# Method Chaining

Lesson Pathway, Standards and Practices**Lessons**

Prerequisites only (★)

* [Introduction to Computational Data Science](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/ds-intro/index.shtml)
* [Simple Data Types](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/simple-datatypes/index.shtml)
* [Contracts](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/contracts/index.shtml)
* [Displaying Categorical Data](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/displaying-categorical-data/index.shtml)
* [Data Displays and Lookups](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/displays-and-lookups/index.shtml)
* [Table Methods](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/table-methods/index.shtml)
* [Defining Functions](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/defining-functions/index.shtml)
* [Defining Table Functions](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/defining-table-functions/index.shtml)
* [Method Chaining](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/method-chaining/index.shtml)
* [If-Expressions](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/if-expressions/index.shtml)
* [Randomness and Sample Size](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/random-samples/index.shtml)
* [Grouped Samples](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/grouped-samples/index.shtml)
* [Choosing Your Dataset](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/choosing-your-dataset/index.shtml)
* [Histograms](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/histograms/index.shtml)
* [Visualizing the “Shape” of Data](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/histograms2/index.shtml)
* [Measures of Center](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/measures-of-center/index.shtml)
* [Spread of a dataset](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/measures-of-spread/index.shtml)
* [Checking Your Work](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/checking-your-work/index.shtml)
* [Scatter Plots](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/scatter-plots/index.shtml)
* [Correlations](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/correlations/index.shtml)
* [Linear Regression](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/linear-regression/index.shtml)
* [Ethics and Privacy](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/ethics-and-privacy/index.shtml)
* [Threats to Validity](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/threats-to-validity/index.shtml)

| **Standards** (click one) CSTAK12CSOklahoma |
| --- |
| CSTA Standards  1B-AP-10  Create programs that include sequences, events, loops, and conditionals.  2-AP-13  Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs  2-AP-17  Systematically test and refine programs using a range of test cases  3A-AP-17  Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.  3A-AP-18  Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.  K-12CS Standards  6-8.Algorithms and Programming.Control  Programmers select and combine control structures, such as loops, event handlers, and conditionals, to create more complex program behavior.  9-12.Algorithms and Programming.Control  Programmers consider tradeoffs related to implementation, readability, and program performance when selecting and combining control structures.  9-12.Algorithms and Programming.Modularity  Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose. These modules can be procedures within a program; combinations of data and procedures; or independent, but interrelated, programs. Modules allow for better management of complex tasks.  P3  Recognizing and Defining Computational Problems  Oklahoma Standards  OK.A1.D.2.1  Select and apply counting procedures, such as the multiplication and addition principles and tree diagrams, to determine the size of a sample space (the number of possible outcomes) and to calculate probabilities.  OK.L1.AP.M.01  Break down a solution into procedures using systematic analysis and design.  OK.L1.AP.M.02  Create computational artifacts by systematically organizing, manipulating and/or processing data. |

**Textbook Alignment**

* IM.Alg1.4.3

**Practices**

* ****
* ****

Students learn how to chain Methods together, and define more sophisticated subsets.

| **Lesson Goals** | Students will be able to…​   * Use method chaining to write more sophisticated analyses using less code * Identify bugs introduced by chaining methods in the wrong order |
| --- | --- |
| **Student-facing Lesson Goals** | * Let’s practice writing functions and combining methods together. |
| **Materials** | * [Lesson Slides](https://docs.google.com/presentation/d/1jBoQJfEf8-SuAEEu3Y-u_aYdEEjAQEjOM4G-kv6I9y4/) * [Animals Starter File](https://code.pyret.org/editor#share=1ZupMVPWvVUOM0HCWyA7cRBghSLKxPWv1) * [The Design Recipe](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/method-chaining/pages/design-recipe-1.html) * [The Design Recipe](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/method-chaining/pages/design-recipe-2.html) * [Chaining Methods](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/method-chaining/pages/chaining-methods.html) * [Chaining Methods 2: Order Matters](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/method-chaining/pages/chaining-methods-order-matters.html) * [Table Transformations with Method Chaining](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/method-chaining/pages/chaining-methods-table-transformations.html) |
| **Preparation** | * Make sure all materials have been gathered * Decide how students will be grouped in pairs * Computer for each student (or pair), with access to the internet. All students should log into [CPO](https://code.pyret.org) and open the "Animals Starter File" they saved from the prior lesson. If they don’t have the file, they can [open a new one](https://code.pyret.org/editor#share=1ZupMVPWvVUOM0HCWyA7cRBghSLKxPWv1) * [Student workbook](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science//workbook/workbook.pdf), and something to write with |
| **Language Table** | | **Types** | Functions | Values | | --- | --- | --- | | **Number** | +, -, \*, /, num-sqrt | 4, -1.2, 2/3, pi | | **String** | string-length, string-repeat, string-contains | "hello", "91" | | **Boolean** | <, <>, <=, >=, <, >, ==, <>, >= | true, false | | **Image** | star, triangle, circle, square, rhombus, ellipse, regular-polygon, radial-star, bar-chart, pie-chart, box-plot, scatter-plot, bar-chart-summarized, pie-chart-summarized | 🔵🔺🔶 | | **Table** | .row-n, .order-by, .filter, .build-column |  | |

| Design Recipe Practice | 25 minutes |
| --- | --- |

### **Overview**

Students practice more of what they learned in the previous lesson, applying the Design Recipe to make table functions that operate on rows of the Animals Dataset. These become the basis of the chaining activity that follows.

