Python Data Essentials: Data Structures

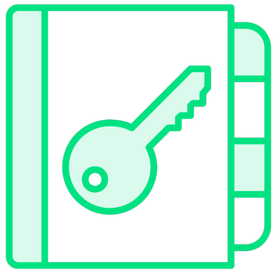
Working with Lists

Mihaela Danci
Data Analyst

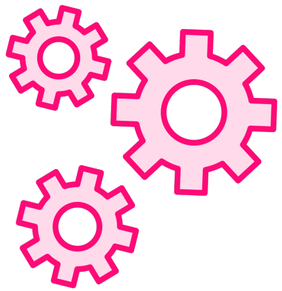
linkedin.com/in/mihaela-danci/

Course Overview

Store



Access



Manipulate

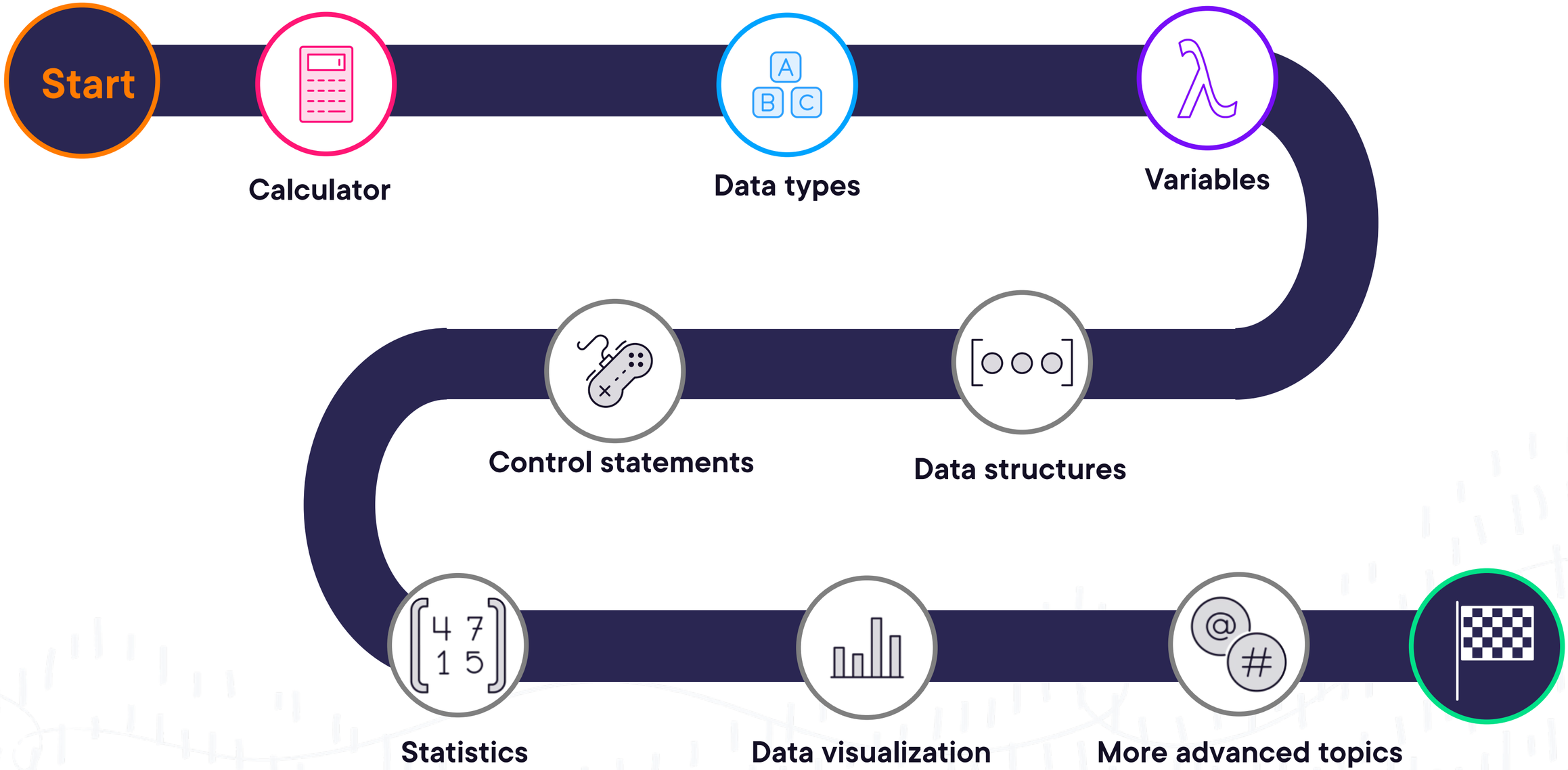
Learning Python is a journey.

This is Tom! He works at Globomantics

- Increase supply chain efficiency

- Which products are bestsellers?
- Which products are low in stock?
- Which products need discounting based on their stocks?

Learning Python



Storing Information

```
product = ”apple”
quantity = 2
price = 0.9
```

Storing Information

```
product = "apple"
quantity = 2
price = 0.9
```

```
product2 = "orange"
quantity2 = 4
price2 = 1.2
```

```
product3 = "lemon"
quantity3 = 6
price3 = 0.4
```

Data structures store a collection of related data so that we can access the data efficiently.

Data Structures

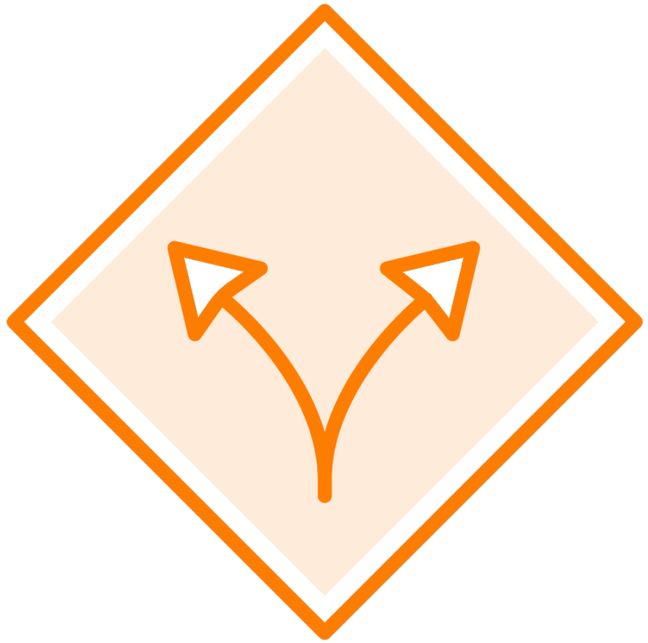
Lists

Tuples

Sets

Dictionaries

Data Structures



Inserting elements

Retrieving elements

Built-in functions

Overview

Lists

- Creating
- Subsetting
- Manipulating

Python Installation

Jupyter Notebook

List Anatomy

Lists

products = [“mango”, “orange”, “lemon”]

