Indices and Ranges

Matt Honeycutt
HEROIC CONSULTING

@matthoneycutt   www.trycatchfail.com

Overview

Introduction to Range and Index types

Making sense of ranges, indices, and offsets

Ranges and indices in action

C# 7 - Span

```
var numbers = new[] { 1, 2, 3, 4, 5 };

var slice = numbers.AsSpan().Slice(2, 2);

//slice now has 3, 4
```

A type and memory-safe representation of a contiguous sequence of another structure.

C# 8 - Range

```
var numbers = new[] { 1, 2, 3, 4, 5 };

var slice = numbers[2..4];

//slice now has 3, 4
```

Easily extract a range from a sequence

C# 8 - Range

```
var numbers = new[] { 1, 2, 3, 4, 5 };

Range range = 2..4;
var slice = numbers[range];

//slice now has 3, 4
```

A Range is a first-class type

C# 8 - Range

```
var numbers = new[] { 1, 2, 3, 4, 5 };

Range range = 2..4;
var slice = numbers[range];

//slice now has 3, 4
```

Range operator:

..

C# 8 - Index

```csharp
var numbers = new[] { 1, 2, 3, 4, 5 };

var number = numbers[1];

//number is now 2
```

Index references a location in a sequence

C# 8 - Index

```
var numbers = new[] { 1, 2, 3, 4, 5 };

var number = numbers[^1];

//number is now 5
```

Use the ^ (hat) operator to declare an Index relative to the end of a sequence

C# 8 - Index

```
var numbers = new[] { 1, 2, 3, 4, 5 };

Index index = ^2;

var number = numbers[index];

//number is now 4
```

Index is a first-class type, too!

C# 8 Ranges and Indices

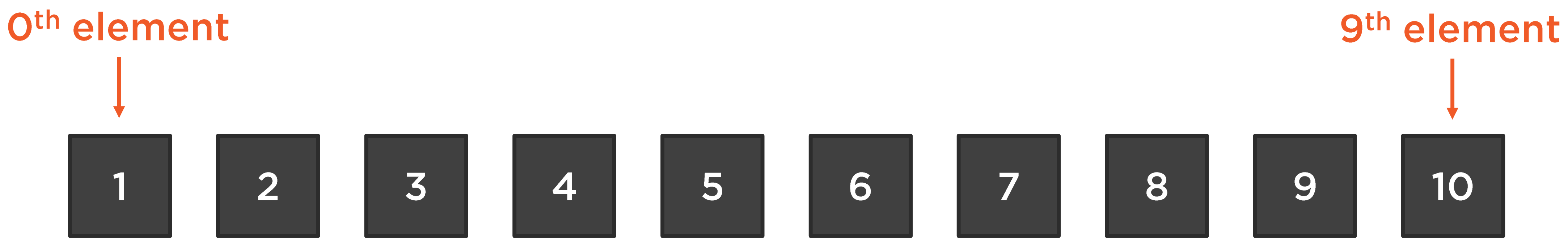
```
var numbers = new[] { 1, 2, 3, 4, 5 };

var lastTwo = numbers[^2..];
```

Get the last two items

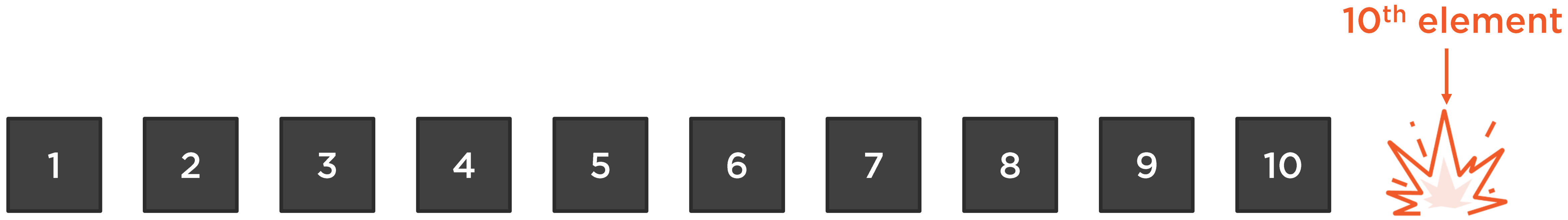
```
var position = 2;
Index startIndex = ^position;
Range range = startIndex..;

var lastTwo = numbers[startIndex];
```

