

# Filtering and Building

Students learn about functions that work with tables, allowing them to filter and build columns

<b>Lesson Goals</b>	<p>Students will be able to...</p> <ul style="list-style-type: none"><li>• filter tables according to different criteria, using helper functions</li><li>• add a column to a table using helper functions</li></ul>
<b>Student-facing Lesson Goals</b>	<ul style="list-style-type: none"><li>• Let's learn how transform one table into another.</li></ul>
<b>Prerequisites</b>	<ul style="list-style-type: none"><li>• <a href="#">Simple Data Types</a></li><li>• <a href="#">Contracts</a></li><li>• <a href="#">Contracts: Making Tables and Displays</a></li><li>• <a href="#">Functions: Contracts, Examples &amp; Definitions</a></li></ul>
<b>Materials</b>	<ul style="list-style-type: none"><li>• <a href="#">PDF of all Handouts and Page</a></li><li>• <a href="#">Table Functions Starter File</a></li><li>• <a href="#">Animals Starter File</a></li><li>• <a href="#">Lesson Slides</a></li><li>• <a href="#">Printable Lesson Plan</a> (a PDF of this web page)</li></ul>
<b>Supplemental Materials</b>	<ul style="list-style-type: none"><li>• <a href="#">Additional Printable Pages for Scaffolding and Practice</a></li></ul>

## Overview

Students practice carefully reading the contents of a starter file to review and gain comfort with row-consuming functions.

## Launch



- Load the [Table Functions Starter File](#), go to the File menu, and click "Save a Copy".
- Hit "Run," then type `shapes-table` in the Interactions Area.
- The way this file defines a table is different from what we've encountered before. How is it different?
  - Pyret also allows us to define Tables *in code*, rather than using a spreadsheet. This file defines the `shapes-table` one row at a time.

## Investigate



With your partner, take 5 minutes to complete [Reading Row and Function Definitions](#).

Once students have completed the activity, follow up to confirm understanding.



- Who can explain what `is-red` does?
  - *It looks up the color of the row, and checks if it is equal to the string "red".*
- Who can explain what `is-polygon` does?
  - *It looks up the value in the polygon column.*
- Who can explain what `is-triangle` does?
  - *It checks if the value in the name column contains "triangle".*

Challenge students to use terminology like "looks up the value in the X column" when describing a lookup.

## Synthesize

- Were you able to guess what the functions did before testing them out? What was your strategy for making an educated guess?

- What is the relationship between a *row definition* and a *row-consuming function*?

# Filtering Rows, Building Columns

35 minutes

## Overview

Students learn to filter the rows and add new columns to a table, using functions they can define themselves.

## Launch

Select 6-8 students, each of whom will represent a unique Row of a dataset called “Students.” Arrange them in a line at the front of the room.

Sometimes we want to **filter the rows** of a table by some criteria. In our animals dataset, for example, we might want to filter the table to show only the rows for dogs, or only the rows for cats older than 5. What are some ways to filter our table of students?



- I need one volunteer who will play the role of `filter`, and another who will be their trusty helper function.
- The helper chooses one [Function Card](#). They’ll read it carefully, making sure they understand what it will do for each “Row” (student!), but **keeping it a secret**. (Choose your students and cards carefully! Some cards only work if you have at least one student with glasses, red hair, etc.)
- `filter` walks from one student to the next, pointing to each one and asking “keep ‘em?”.
- The helper will answer with whatever the function on the card would produce for that student: `true` or `false`
- If the helper replies “true”, `filter` asks that student to take a step forward
- If the helper says “false”, `filter` asks them to take a step back.

Here’s how that might look if `filter` chose a function card `has-glasses`.

- `filter` stands in front of Student 1. “Keep ‘em?”
- The helper checks if they are wearing glasses, and says “true”
- `filter` to Student 1 (who wears glasses): Step forward. (Student 1 steps forward.)
- `filter` stands in front of Student 2. “Keep ‘em?”
- The helper checks if they are wearing glasses, and says “false”
- `filter` to Student 2 (who does not wear glasses): Step back. (Student 2 steps back.)

Have your helper function and `filter` work together to act out one round of this. Ask the `filter` if they can guess what’s on the helper function’s card, then ask the rest of the class. Give a *different*

card to a *different* helper, and repeat the exercise.

## Investigate

Pyret Tables have their own functions for filtering tables - and more! Let's explore some of these.



Take 10 minutes to complete [Exploring Table Functions](#)

Confirm that students understand the following key ideas:

- `filter` consumes a Table and a *function that consumes Rows and produces Booleans*. It produces a new table containing only rows for which the function returns `true`.
- `build-column` consumes a Table, the name of a new column, and a *function that consumes Rows*. It produces a new table with that new column, and fills in the values by applying the function to every Row.



- How would you filter and build using the Animals Dataset?
- For practice, complete [What Table Do We Get?](#).
- *Optional:* Complete [The Design Recipe: is-cat / is-male](#) and [The Design Recipe: is-young / is-fixed](#) to practice using the Design Recipe to write filter functions for the [Animals Starter File](#).

## Common Misconceptions

**Students often think that these functions *change* the table!** In Pyret, all table functions produce a *brand new table*. If we want to save that table and use it later, we need to define it. For example:  
`cats = filter(animals-table, is-cat).`

## Synthesize

Using Table Functions is a *huge* upgrade in our ability to analyze data!

- If the shelter is purchasing food for older cats, what filter would we write to determine how many cats to buy for?
- A dataset from Europe might list everything in metric (centimeters, kilograms, etc), so we could build a column to convert that to imperial units (inches, pounds, etc).
- A dataset about sports teams might include columns for how many games each team won and how many they lost, but it's more useful to build a column to see *what percentage of games* those teams won.

- What columns might you want to add to your dataset?
- What filters might be interesting to apply to your dataset?