Lists

products = [“mango”, “orange”, “lemon”]

Lists

products = [“mango”, “orange”, “lemon”]

Lists

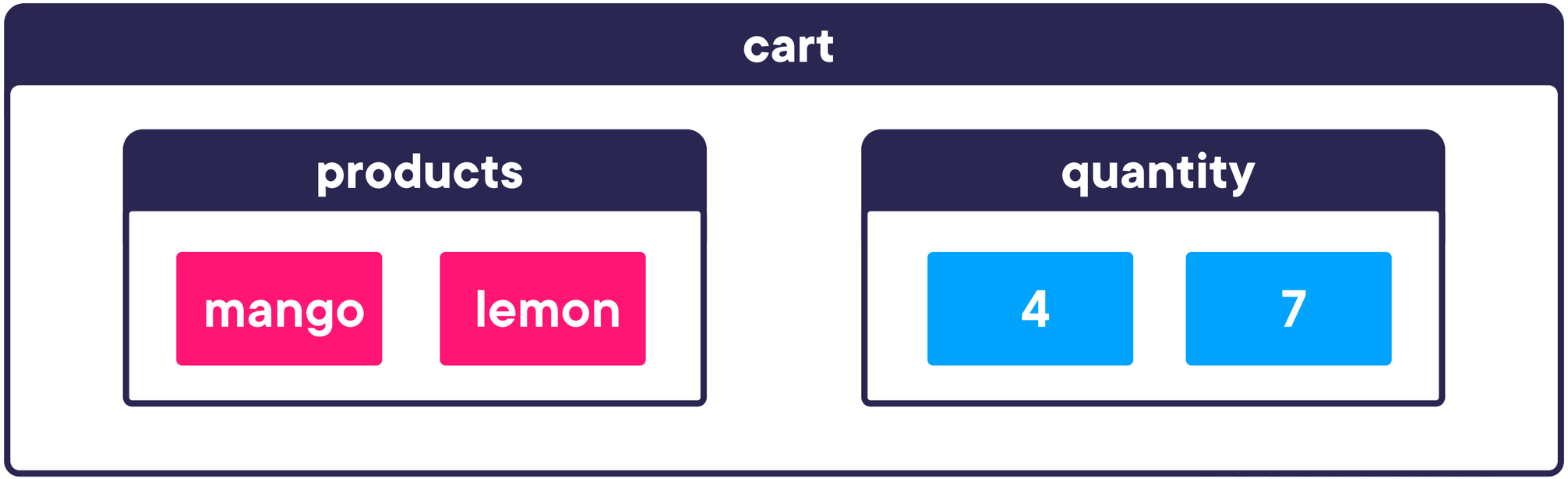
products = [“mango”, “orange”, “lemon”]

