



Operating the Steps Pane: Instructions, Actions and Components

Studio Basics · <https://docs.functionize.com/hc/en-us/articles/43798530775319-Operating-the-Steps-Pane-Instructions-Actions-and-Components>

The middle column of the test editor is the steps pane, and most people only ever read it. It is also where you edit a test directly — search it, select across it, reorder it, and turn part of it into something reusable.

More importantly, it is where Studio shows you **what it understood**. If you only take one thing from this page, take the second section.

Instructions and actions are not the same thing

 **functionize Studio**
Instructions and actions

www.functionize.com

ONE INSTRUCTION, THREE ACTIONS

☒ 0 selected

1

Navigate to Electronics and open the Air Conditioner category.

1.1

Pageinit

Load URL 'https://ecom.functionizeapp.com/.../index.php?route=common/home'

1.2

Click

Click A element with TEXT 'ELECTRONICS'

1.3

Pageinit

Load URL '...?route=product/category&path=108'

This is the most useful thing in the editor. You wrote one sentence; Studio generated three actions and numbered them 1.1, 1.2 and 1.3. When a test does something you did not expect, the gap between your sentence and these actions is where the surprise came from.

THE THREE BULK OPERATIONS

Create component	Turns the selected actions into a reusable component. This is how a sign-in sequence stops being repeated in fourteen tests.
Skip selected	Leaves the actions in place but stops them running. Better than deleting while you are working out whether they are the problem.
Delete selected	Removes them. The instruction stays; its generated actions go.

One sentence becomes several numbered actions - and that is where diagnosis starts

 www.functionize.com

You write an **instruction** — one sentence, in plain language. Studio generates one or more **actions** from it, and those are what actually run against the browser.

The numbering makes the relationship visible. Instruction **1** produces actions **1.1**, **1.2**, **1.3**.

So a single sentence like "Navigate to Electronics and open the Air Conditioner category" becomes three actions: load a URL, click the link whose text is **ELECTRONICS**, load the resulting category URL.

This is where diagnosis starts

When a test does something you did not expect, the gap between your sentence and the generated actions is almost always where the surprise came from. Read the actions before you change the instruction.

The parameters are colored so you can scan them: the element type, the attribute being matched on (`TEXT` , `URL`), and the value. An action that matched on something fragile is visible here long before it fails.

The four controls above the list

 **functionize** **Studio**
Operating the steps pane

www.functionize.com

THE FOUR CONTROLS ABOVE THE LIST



+

≡

1

Navigate to Electronics and open the Air Conditioner category.

...

2

Open the Boltas 2 Ton 3 Star Split AC product.

...

3

Verify the product price is \$1,299.00.

...

Search actions

Filters the list as you type. Useful once a test is long enough that scrolling is the slow part.

Bulk select actions

Checkboxes on every **action**, not just every instruction — and it opens each instruction up so you can see them.

Expand all instructions

Opens every instruction at once, without entering select mode.

Add component

Inserts a saved component — a sequence you have already built somewhere else.

The lime ring marks the action currently selected. Whatever is selected is what the **Data**, **Log** and **Extension** tabs at the bottom are describing.

The four controls above the list, and what each one is for

www.functionize.com

Search actions	Filters the list as you type. Worth reaching for as soon as a test is long enough that scrolling is the slow part.
Bulk select actions	Puts a checkbox on every <i>action</i> , not just every instruction — and opens each instruction so you can see them. This is the fastest way to inspect a whole test.

Expand all instructions	Opens every instruction at once without entering select mode. Use this when you want to read rather than edit.
Add component	Inserts a saved component — a sequence you built somewhere else.

The lime ring marks the currently selected action. Whatever is selected is what the **Data**, **Log** and **Extension** tabs at the bottom are describing — see [Understanding a Test Run and Its Results in Functionize Studio](#).

Three things you can do to a selection

Once you are in bulk select mode, three operations appear at the top of the pane.

Create component	Turns the selected actions into a reusable component. This is how a sign-in sequence stops being repeated in fourteen tests, and it is the single highest-value habit in this article.
... and they are account-wide	Components are shared across the whole account. The Add component list shows every component anyone has built, in any project — so a component called <i>login</i> is a name your whole organization can only use once meaningfully. Qualify it: <i>storefront — guest sign-in</i> . See Naming, Tagging and Not Drowning .
Skip selected	Leaves the actions in place but stops them running. Better than deleting while you are still working out whether they are the problem.
Delete selected	Removes them. The instruction stays; its generated actions go.

Skip before you delete. A skipped action is a hypothesis you can reverse in one click. A deleted one is a hypothesis you have to rebuild.

Renaming, deleting and reordering one instruction

 **functionize Studio**
Renaming, deleting and reordering

www.functionize.com

WHAT EACH INSTRUCTION CAN DO

3 Verify the product price is \$1,299.00. ...

Rename

Delete

Rename, and why it matters

Renaming changes the **instruction** — the sentence you wrote. It does not regenerate the actions underneath. If you need different behavior, ask the agent for it instead; if you only need a clearer label, rename.

REORDERING

Every instruction has a drag handle. Moving one moves its generated actions with it. Worth knowing: reordering changes what the test *does*, so a test that passed before a reorder has not been verified after it — run it.

What you can do to a single instruction

www.functionize.com

Every instruction carries an options menu with two items, **Rename** and **Delete**, and a drag handle for reordering.

- **Rename changes the sentence, not the behavior.** It does not regenerate the actions underneath. If you need the test to do something different, ask the agent for it instead — see [Exercise 4 - Change a Test by Asking](#). If you only need a clearer label, rename.
- **Reordering moves the generated actions too.** An instruction and its actions travel together.
- **A reordered test is an unverified test.** Passing before a reorder tells you nothing about after it. Run it.

When to edit here, and when to ask the agent

The two ways of changing a test are not interchangeable, and choosing wrongly is a common source of wasted effort.

Edit in the steps pane	Structural changes that need no new understanding of the application: reordering, removing a step you no longer want, skipping something while you diagnose, renaming for clarity, pulling a sequence out into a component.
Ask the agent	Anything requiring the application to be understood again: a different assertion, a changed flow, a new field, a different outcome. The agent regenerates the actions; editing by hand does not.

A useful rule: if your change can be described as *moving or removing*, do it in the pane. If it is *proving something different*, ask.

Where to go next

- [Understanding a Test Run and Its Results in Functionize Studio](#) — the evidence tabs underneath the pane, including what the Log tab tells you.
- [Diagnosing a Failed Test in Functionize Studio](#) — using all of this when something goes red.
- [Writing Prompts That Produce Durable Tests](#) — how to write the instruction so the generated actions age well.
- [Exercise 4 - Change a Test by Asking](#) — ten minutes of practice on the demo store.