

Shortcuts Dev Tip

Handy tips and techniques for developing Apple Shortcuts, gained through hands-on experience.

Hua-Ming Huang

Last modified: March 18, 2026

Table of Contents

Keyboard hotkeys (macOS)	1
Tips & tricks to minimize # of actions used	2
Things I wished I'd known earlier	2
Good practices	2
Testing	3
Applications	3

Keyboard hotkeys (macOS)

- Cmd + \square/\square ¹
- Cmd + D ²
- Cmd + Backspace ³
- Cmd + N ⁴
- Cmd + R ⁵
- Cmd + Shift + A ⁶
- Cmd + Shift + C ⁷
- Cmd + Shift + I ⁸
- Cmd + Shift + R ⁹
- Cmd + Shift + S ¹⁰

¹Move action up/down
²Duplicate a shortcut
³Delete a shortcut
⁴Create new shortcut
⁵Run a Shortcut
⁶Add action: "Ask for Input"
⁷Add action: "Comment"
⁸Add action: "If-Otherwise-End If"
⁹Add action: "Repeat"
¹⁰Add action: "Show Result"

Tips & tricks to minimize # of actions used

- When using “Set Dictionary Value” to create a dictionary, you can leave the dictionary field blank on the right instead of adding a separate “Dictionary” action and referencing it.
- When handling nested dictionaries and arrays, use **Dot Notation** (e.g., `myDict.key1.key2` or `myArray.0.key`) to access values directly in one step—instead of chaining multiple “Get Value from Dictionary” actions. This approach makes complex data structures much easier to manage, especially when dealing with data from APIs or JSON.
- When formatting dates, wrap **literal characters**—such as punctuation, 'T', 'Z', or any static text—in **single quotes** to treat them as fixed text. This allows you to build custom date-time formats like `yyyy-MM-dd'T'HH:mm:ss'Z'` in a single step without chaining multiple **Format Date** actions. For example, use `yyyy'_Week_'ww` to combine the year and week number in a single action, rather than using 2 separate **Format Date** actions for `yyyy` and `ww`.

Things I wished I'd known earlier

- Using the ‘Match Text’ action with the regular anchor `^` keyword provides more reliable results compared to using the built-in ‘Begins With’ condition.
- On macOS, you can instantly replace selected text in place—no need to copy and paste manually. Just select the text and trigger your shortcut with a keyboard shortcut. This is especially helpful for repetitive formatting tasks like slugifying or hyperlinking text.

Good practices

- Add a “Stop This Shortcut” action at the end of each shortcut to prevent unintended side effects, especially when dealing with conditional logic or automation.
 - **Why It’s a Good Practice:** * Prevents Unwanted Execution: Ensures that no additional actions run accidentally, especially if the shortcut is triggered by another shortcut or automation. * Improves Debugging: Clearly marks the endpoint, making it easier to understand where execution stops. * Avoids Redundant Processing: If a shortcut continues running unnecessarily, it can consume resources or cause unexpected behavior.
 - **When You Might Skip It:** * If a “Stop and Output” Action Exists: This already terminates the shortcut and returns a value, so an additional stop action isn’t needed. * For Simple Shortcuts: If your shortcut only has a few actions and no branching logic, it may not be necessary.
 - In general, if your shortcut has multiple conditions, loops, or external triggers, adding “Stop This Shortcut”

at the end is a good habit.

Testing

- Use Siri to run a shortcut by simply saying its name. For example, say, ‘Hey Siri, Turn off Color Filters.’

Applications

- You can rename Shortcuts dynamically to display quick information—like battery level, date, or weather—directly in the shortcut’s name. By using the “Rename Shortcut” action, you can update its title based on live data, making your Home Screen or widget area more informative at a glance. It’s a clever way to turn shortcuts into mini status indicators without needing to open an app.