Lists

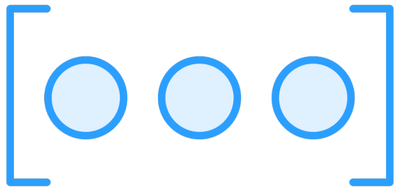
Lists

Lists

Nested Lists

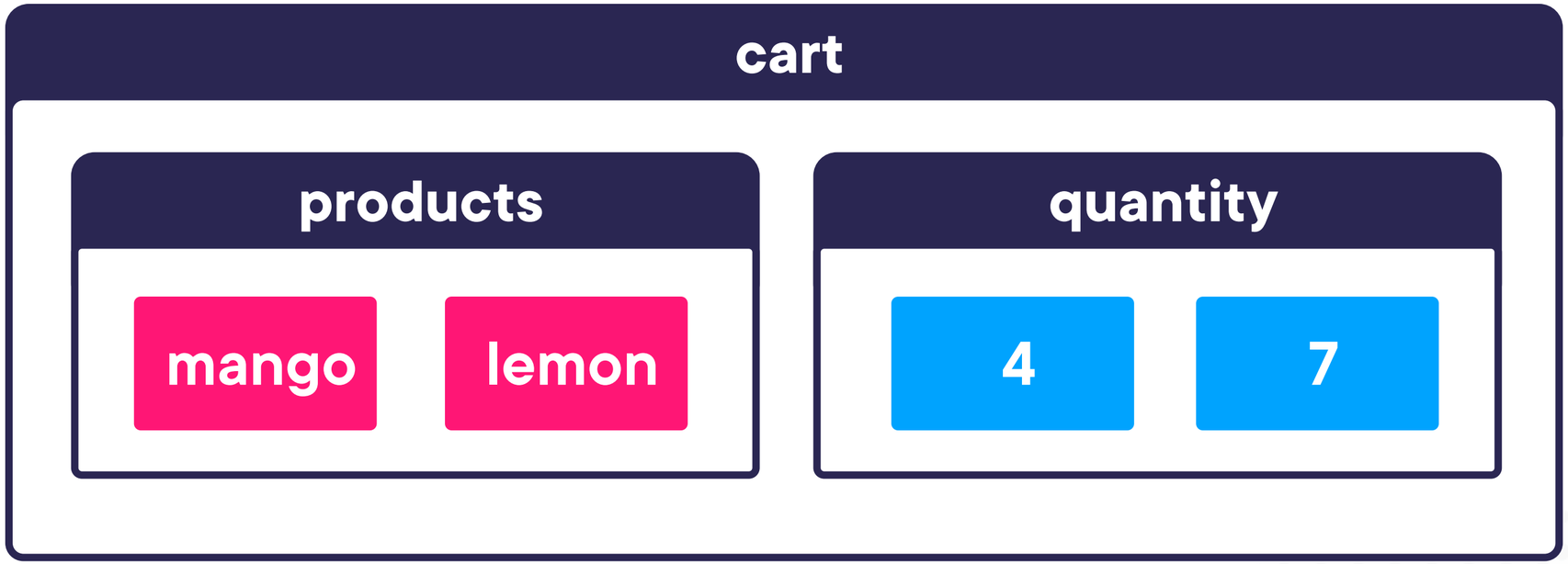


List Properties

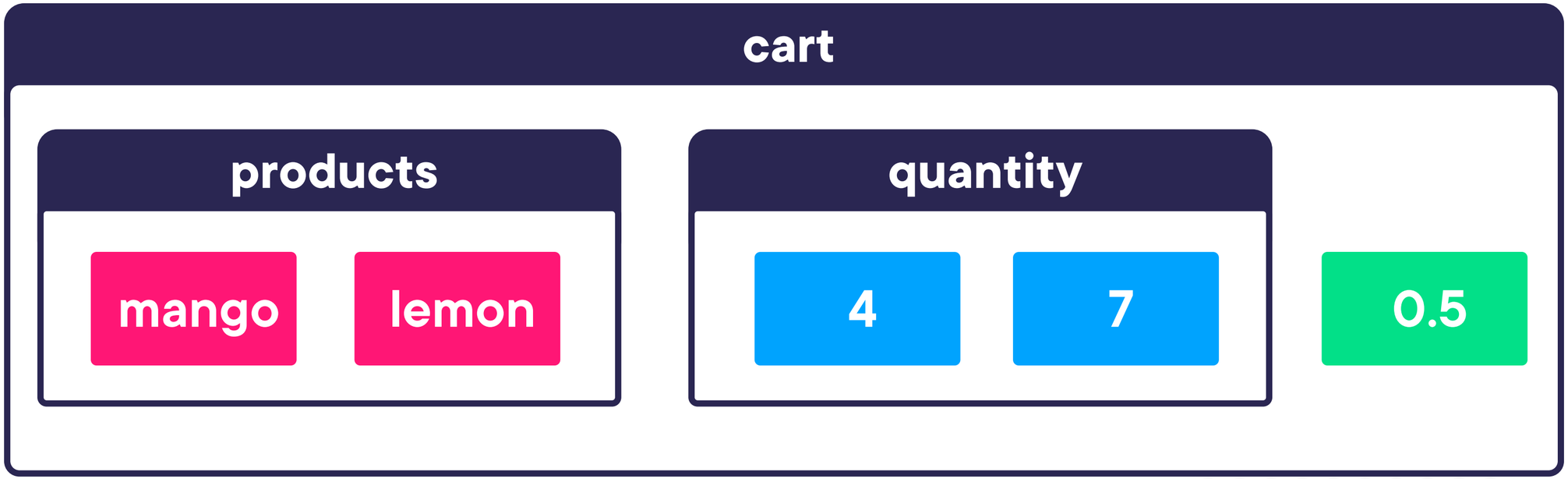


Ordered

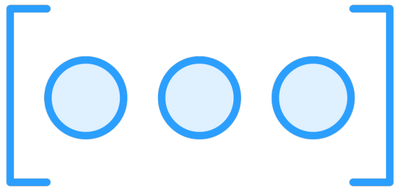
List Properties



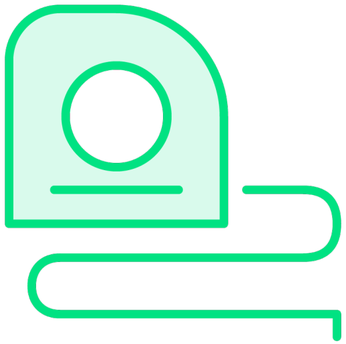
List Properties



List Properties



Ordered



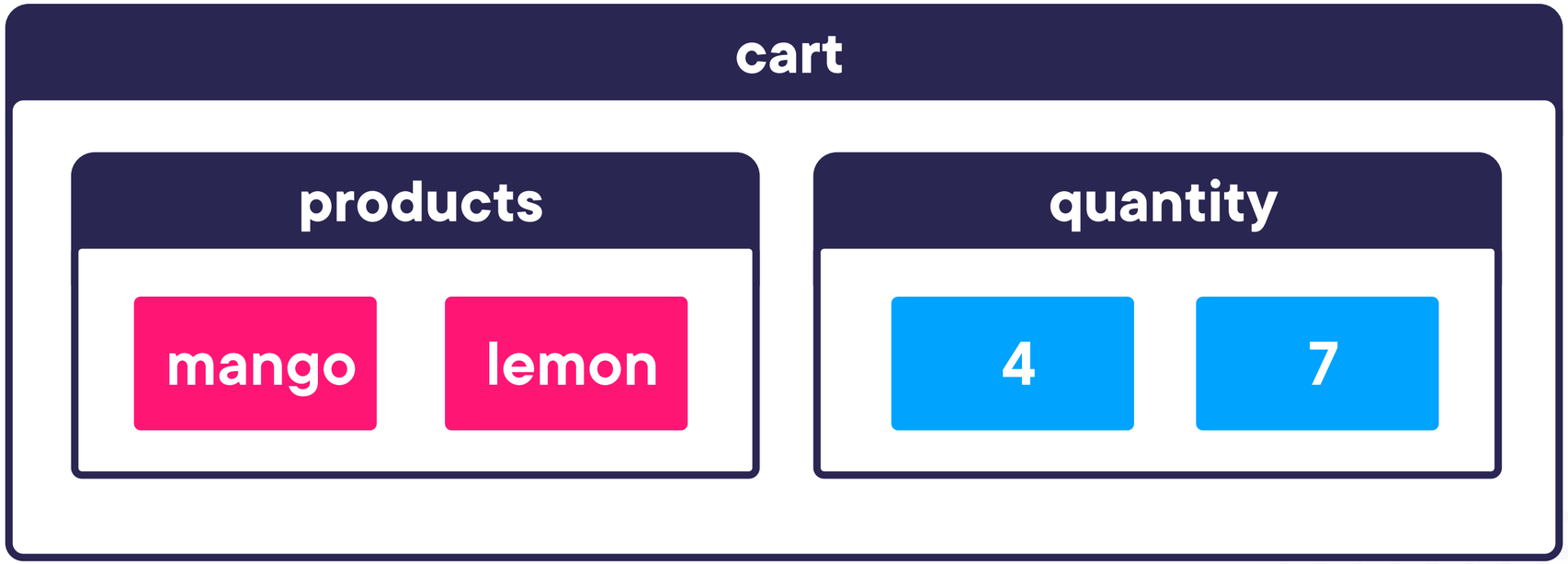
Length

List Length

