# run-liners: User Manual

# **Contents**

1.	Introduction	. 2
2.	run-in-enum: Run-in numbered lists	. 2
	run-in-list: Run-in bullet points	
	run-in-terms: Run-in term definitions	
	run-in-verse: Run-in poetry lines	
	Direct run-in-join	
	Conclusion	

#### 1. Introduction

This package provides a set of functions that create "run-in" lists and enumerations, following guidelines similar to the **Chicago Manual of Style**. You can produce run-in enumerations, bullet lists, terms/definitions, or verses, all joined in a single line with flexible separators and optional words like "and," "or," or *none* (i.e. omit a coordinator).

We'll illustrate each function using **the left column**: a **typst** block containing the Typst code you'd write and **the right column**: the rendered result.

#### 2. run-in-enum: Run-in numbered lists

run-in-enum is typically used for run-in enumerations like "(1) apple, (2) banana, and (3) cherry." It offers these parameters:

- separator: Defaults to "auto" (which tries to detect commas and switch to semicolons).
- coordinator: The word or phrase before the last item—defaults to "and", can be "or", "and/or", or none.
- numbering-pattern: A pattern like "(1)", "(A)", or any format the numbering() call supports.
- numbering-formatter: A function that can style or transform the generated number.

#### 2.1. Example: Basic Three-Item Enumeration



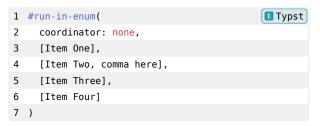
(1) Apple, (2) Banana, and (3) Cherry

This example uses all defaults:

- separator: "auto" → because there's no comma in the content, it uses commas.
- coordinator: "and" → puts "and" before the last item.
- numbering-pattern: "(1)"  $\rightarrow$  "(1)(2)(3)"
- numbering-formatter: (it) => [#it] → no special styling.

#### 2.2. Example: Coordinator = none, 4 Items, Comma in content

If one of your items contains [text, with comma], the auto separator chooses "; ".

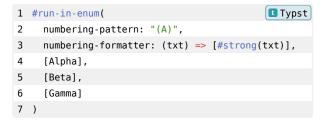


(1) Item One; (2) Item Two, comma here; (3) Item Three; (4) Item Four

Notice it used semicolons because of comma detection, **and** no coordinator word (because **coordinator**: none).

#### 2.3. Example: Custom Numbering Pattern & Styling

You can change numbering-pattern to "(A)" or "(i)" and style the numbers with numbering-formatter.



(A) Alpha, (B) Beta, and (C) Gamma

Here, (A), (B), (C) appear in **bold**.

#### 2.4. Example: Single Item & Two Items Edge Cases

When only one item is present, no separator or coordinator is used. With two items, you'll see just one separator plus "and/or/or" (depending on coordinator).



# 3. run-in-list: Run-in bullet points

run-in-list behaves like run-in-enum but uses a **bullet (or any marker)** before each item. The rest of the logic is the same—separator, coordinator, etc. See the "run-in-enum" section for **how auto detection** and **coordinator** work.

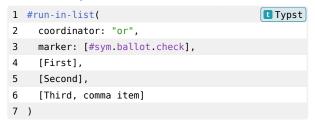
### 3.1. Example1: Default Bullets

```
1 #run-in-list(
2 [Apple],
3 [Banana],
4 [Cherry]
5 )
```

• Apple, • Banana, and • Cherry

You get bullets [•] for each item by default.

#### 3.2. Example: Custom Marker, Coordinator = 'or'



☑ First; ☑ Second; or ☑ Third, comma item

Because of the comma, it uses semicolons. The coordinator is now "or".

## 4. run-in-terms: Run-in term definitions

run-in-terms is for run-in terms and definitions. By default, each term is **bold** ([#strong(term)]) followed by the definition, and you still get the same **separator** and **coordinator** logic as in run-in-enum.

#### 4.1. Example: Three Terms, No Coordinator

```
1 #run-in-terms(
2 coordinator: none,
3 (
4 ([Goal], [Summarize the prior research]),
5 ([Methods], [Outline the approach]),
6 ([Outcome], [Analyze the results])
7 )
8 )
```

**Goal**: Summarize the prior research, **Methods**: Outline the approach, **Outcome**: Analyze the results

See how no "and" or "or" appears. It detects commas if they exist.

#### 4.2. Example: Custom Term Formatter

```
1 #run-in-terms(
2 term-formatter: (txt) => [#underline(txt)],
3 ([Key], [Value 1]),
4 ([Another Key], [Value 2])
5 )
```

Key: Value 1 and Another Key: Value 2

Each term is underlined, with a colon after the term.

# 5. run-in-verse: Run-in poetry lines

run-in-verse simply joins lines (verses) with a given **separator** (default: [ /~]), and no coordinator. It's much simpler since you typically don't want "and" in between lines of poetry.

## 5.1. Example: Default Verse Joining

```
1 #run-in-verse(
2 [Line 1],
3 [Line 2],
4 [Line 3]
5 )
```

Line 1 / Line 2 / Line 3

#### 5.2. Example: Custom Separator

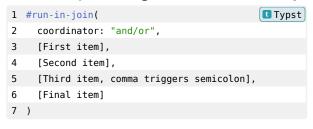
Line A, Line B, Line C

# 6. Direct run-in-join

All these functions ultimately call **run-in-join**, which you can use yourself if you want to create custom transformations. You supply:

- 1. **separator** (or "auto" for comma detection).
- 2. **coordinator** or none.
- 3. A **spread** of content items (..content).

#### 6.1. Example: Using run-in-join Directly



First item; Second item; Third item, comma triggers semicolon; and/or Final item

You can see that it inserts semicolons (auto detection) and "and/or" before the last item.

# 7. Conclusion

That's your tour of the **run-in** functions! Remember:

- If you want enumerations with (1) or (A) prefixes, use run-in-enum.
- For bullet lists, use run-in-list.
- For term-definition pairs, run-in-terms.
- For verse lines, run-in-verse.
- Or just call **run-in-join** yourself for custom needs.

Explore different **separators** (like [ /~], ", or "; ") and **coordinators** ("and", "or", "none", etc.) to get the exact run-in style your text requires.

Happy Typst-ing!