



data  
iku

KNOWLEDGE

[Home](#) » [Data Preparation](#) » [Prepare Recipe](#)

# Concept | Prepare recipe

Watch the video



The [Prepare recipe](#) is a visual recipe in Dataiku that allows you to create data cleansing, normalization, and enrichment scripts in a visual and interactive way.

## Adding transformation steps to the script

To prepare your data, you must add steps to the recipe script.

## Using the processor library

An essential advantage of the Prepare recipe is its library of around 100 data processors. Most [processors](#) are designed to handle one specific task, such as filtering rows, rounding numbers, extracting regular expressions, concatenating or splitting columns, and much more.

The screenshot displays the Dataiku Prepare Recipe interface. The top navigation bar includes 'Prepare Recipe Demo', 'Recipes', and 'Dataiku Academy'. The main area is divided into 'Script' and 'Design Sample' tabs. The 'Script' tab shows 'no step' and '10000 rows 7 cols'. The 'Design Sample' tab shows 'Viewing design sample' with '10000 rows, 7 cols'. A table of data is visible with columns: order\_date, pages\_visited, order\_id, customer\_id, tshirt\_category, tshirt\_price, and tshirt\_quantity. A 'Processors library' panel is open, showing a list of 91 processors. The 'Reshaping' category is selected, and the 'Find and replace' processor is highlighted. The 'Find and replace' processor is described as 'Find and replace' and 'Split column'. The 'Transform string' processor is described as 'Transform string' and 'Formula'. The 'Extract with regular expression' processor is described as 'Extract with regular expression' and 'Concatenate columns'. The 'Simplify text' processor is described as 'Simplify text' and 'Tokenize text'. The 'Extract ngrams' processor is described as 'Extract ngrams' and 'Extract numbers'. The 'Negate boolean value' processor is described as 'Negate boolean value'.

order_date	pages_visited	order_id	customer_id	tshirt_category	tshirt_price	tshirt_quantity
2016/09/04	9	HTS-0002	038040	White T-Shirt M	20.0	1
2014/11/14	11	HTS-0001	801797	White T-Shirt M	20.0	1
2017/03/06	10	HTS-0003	038040	White T-Shirt E	18.0	2

Processors empower you to perform a huge variety and combination of tasks. One processor, for example, is a [Formula language](#), similar to what you might find in a spreadsheet.



Feedback

which you can use to create new columns from those already present, drawing on a range of built-in functions.

Another processor even lets you create a [Python function](#) for each row.

In addition to directly adding steps from the processor library, you can add steps to the script in a number of other ways.

## Using the column context menu

In the column context menu, Dataiku will suggest steps to add based on the column's meaning.

For example, Dataiku will suggest to parse date columns, or remove rows with invalid values according to the column meaning. For a text column, it will suggest string transformations, such as converting to lowercase.

The screenshot shows the Dataiku interface for a script named 'compute\_orders\_prepared'. The interface is divided into several sections: a left sidebar with a 'Script' tab showing 7 steps, a 'Sample settings' section for the first 10,000 rows, and a main 'Script output' section. The 'Script output' section displays a table with 10,000 rows and 6 columns: 'order\_date', 'order\_id', 'pages\_visited', 'customer\_id', 'tshirt\_category', and 'total'. A context menu is open over the 'order\_date' column, showing various actions such as 'Delete', 'Rename', 'Move', 'Analyze...', 'Edit column details...', 'Convert to lowercase', 'Convert to uppercase', 'Transform string', 'More actions', 'Filter', 'Sort', and 'Conditional Formatting'. The 'order\_date' column is highlighted as a 'Date' type. The 'tshirt\_category' column is highlighted as a 'Text' type. The 'total' column is highlighted as a 'Double' type. The 'Script' section on the left shows a list of steps: 'Move order\_id before pages\_visited', 'Replace 6 values in tshirt\_category', 'Parse date in order\_date', 'Remove order\_date', 'Rename column 'order\_date\_parsed' to 'order\_date'', and 'Create column total with formula tshirt\_price \*'. A 'RUN' button is visible at the bottom of the script section.