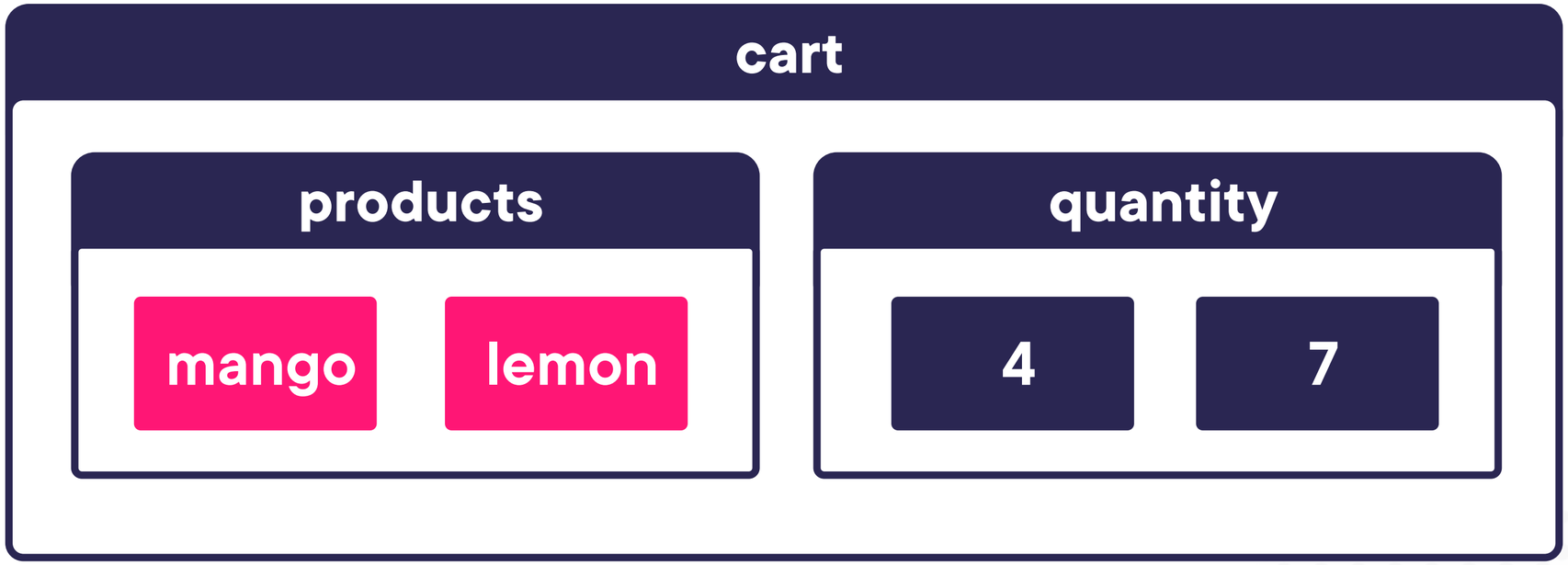
List Length

List Length

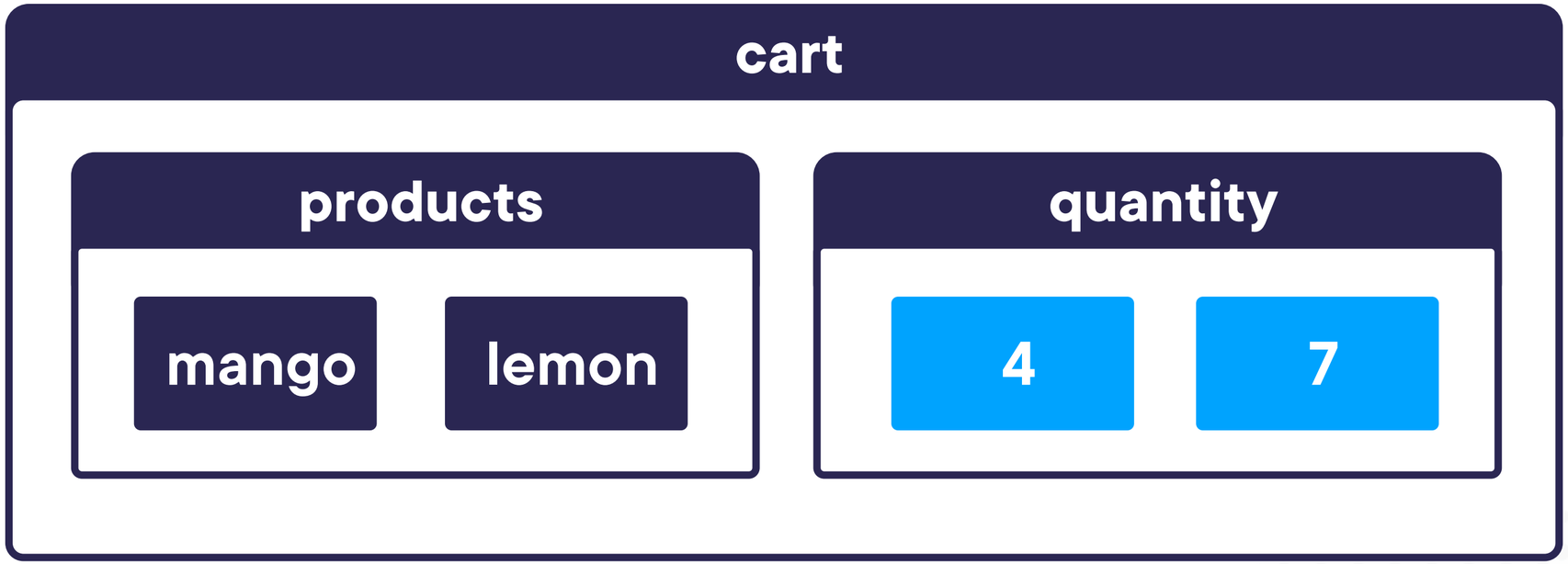
List Length



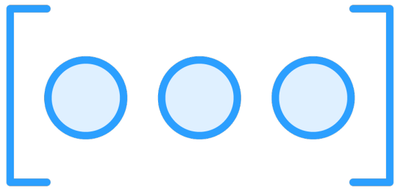
List Length



List Length

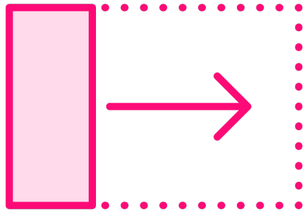


List Properties



Ordered

Length



Mutable

Demo

Create lists

- Product names
- Prices
- Stock quantities

Accessing Elements

Accessing Data

Indexing

Accessing Elements

x = [4, 7, 5, 2, 8, 1]

