Arrays

Matt Henry
VP of Product & Independent Developer

fancymatt.com

Ways Arrays are Like Objects

Arrays contain properties

Access properties with bracket notation

Array functions come from their prototype

Can create arrays in many of the same ways as objects

Ways Arrays are Unlike Objects

Properties are zero-based numbers

Cannot use dot notation

New Array.prototype functions to learn

Overview: Array Modules

Module 5: Arrays

- Arrays are Objects
- Creating Arrays
- Identifying Arrays
- Accessing Array Elements
- Cutting Up Arrays
- Putting Together Arrays
- Cloning Arrays
- Sorting Arrays
- Sparse Arrays
- Other Array Types

Module 6: Iteration Deep Dive

- Array Iteration
- For Loops
- Summarizing Arrays
- Searching Through Arrays
- Map

Overview: Array Modules

Module 5: Arrays

- Arrays are Objects
- Creating Arrays
- Identifying Arrays
- Accessing Array Elements
- Cutting Up Arrays
- Putting Together Arrays
- **Cloning Arrays**
- **Sorting Arrays**
- **Sparse Arrays**
- **Other Array Types**

Module 6: Iteration Deep Dive

- Array Iteration
- For Loops
- Summarizing Arrays
- Searching Through Arrays
- Map

Overview: Array Modules

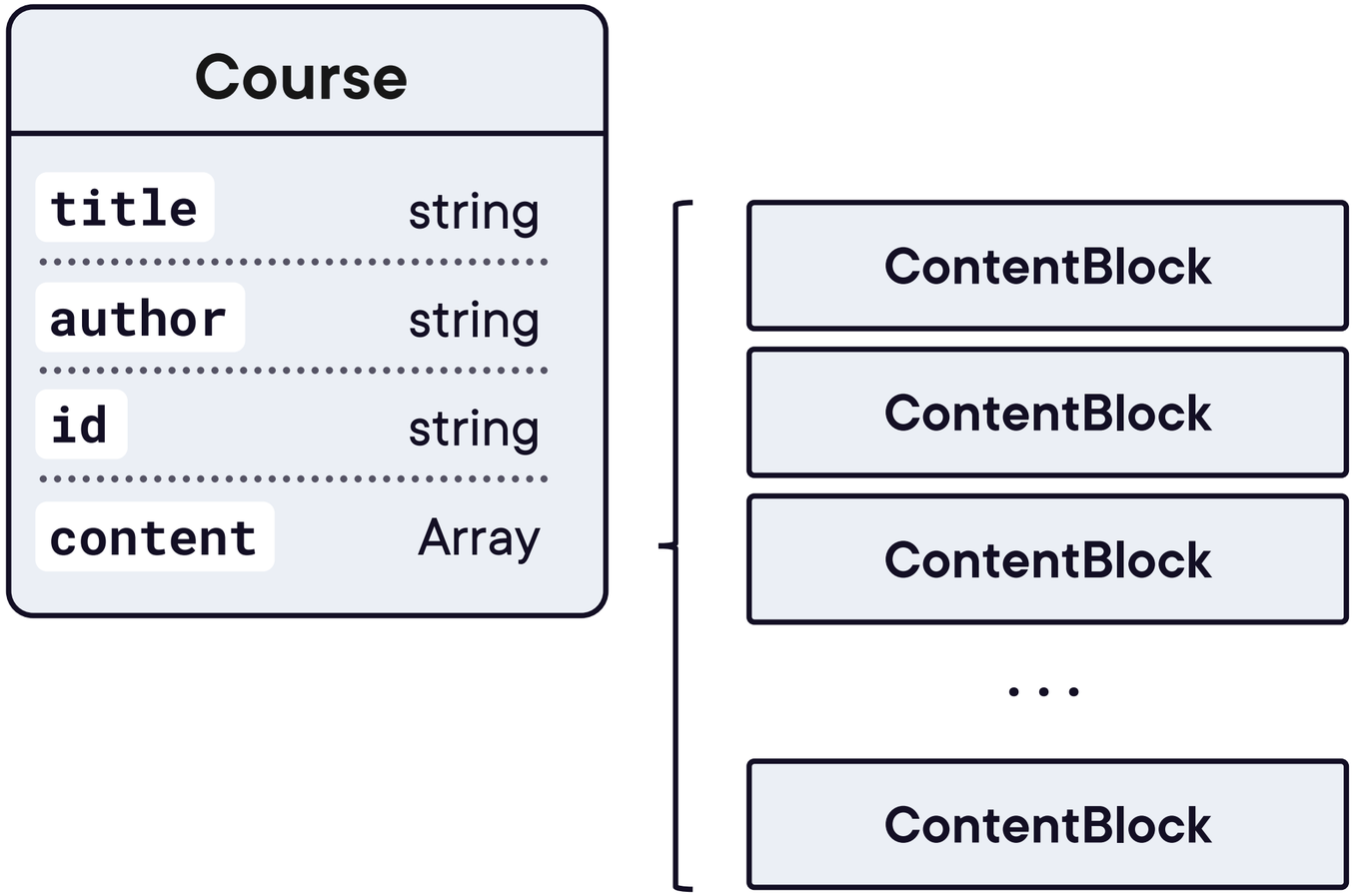
Module 5: Arrays

- Arrays are Objects
- Putting Together Arrays
- Creating Arrays
- Cloning Arrays
- Identifying Arrays
- Sorting Arrays
- Accessing Array Elements
- Sparse Arrays
- Cutting Up Arrays
- Other Array Types