Build up range from variables

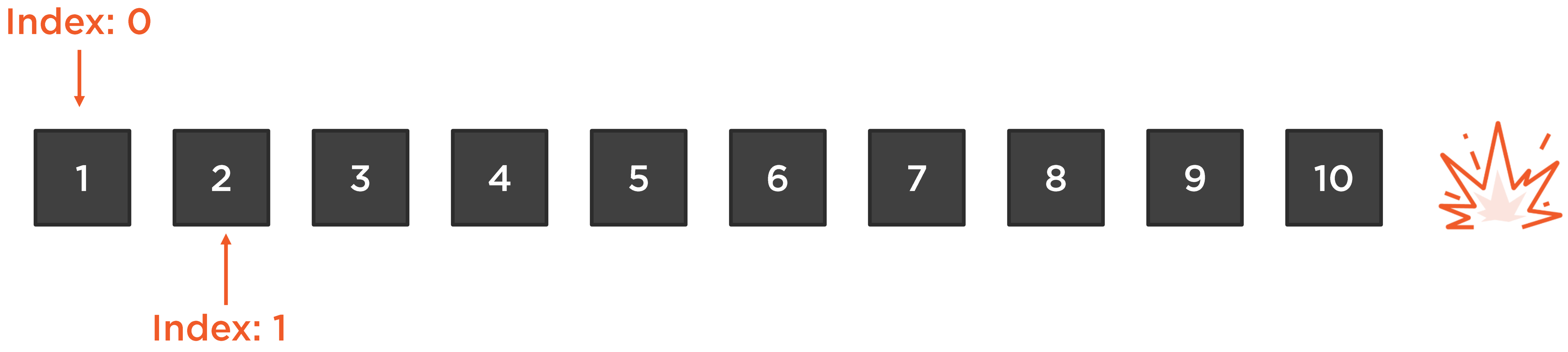
Offsets



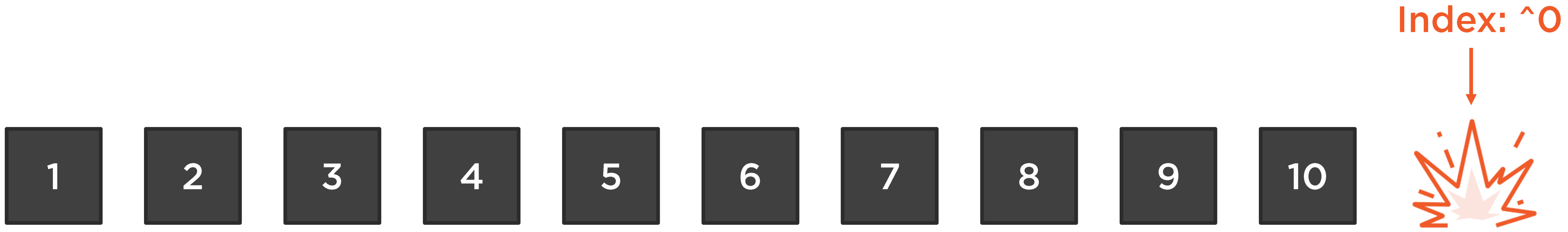
Offsets



Offsets