Value:	4	7	5	2	8	1
Index:	0	1	2	3	4	5

`x[1]` → `7`

Accessing Elements

x = [4, 7, 5, 2, 8, 1]					
Value:	4	7	5	2	8	1
Zero Index:	0	1	2	3	4	5
Negative Index:	-6	-5	-4	-3	-2	-1

x[1]	7
x[-5]	7

Accessing Data

Indexing

Slicing

x = [4, 7, 5, 2, 8, 1]

Value:	4	7	5	2	8	1
Zero Index:	0	1	2	3	4	5

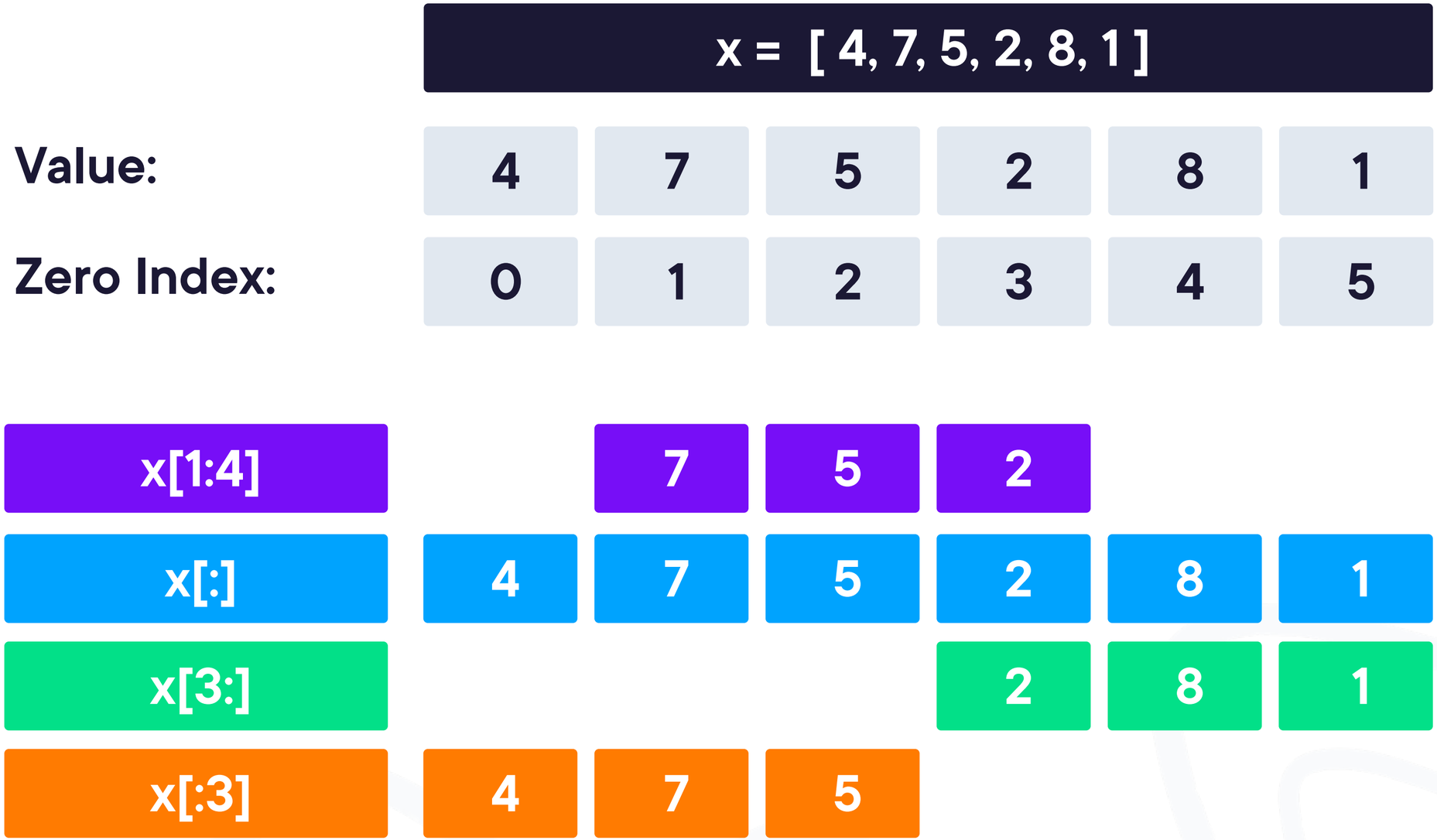
x[1:4]

7 5 2

x[1:4]

x[1:4]

x[inclusive : exclusive]


x = [4, 7, 5, 2, 8, 1]
Value:
4
7
5
2
8
1
Zero Index:
0
1
2
3
4
5
x[1:4]
7
5
2
x[:]
4
7
5
2
8
1
x[3:]
2
8
1
x[:3]
4
7
5