### **Launch**

The Design Recipe is a powerful tool for solving problems by writing functions. It’s important for this to be like second nature, so let’s get some more practice using it!

### **Investigate**

Define the Compute functions on [The Design Recipe](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/method-chaining/pages/design-recipe-1.html) and [The Design Recipe](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/method-chaining/pages/design-recipe-2.html).

| **Optional: Combining Booleans**  Suppose we want to build a table of Animals that are fixed *and* old, or a table of animals that are cats *or* dogs?  By using the and and or operators, we can *combine* boolean tests , as in: (1 > 2) and ("a" == "b") . This is handy for more complex programs! For example, we might want to ask if a character in a video game has run out of health points *and* if they have any more lives. We might want to know if someone’s ZIP Code puts them in Texas or New Mexico. When you go out to eat at a restaurant, you might ask what items on the menu have meat and cheese.  For many of the situations where you might use and, there’s actually a much more powerful mechanism you can use, called "Method Chaining"! |
| --- |

### **Synthesize**

Did students find themselves getting faster at using the Design Recipe? Can students share any patterns they noticed, or shortcuts they used?

| Chaining | 25 minutes |
| --- | --- |

### **Overview**

Students learn how to compose multiple table operations (sorting, filtering, building) on the same table - a technique called "chaining".

### **Launch**

Now that we are doing more sophisticated analyses, we might find ourselves writing the following code:

# get a table with the nametags of all the fixed animals, ordered by species  
with-labels = animals-table.build-column("labels", nametag)  
fixed-with-labels = with-nametags.filter(is-fixed)  
result = fixed-with-labels.order-by("species", true)

That’s a lot of code, and it also requires us to come up with names for each intermediate step! Pyret allows table methods to be *chained together*, so that we can build, filter *and* order a Table in one shot. For example:

# get a table with the nametags of all the fixed animals, ordered by species  
result = animals-table.build-column("labels", nametag).filter(is-fixed).order-by("species", true)

This code takes the animals-table, and builds a new column. According to our Contracts Page, .build-column produces a new Table, and that’s the Table whose .filter method we use. That method produces *yet another Table*, and we call that Table’s order-by method. The Table that comes back from that is our final result.

| **Teaching Tip**  Use different color markers to draw *nested boxes* around each part of the expression, showing where each Table came from. |
| --- |

It can be difficult to read code that has lots of method calls chained together, so we can add a line-break before each “.” to make it more readable. Here’s the exact same code, written with each method on its own line:

# get a table with the nametags of all the fixed animals, order by species  
animals-table  
 .build-column("label", nametag)  
 .filter(is-fixed)  
 .order-by("species", true)

**Order matters: Build, Filter, Order.**

Suppose we want to build a column and then use it to filter our table. If we use the methods in the wrong order (trying to filter by a column that doesn’t exist yet), we might wind up crashing the program. Even worse, the program might work, but produce results that are incorrect!

### **Investigate**

**When chaining methods , it’s important to build first, then filter, and then order.**

How well do you know your table methods? Complete [Chaining Methods](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/method-chaining/pages/chaining-methods.html) and [Chaining Methods 2: Order Matters](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/method-chaining/pages/chaining-methods-order-matters.html) in your Student Workbook to find out.

### **Synthesize**

As our analysis gets more complex, chaining methods is a great way to re-use work we’ve already done. And less duplicate work means a smaller chance of bugs. Composing operations is a powerful way to work, so it’s critical to think carefully when we use it!

## [**🔗**](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/method-chaining/#_additional_exercises)**Additional Exercises**

* [Table Transformations with Method Chaining](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/method-chaining/pages/chaining-methods-table-transformations.html)

*These materials were developed partly through support of the National Science Foundation, (awards 1042210, 1535276, 1648684, and 1738598).  Bootstrap:Data Science by the* [*Bootstrap Community*](https://bootstrapworld.org/community/index.shtml) *is licensed under a* [*Creative Commons 4.0 Unported License*](https://creativecommons.org/licenses/by-nc-nd/4.0/)*. This license does not grant permission to run training or professional development. Offering training or professional development with materials substantially derived from Bootstrap must be approved in writing by a Bootstrap Director. Permissions beyond the scope of this license, such as to run training, may be available by contacting* [*contact@BootstrapWorld.org*](https://www.bootstrapworld.org/materials/spring2021/en-us/courses/data-science/lessons/method-chaining/mailto:contact@BootstrapWorld.org)*.*

Last updated 2021-10-12 13:37:29 -0400