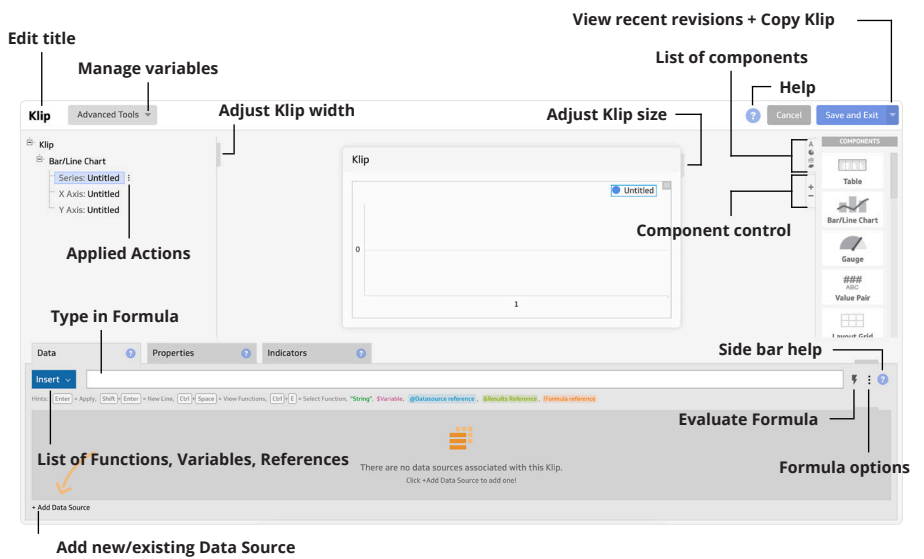


Klipfolio Cheat Sheet

Keyboard Shortcuts

Evaluate Formula	⌘ / Ctrl + A
Select only the function the cursor is on	⌘ / Ctrl + E
Undo	⌘ / Ctrl + Z
Create new line in formula bar	Shift + Enter
Save Klip	ALT + S
Apply actions	Enter
Get to the beginning of the line (HOME)	fn + →
Get to the end of the line (END)	fn + ←
Start a new line	Shift + Enter
Cut	⌘ / Ctrl + X
Copy	⌘ / Ctrl + C
Paste	⌘ / Ctrl + V

Klip Builder



Applied Actions

Filter

To apply a **Filter**, click the menu next to the highlighted row in the component tree and select **Filter** or right-click the series in a chart or a column in a table.

Group

To apply a **Group**, right-click the sub-component, click the menu next to it in the component tree or check **Group repeating labels** in the Properties panel.

Aggregate

To apply an **Aggregation**, click the menu next to the sub-component in the component tree and select **Aggregation**

Sort

To apply a **Sort**, click the menu next to the sub-component in the component tree and select **Sort** or right-click the data in a component. You can also use the **Sort** menu on the Properties panel.

Ways to share your Klips:

- User and group sharing
- Published links
- Automated email snapshots
- PDF reports
- Office wallboards
- Slack
- Embed Klips on your website or private intranet
- Add your comments to a Klip

Klip Components

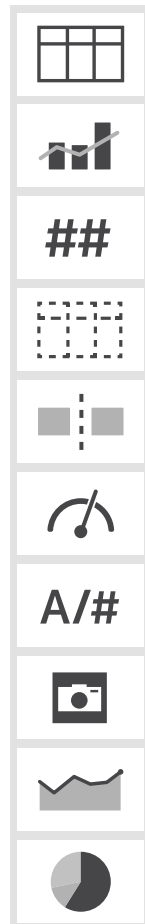


Table: displays information in columns and rows, ideal for showing tabular data.

Bar/Line Chart: displays a comparison of values in either a bar or line chart form.

Value Pair: includes a primary and secondary value.

Layout Grid: is divided into cells by horizontal and vertical grid lines.

Separator: is a line that's used to create visual separation between components in a Klip.

Gauge: displays a value that's relative to a predetermined target.

Label: is ideal for creating section headers and adding descriptions to Klips.

Image: component displays an image that you specify by URL.

Sparkline: includes a line or bar graph that displays trends over time and/or the current value of a KPI.

Pie Chart: categorical data divided into sections, so you can see each section's value in comparison to the whole.

Scatter Chart: shows a visual comparison of two sets of values in a chart.

Bubble Chart: shows the correlation between three sets of data.

Map: enables you to display values for regions on a map.

Inline Frame: enables you to display content within an IFrame.