Demo

Indexing

Slicing

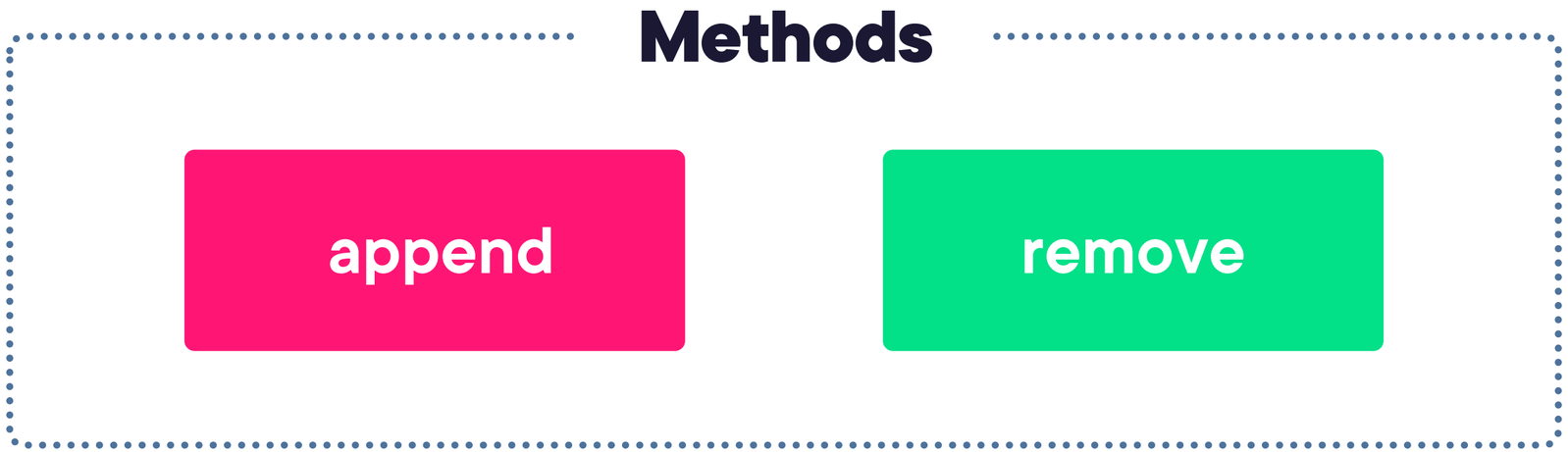
Manipulating Lists

Manipulating Lists

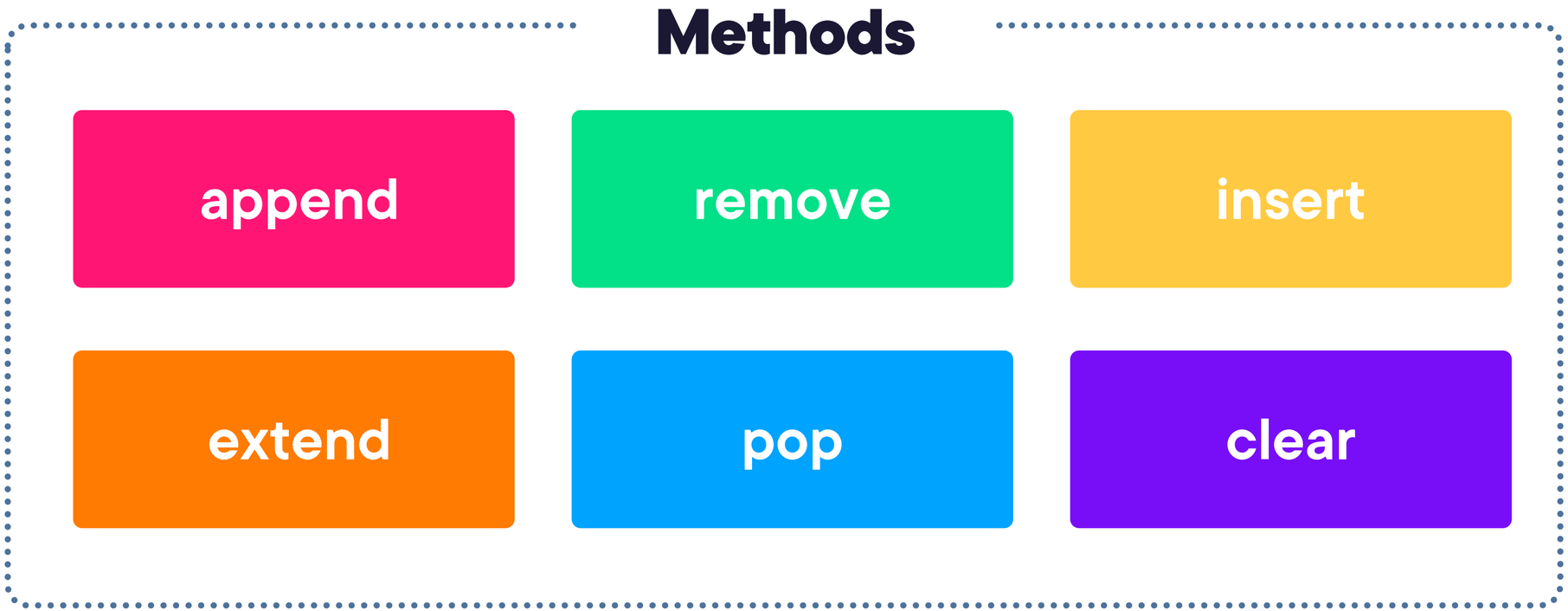
append

remove

Manipulating Lists



Manipulating Lists



Methods

A method is a function built in into a specific data structure.

Methods

```
# Creating a list
products = [‘orange’, ‘mango’]


#  Calling the len function
len(products)


#  Calling the append method
products.append(‘banana’)
```