order_date	order_id	pages_visited	customer_id	tshirt_category	total
2016-09-04T00:00...				White T-Shirt M	
2014-11-14T00:00...				White T-Shirt M	
2017-02-26T00:00...				White T-Shirt F	
2013-12-01T00:00...				White T-Shirt F	
2015-10-22T00:00...				White T-Shirt M	
2016-01-15T00:00...				Black T-Shirt F	
2014-11-25T00:00...				Hoodie	
2014-11-01T00:00...				Black T-Shirt M	
2013-11-03T00:00...	HTS-494332-0001	10	494332	White T-Shirt F	
2013-10-02T00:00...	HTS-463794-0001	14	463794	Tennis Shirt	

## Using the Analyze window

Another method to add steps to the script is through the Analyze window.

Within a Prepare recipe, the Analyze window can guide data preparation, for example merging categorical values.



The screenshot shows the Dataiku interface for a script named 'compute\_orders\_prepared'. The 'Script' tab is active, displaying a list of steps on the left and a step preview on the right. The 'Sample settings' box is highlighted, showing 'First 10,000 rows'. The 'Perform to\_lower on tshirt\_category' step is selected, and its settings are shown on the left. The step preview on the right shows a table with 10,000 rows and 6 columns. The 'tshirt\_category' column is highlighted in the preview table.

order_date	order_id	pages_visited	customer_id	tshirt_category	total
2016-09-04T00:00...	HTS-038040-0002	9	038040	white t-shirt m	
2014-11-14T00:00...	HTS-801797-0001	11	801797	white t-shirt m	
2017-02-26T00:00...	HTS-vft1eu-0003	10	vft1eu	white t-shirt f	
2013-12-01T00:00...	HTS-914324-0001	10	914324	white t-shirt f	
2015-10-22T00:00...	HTS-88ua9r-0001	12	88ua9r	white t-shirt m	
2016-01-15T00:00...	HTS-061311-0003	9	061311	black t-shirt f	
2014-11-25T00:00...	HTS-479441-0001	6	479441	hoodie	
2014-11-01T00:00...	HTS-352809-0001	10	352809	black t-shirt m	
2013-11-03T00:00...	HTS-494332-0001	10	494332	white t-shirt f	
2013-10-02T00:00...	HTS-463794-0001	14	463794	tennis shirt	
2013-07-23T00:00...	HTS-132885-0001	17	132885	white t-shirt m	

Notice that steps in the script constitute a list of instructions. These instructions are not immediately applied to the dataset itself.

For example, adding a **Delete Column** step removes that column from the step preview, but it does not actually delete the column in the dataset, as it would in a spreadsheet.

Only when you choose to actually run the recipe will Dataiku execute the instructions on the full input dataset, and thereby produce a new output dataset. The original input dataset always remains.

## Managing the script

If a script starts to grow in complexity, a number of features can help you manage them. You can:

- Disable steps.
- Organize individual steps into groups of steps.
- Add colors and comments to steps in order to send reminders to yourself and colleagues.
- [Copy and paste steps](#) within the same recipe or to another recipe, even if that recipe is in another project or another Dataiku instance.

The screenshot shows the Dataiku Prepare Recipe Demo interface. The 'Script' tab is active, displaying a list of steps on the left and a table of 7597 rows on the right. A red box highlights the 'Copy this step' option in the context menu for the 'Remove order\_id' step. The table columns include order\_date, order\_date\_parsed, order\_id, pages\_visited, customer\_id, tshirt\_category, tshirt\_price, and tshirt\_quantity.

## What's next?

In this article, you learned how to use the Prepare recipe for data cleansing, normalization, and enrichment.

### Note

Instead of building recipes directly in the Flow, when your workflow is in production, to avoid disturbing it, you can use a [visual analysis in the Lab](#) for experimental work.

Continue getting to know the basics of Dataiku by learning about [date handling](#).

### Tip

You can find this content (and more) by registering for the Dataiku Academy course, [Visual Recipes](#). When ready, challenge yourself to earn a [certification](#)!