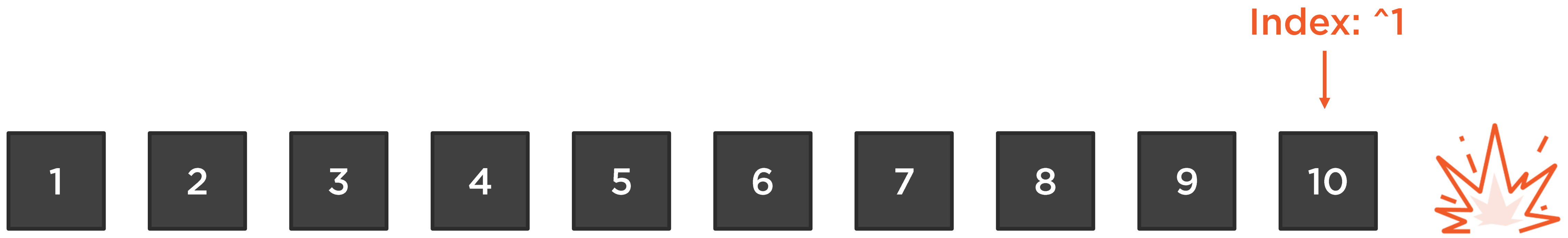


Offsets



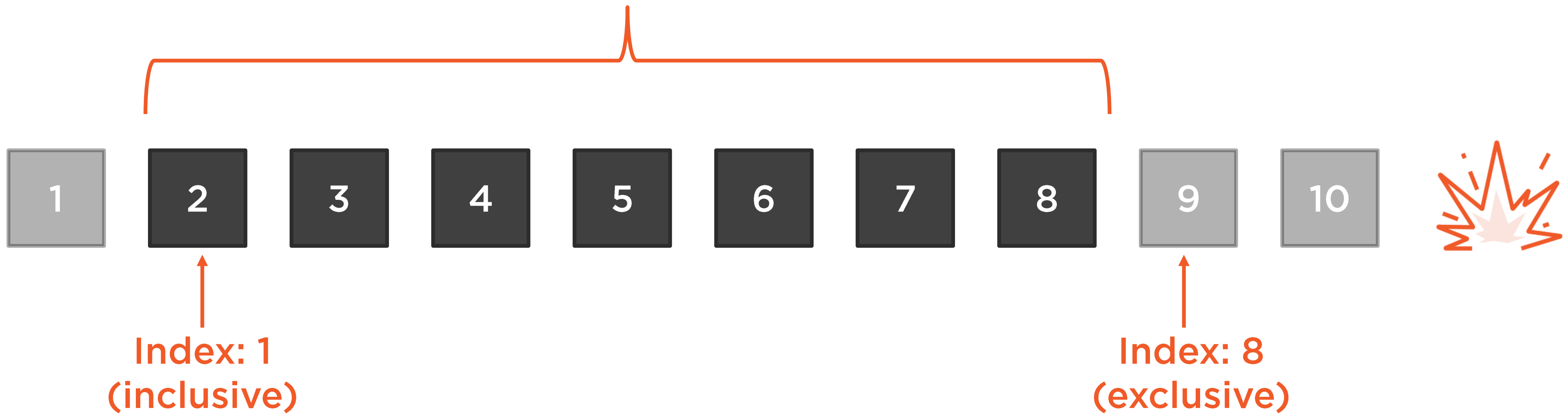
```
numbers[^0] == numbers[numbers.Length]
```