Methods

Name

Parameters

Output

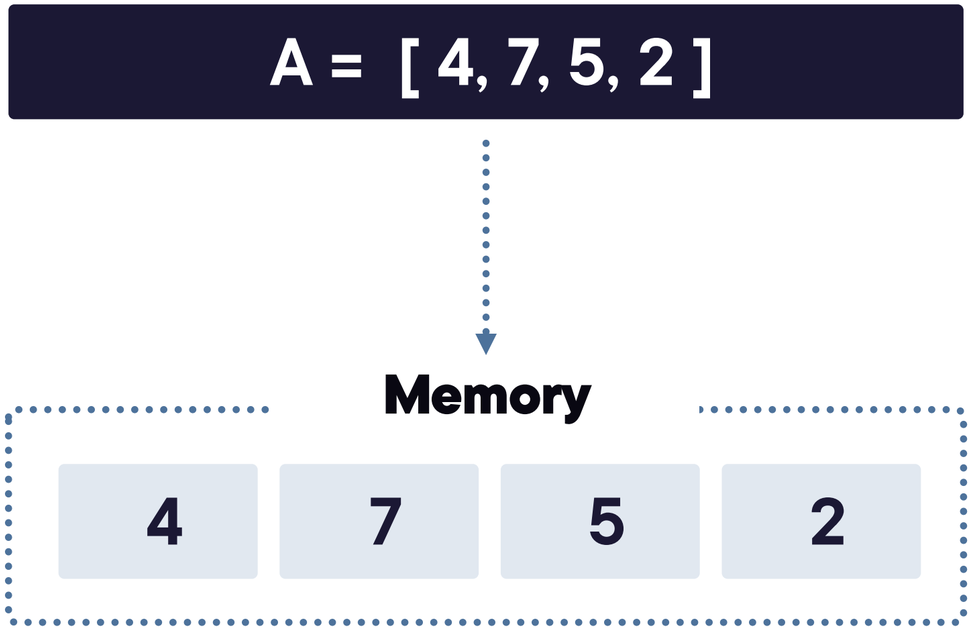
Demo

Use methods

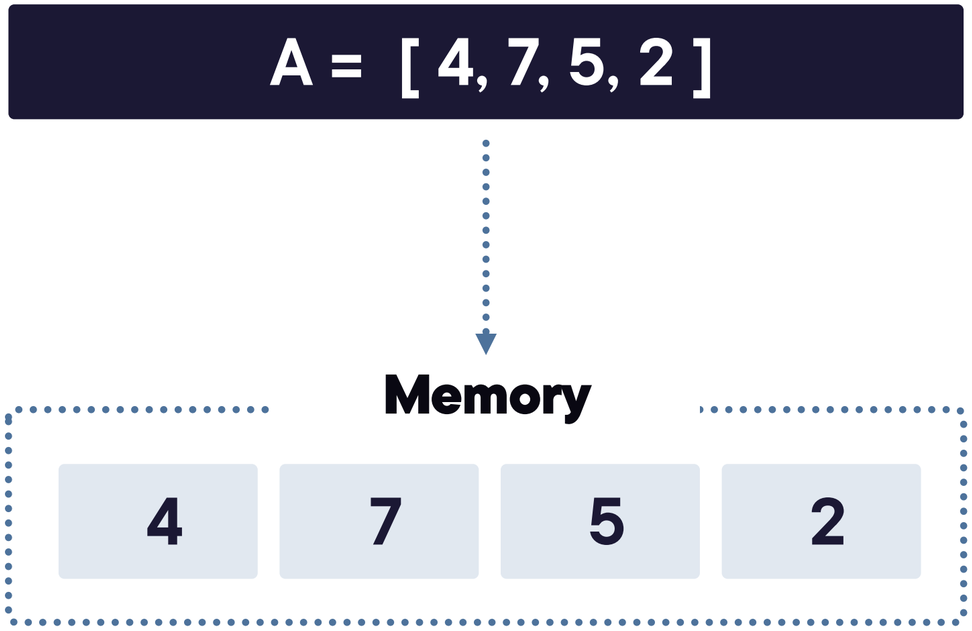
- Add elements
- Remove elements

Copying Lists

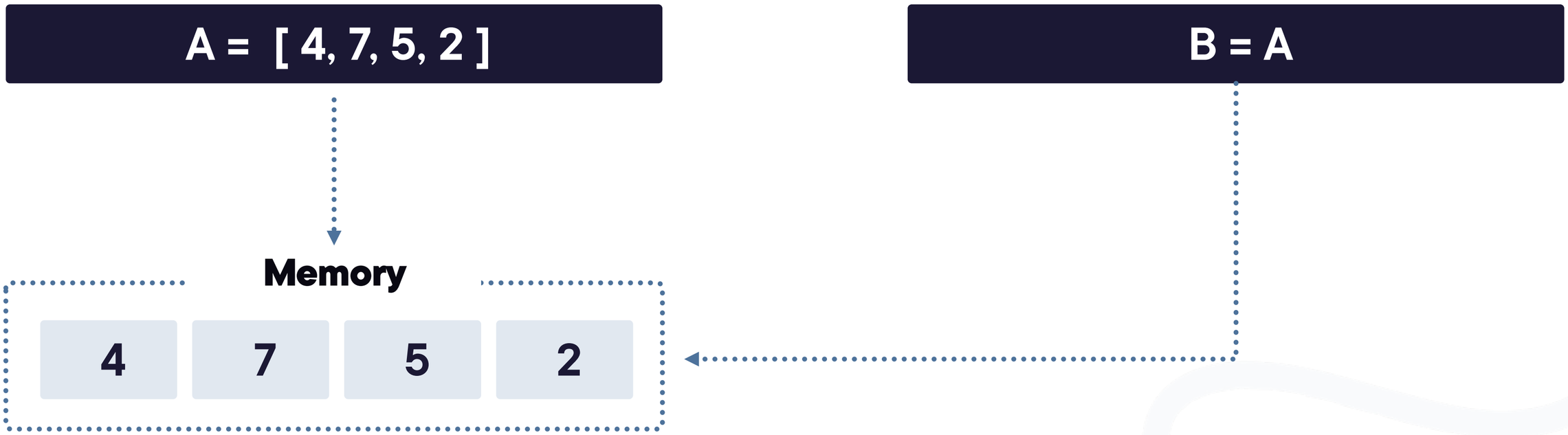
Creating Lists



Creating Lists

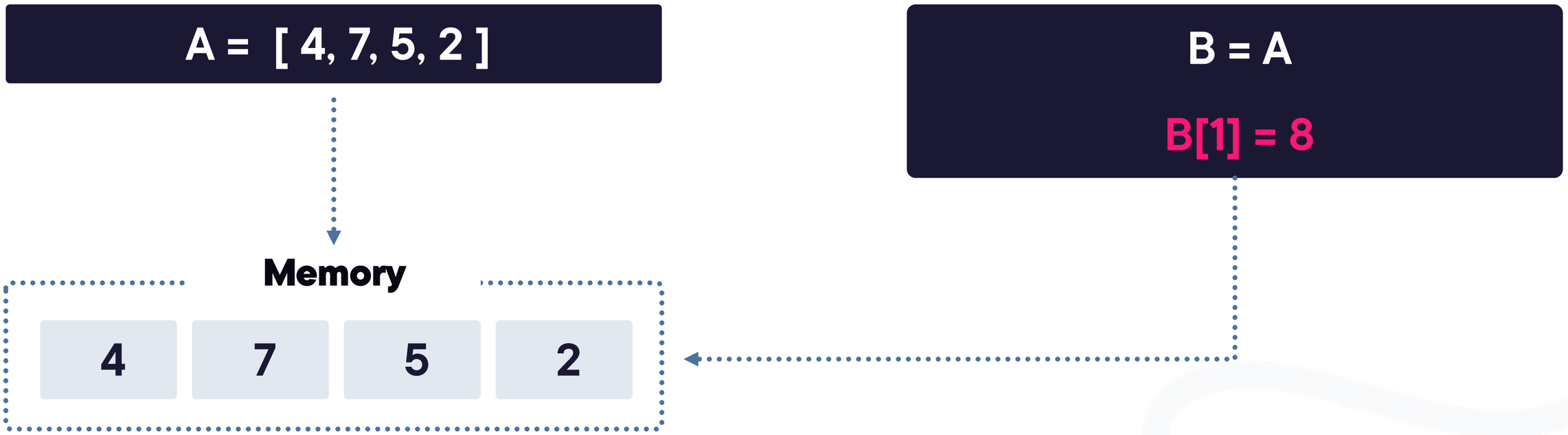


Creating Lists



Creating Lists

Creating Lists



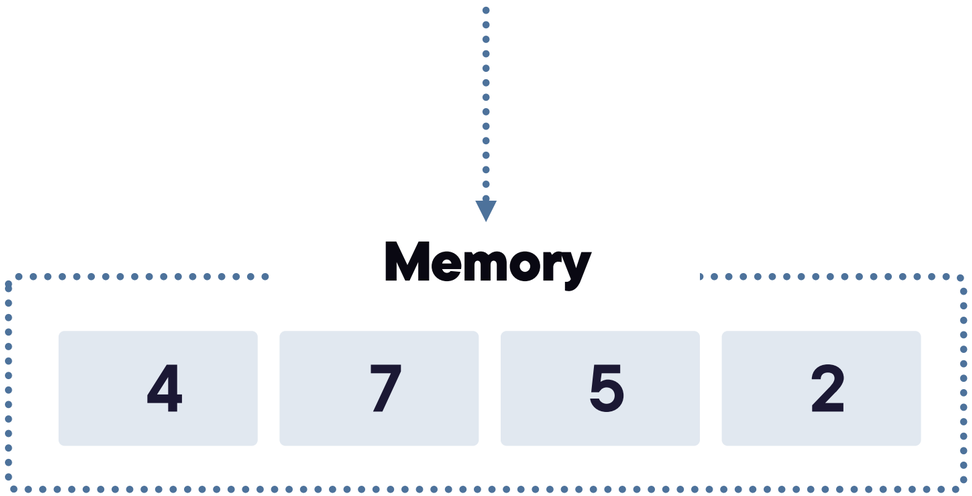
Creating Lists

