Starting

Starting typing in Typst is easy. You don’t need packages or other weird things for most of things.

Blank line will move text to a new paragraph.

Btw, you can use any language and unicode symbols without any problems as long as the font supports it: ßçœ̃ɛ̃ø∀αβёыა😆 …

Markup

This was a heading. Number of = in font of name corresponds to heading level.

Second=level heading

Okay, let’s move to *emphasis* and **bold** text.

Markup syntax is generally similar to `AsciiDoc` (this was `raw` for monospace text!)

New lines & Escaping

You can break
line anywhere you
want using “\” symbol.

Also you can use that symbol to escape _all the symbols you want_, if you don’t want it to be interpreted as markup or others special symbols.

Comments & codeblocks

You can write comments with `//` and `/* comment */`:

```
Just in case you didn't read the source, this is how it is written:

// Like this
/* Or even like
this */

By the way, I'm writing it all in a _fenced code block_ with *syntax highlighting*!
```

Smart quotes

What else?

There are no much things in basic “markup” syntax, but we will see much more interesting things very soon! I hope you noticed auto-matched “smart quotes” there.

Lists

- Writing list in a ssimple way is great.
- Nothing complex, start your points with `-` and this will become a list.
  - Indented lists are created via indentation.

1. Numbered lists star with `+` instead of `-`.
2. There is no alternative markup syntax for lists
3. So just remember `-` and `+`, all other symbols wouldn’t work in an unintended way.
   1. That is a general property of Typst’s markup.
   2. Unlike Markdown, there is only one way to write something with it.

Notice

Typst numbered lists differ from markdown-like syntax for lists. If you write them by hand, numbering is preserved:

1. Apple
1. Orange
1. Peach

Math

I will hist mention math ($a + \frac{b}{c} = \sum_1 x^i$) is possible and quite pretty there:

$$7.32\beta + \sum_{i=0}^{\nabla} \frac{Q_i(a_i - \varepsilon)}{2}$$

To learn more about math, see corresponding chapter.

Functions

Okay, let’s now move to more complex things.

First of all, there are **lots of magic** int Typst. And it major part of it is called “scripting”.

To go to scripting mode, type # and **some funcion name** after that. We will start with *something dull*:

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

*That **function** just generated 50 “Lorem Ipsum” words!*

More functions

functions can do everything!

Like Really Everything

Figure 1: This is an imaginary screenshot from one of first theses written in Typst.
All these things are written with custom functions too.

How to call functions

First, stat with `#`. Then write the name. Finally, write some parentheses and maybe something inside.

You can navigate lots of built-in functions in Official Reference.

> That’s right, links, quotes and lots of other document elements are created with functions.
>
> — Typst Examples Book

Function arguments

There are *two types* of function arguments:

1. **Positional.** Like `50` in `lorem(50)`. Just write them in parentheses and it will be okay. If you have many, use commas.
2. **Named.** Like in `#quote(attibution: "Whoever")`. Write the value after a name and a colon.

If argument is named, it has some *default value*. To find out what it is, see Official Typst Reference.

Content

The most “universal” type un Typst language is **content**. Everything you write in the document becomes content.

But you can explicitly create it with *scripting mode* and **square brackets**.

There, in square brackets, you can use any markup functions or whatever you want.