',
        textcolour: '#ffe6ec',
        icon: 3p,
    },
    cc12: {
        code: CC12,
        name: Statistical Design and Analysis,
        section: 1J,
        teacher: Jessie Martirez,
        lab-teacher: Jessie Martirez,
        room: S311,
        lab-room: M306,
    }
}
```

```

    schedule: 8mw,
    lab-schedule: 7mwf,
    colour: '#ef7208',
    textcolour: '#fffce1',
    icon: scatter_plot,
  },

  cc13: {
    code: CC13,
    name: Systems Analysis and Design,
    section: 1J,
    teacher: Felipe Pati Jr.,
    room: M301,
    schedule: 5thr,
    colour: '#fcdf6a',
    textcolour: '#814500',
    icon: manufacturing,
  },

  cc22: {
    code: CC22,
    name: Introduction to Platform Technologies,
    section: 1J,
    teacher: Leonard Prim Reyes,
    lab-teacher: Leonard Prim Reyes,
    room: S311,
    lab-room: M304,
    schedule: 5mw,
    lab-schedule: 4mwf,
    colour: '#032568',
    textcolour: '#e8f0ff',
    icon: terminal,
  },

  pathfit3: {
    code: Pathfit3,
    name: Martial Arts,
    section: 1J,
    teacher: Niño Felix,
    room: Gymnasium,
    schedule: 7th,
    colour: '#c51ibe',
    textcolour: '#ffddefb',
    icon: sports_martial_arts,
  },

  nstp2: {
    code: NSTP2,
    name: National Service Training Programme 2,
    section: 1J,
    teacher: Sheena Palaroan,
    room: N/A,
    schedule: 4thr,
    colour: '#8e3c3c',
    textcolour: '#ffddcc',
    icon: flag,
  }
}

```

The above configuration can be read with a Typst file with the following contents:

```
#import "@local/romeoscheduler:0.0.1": scheduler
#show "Communication": "Communi-cation" // hyphenation for CC11 title so it breaks across lines
#scheduler(yaml("sample-conf-1j.yaml"))
```

Which will produce the following output:

CITCS-1J - 3rd Term Schedule						
	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
Period 1 7:30a 8:50a	U706 Bretz Harlynne Moltio 7:30a - 8:50a CC11-1J Communication in the Workplace		U706 Bretz Harlynne Moltio 7:30a - 8:50a CC11-1J Communication in the Workplace		U706 Bretz Harlynne Moltio 7:30a - 8:50a CC11-1J Communication in the Workplace	
Period 2 8:50a 10:10a	U705 Luis William Meing 8:50a - 10:10a CC9-1J Discrete Structures		U705 Luis William Meing 8:50a - 10:10a CC9-1J Discrete Structures		U705 Luis William Meing 8:50a - 10:10a CC9-1J Discrete Structures	
Period 3 10:10a 11:30a						
Period 4 11:30a 12:50p	M304 Leonard Prim Reyes 11:30a - 12:50p NSTP2-1J National Service Training Programme 2	N/A Sheena Palaroan 11:30a - 12:50p NSTP2-1J National Service Training Programme 2	M304 Leonard Prim Reyes 11:30a - 12:50p NSTP2-1J National Service Training Programme 2	N/A Sheena Palaroan 11:30a - 12:50p NSTP2-1J National Service Training Programme 2	M304 Leonard Prim Reyes 11:30a - 12:50p NSTP2-1J National Service Training Programme 2	N/A Sheena Palaroan 11:30a - 12:50p NSTP2-1J National Service Training Programme 2
Period 5 12:50p 2:10p	S311 Leonard Prim Reyes 12:50p - 2:10p CC22-1J Introduction to Platform Technologies	M301 Felipe Pati Jr. 12:50p - 2:10p CC13-1J Systems Analysis and Design	S311 Leonard Prim Reyes 12:50p - 2:10p CC22-1J Introduction to Platform Technologies	M301 Felipe Pati Jr. 12:50p - 2:10p CC13-1J Systems Analysis and Design		M301 Felipe Pati Jr. 12:50p - 2:10p CC13-1J Systems Analysis and Design
Period 6 2:10p 3:30p						
Period 7 3:30p 4:50p	M306 Jessie Martirez 3:30p - 4:50p CC12-1J Statistical Design and Analysis	Gymnasium Nina Felix 3:30p - 4:50p Pathfit3-1J Martial Arts	M306 Jessie Martirez 3:30p - 4:50p CC12-1J Statistical Design and Analysis	Gymnasium Nina Felix 3:30p - 4:50p Pathfit3-1J Martial Arts	M306 Jessie Martirez 3:30p - 4:50p CC12-1J Statistical Design and Analysis	
Period 8 4:50p 6:10p	S311 Jessie Martirez 4:50p - 6:10p CC12-1J Statistical Design and Analysis	S312 Jhumfel Dela Cruz 4:50p - 6:10p CC10-1J Introduction to Networks	S311 Jessie Martirez 4:50p - 6:10p CC12-1J Statistical Design and Analysis	S312 Jhumfel Dela Cruz 4:50p - 6:10p CC10-1J Introduction to Networks		S312 Jhumfel Dela Cruz 4:50p - 6:10p CC10-1J Introduction to Networks
Period 9 6:10p 7:30p		S311 Jhumfel Dela Cruz 6:10p - 7:30p CC10-1J Introduction to Networks		S311 Jhumfel Dela Cruz 6:10p - 7:30p CC10-1J Introduction to Networks		

Figure 2: Schedule graphic output for the sample configuration specified above.