Offsets



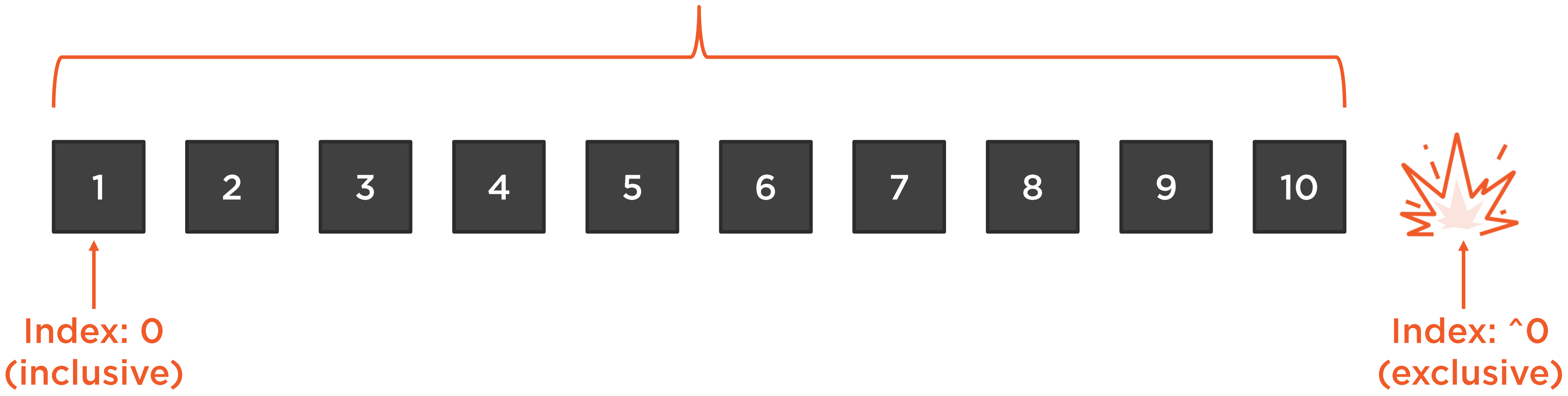
```
numbers[^0] == numbers[numbers.Length]
```

Specifying Ranges



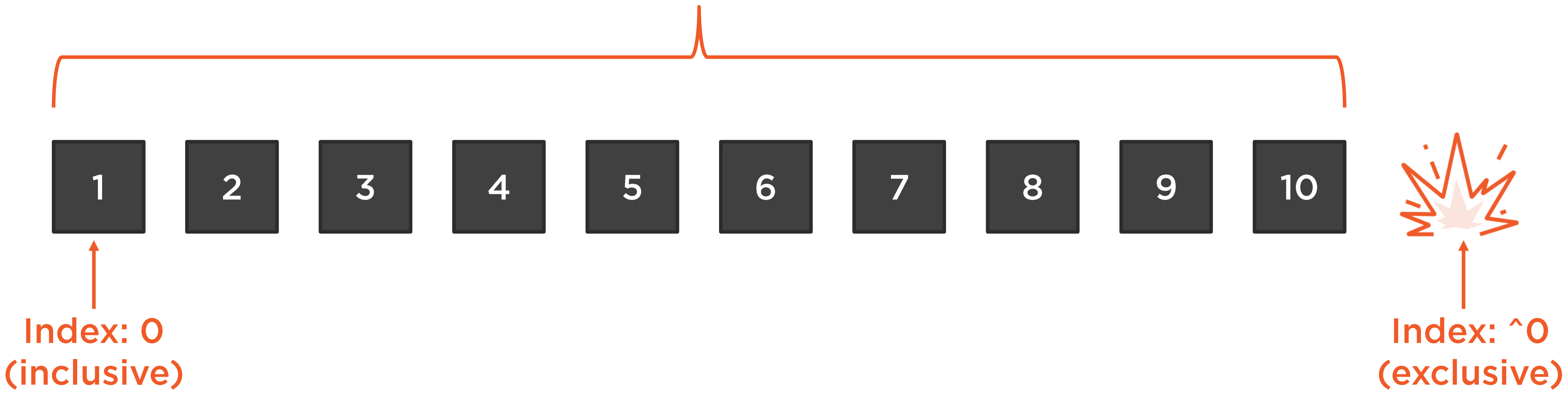
```
var range = numbers[1..8]
```

Specifying Ranges



```
var range = numbers[0..^0]
```

Specifying Ranges



```
var range = numbers[0..]
```

Code time!

Demo

Using indices and ranges

Summary

Range

- Range operator: ..

Index

- Hat operator: ^

Important things to remember

- Start of range is inclusive
- End of range is exclusive
- Indexing from the start is 0-based
- Indexing from the end is relative to the length

Up Next

Built-in JSON support