Creating Lists

Creating Lists

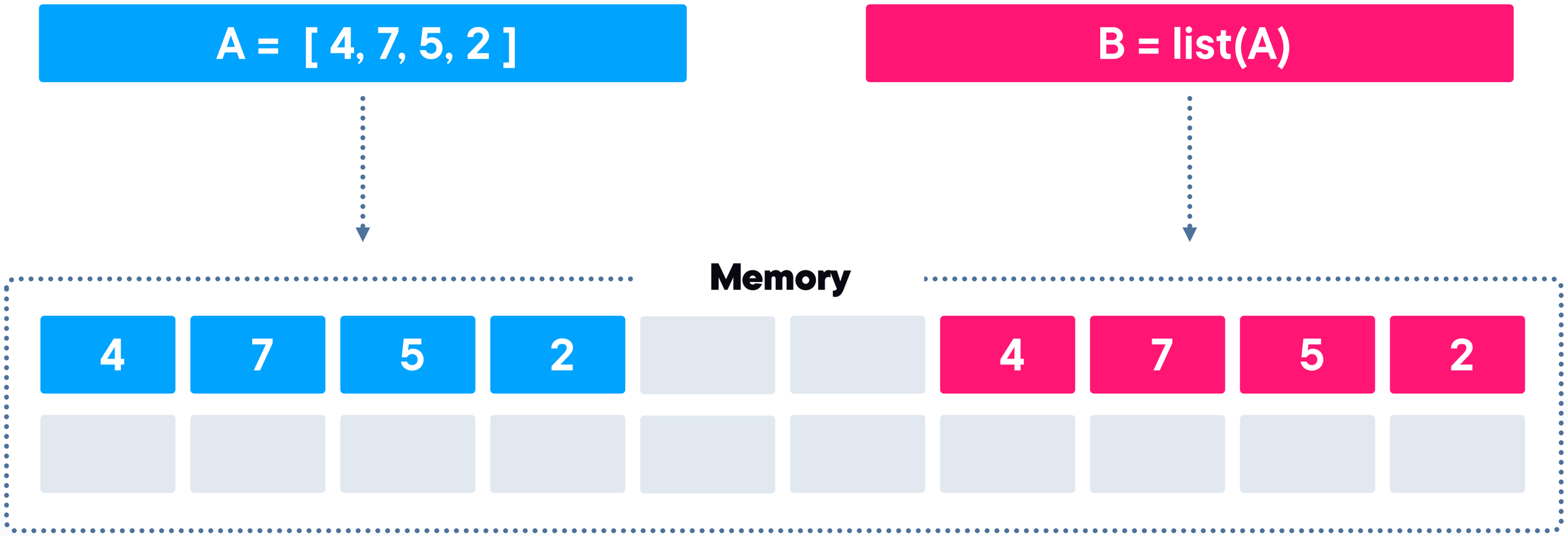
Creating Lists

```
A = [ 4, 7, 5, 2 ]
```




```
B = list(A)
```

Creating Lists



Cloning Lists

List function

Colon operator

Demo

Aliasing

Cloning

Demo

Explore methods

- Sort
- Count

Summary

Lists are data structures

Lists are mutable

Retrieve elements

- Zero Index
- Negative Index

Methods

Up Next:

Managing Tuples and Sets