User Input Control: works with variables (that you create) and enables you to: search for values using a text field, select values to display from a drop-down list, or select a date range using a date picker.

Funnel Chart: enables you to track your data as it moves through a process.

Button: is used to control the action of variables.

Pictograph: tells a story about your data using images or symbols, displaying data in a visual way.

HTML Template: enables you to create a custom Klip using any combination of HTML, Javascript, and CSS.

News: displays a stream of headlines from an online news source and is ideal for monitoring RSS feeds.

What is a component? Components are the building blocks for Klips and determine the way your Klips look and behave. Choosing the right components helps you represent and interpret your data effectively.

Top Klipfolio Functions

Lookup: to correlate data between two data sources.

`LOOKUP(input, keys, results)`

input: A list of 1 or more items.

keys: A list of 1 or more items that intersects with the input list.

results: A list of 1 or more items typically from the same source as keys. Must contain the same number of items as keys.

Date: to convert dates specified in a given format into Unix time format.

`DATE(dates, format, [timezone])`

dates: A list of 1 or more date/time values.

format: The date format of the values in the dates parameter.

timezone: The time zone.

Count: to return a count of all non-blank (numeric and text) items in data.

`COUNT(data)`

data: A list of 1 or more items.

Array: to join together single values and return them as a single list of data in the order provided.

`ARRAY(data)`

data: The values to join together including strings or columns of data from different data sources.

Contains: to test each value in the haystack parameter to see if it contains the value in the needle parameter.

`CONTAINS(haystack, needle)`

haystack: A list of 1 or more items.

needle: The case-sensitive item to search for in the haystack parameter.

Group: to group data into unique instances and hide duplicate values. The results are returned in alphabetical order.

`GROUP(values)`

values: A list of 0 or more values.

Select: to select values from a list according to specified criteria.

`SELECT(data, condition)`

data: A list of 1 or more items.

condition: A list of true and false values, typically a formula which combines 1 or more Logic functions. Data and condition must have the same number of items.

Groupby: to return values based on a specified aggregation method so that the unique values align with a parallel column.

`GROUPBY(values, measure, [method])`

values: A list of 0 or more values.

measure: A list of 0 or more values. Values and measure must have the same number of items.

method: [optional] The aggregation method to use when grouping. The default method is SUM.

Slice: to return the subset of values between the start and end positions. If start and end parameters are not specified the first row is removed.

`SLICE(values, [start], [end])`

values: A list of 0 or more items.

start: [optional] Indicates the number of items sliced off the top.

end: [optional] Indicates the position of the last item to be returned.

Countif: to test each value in a condition and count the true results of the condition.

`COUNTIF(condition)`

condition: A list of true and false values.

DATE_CONVERT: to convert values from one date format to another date format.

`DATE_CONVERT(values, format in, format out)`

values: A list of 1 or more values to test.

format in: The current date format of the date/time values.

format out: The date/time format you want to convert the dates to.

If: to test a condition and specify the result of the condition if it evaluates to true or false.

`IF(condition, if true, if false)`

condition: A list of 1 or more values to test.

if true: Data returned if the condition is true.

if false: Data returned if the condition is false.

Switch: to switch a value to another value based on whether the case is evaluated to be true. If no match is found null is returned.

`SWITCH(data, case, values)`

data: A list of 1 or more values.

case: The condition to be evaluated as either true or false.

values: The value to be returned if the case is true.

Concat: to join two or more values into one text string.

`CONCAT(data)`

data: The value or values to join together.

Time: to convert a date/time duration, specified as a combination of days, hours, minutes and seconds, to number of seconds.

`TIME(values, format)`

values: A list of 1 or more date/time durations.

format: The format of the date/time duration.

Substitute: replaces a set of characters with another set of characters in a text string.

`SUBSTITUTE(text, old text, new text, [occurrence])`

text: The values to be manipulated.

old text: The values that will be replaced.

new text: The values to replace the values in old text.

occurrence: [optional] Indicates the instance that will be replaced.

Mapflat/Map: to repeat a formula or datasource reference (specified by the **formula** parameter) for each value in the **values** parameter.

`MAPFLAT(values, variable name, formula)`

values: A list of 1 or more values.

variable name: Name of the variable used in the formula parameter, typed as a literal string.

formula: Formula which uses variable name (entered as a \$variable) and is executed for each item in values. If the formula returns more than 1 item, only the first value is returned.

Between: to return true or false if a value is numerically between a (inclusive) start and end.

`BETWEEN(values, start, end)`

values: A list of 1 or more numeric items.

start: Numeric start of range.

end: Numeric end of range