Module 6: Iteration Deep Dive

- Array Iteration
- For Loops
- Summarizing Arrays
- Searching Through Arrays
- Map

Course Object Structure



Five Ways to Create Arrays

New Syntax

```
const myArray = new Array();
```

Array Literal

```
const myArray = [];
```

Object.create

```
const myArray = Object.create(Array.prototype);
```

Array.of

```
const myArray = Array.of();
```

Array.from

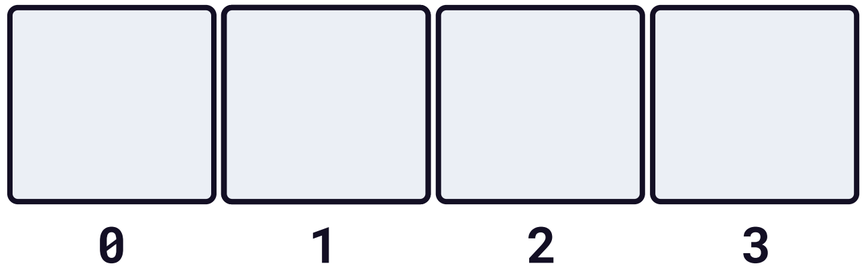
```
const myArray = Array.from(arrayLikeObject);
```

Array-like Objects

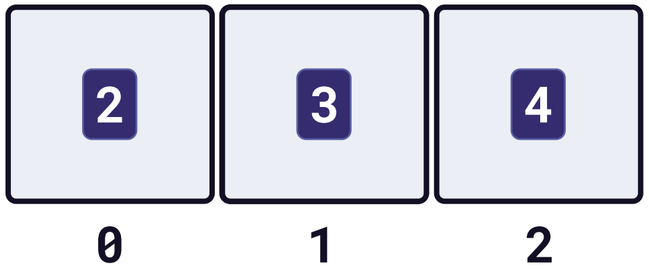
An object which is iterable, but does not link to Array.prototype.

Passing One Number Into Array Constructor

```
const myArray = new Array(4);
```




```
const myArray = new Array(2,3,4);
```



Up Next:

Identifying Arrays

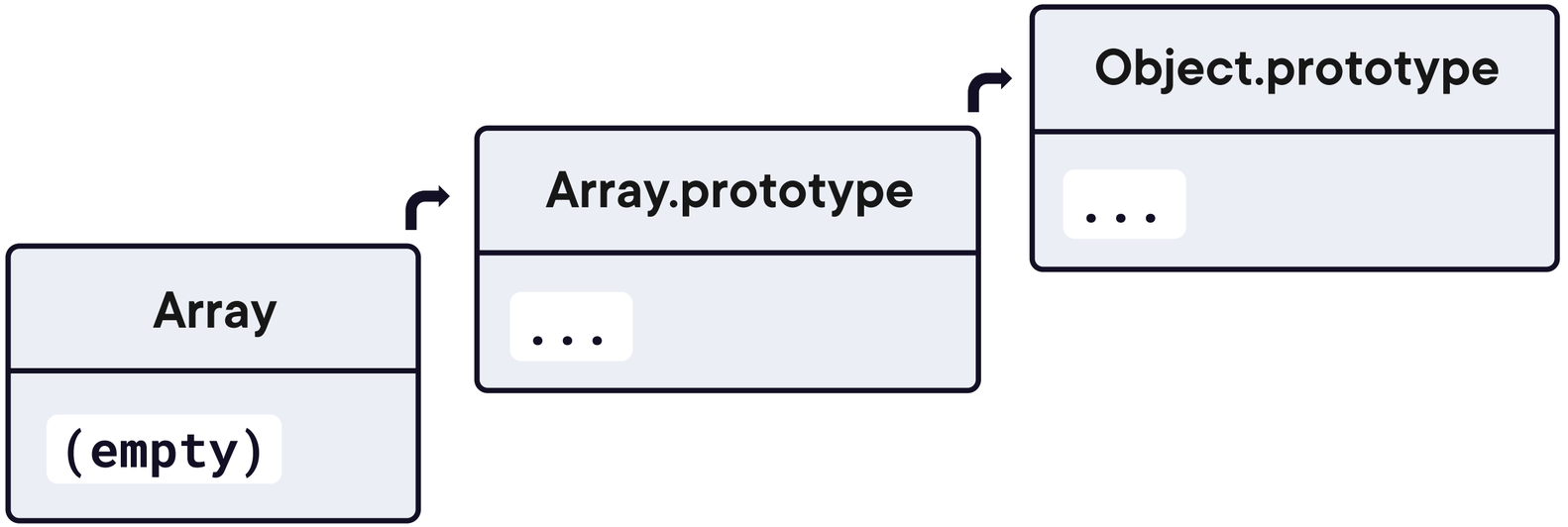
How to Check if Something is an Array

instanceof

Object.isPrototypeOf

Array.isArray

All Arrays Link to Object.Prototype



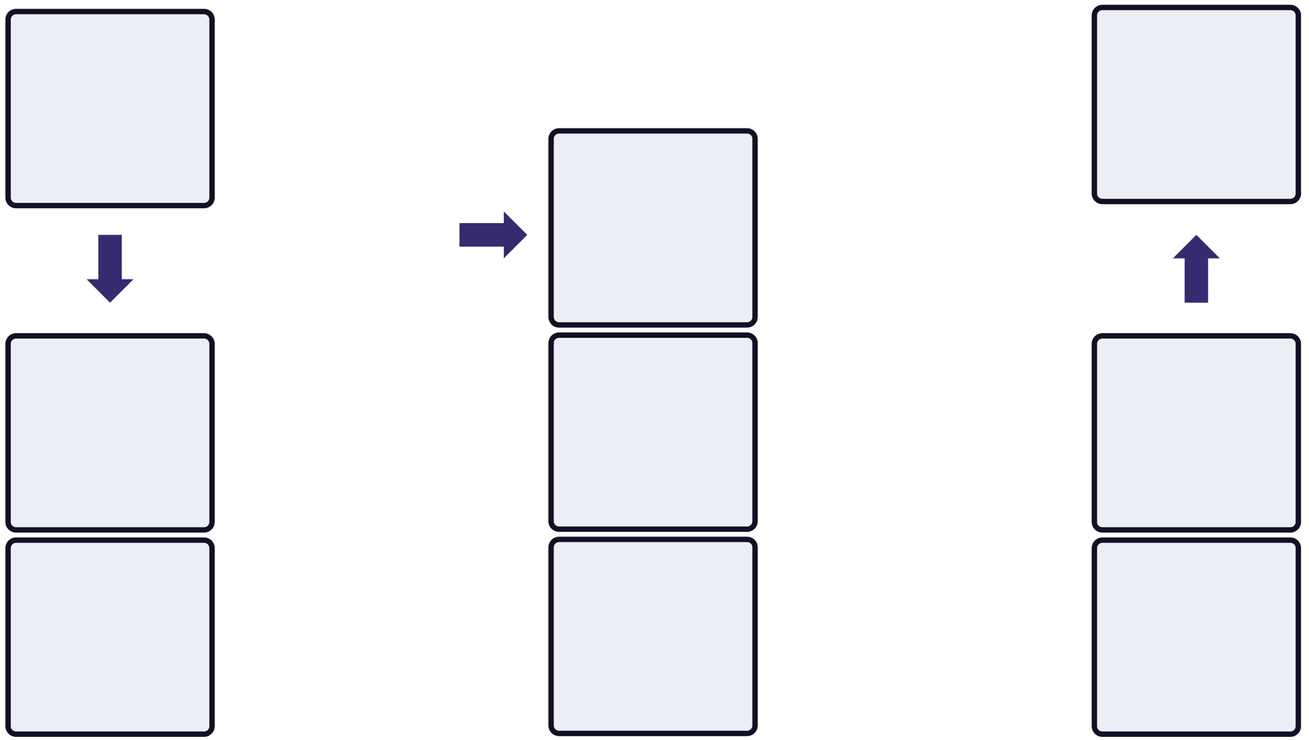
Comparing Ways to Check if Something is an Array

	Array	Array in Prototype Chain
instanceof		
Object.isPrototypeOf		
Array.isArray		

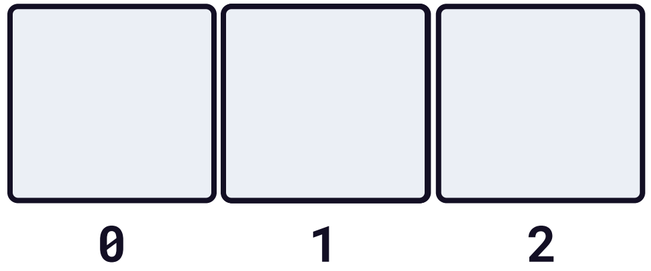
Up Next:

Adding and Accessing Array Elements

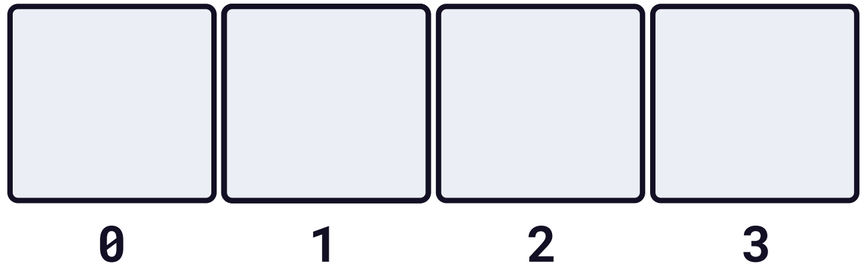
In This Video



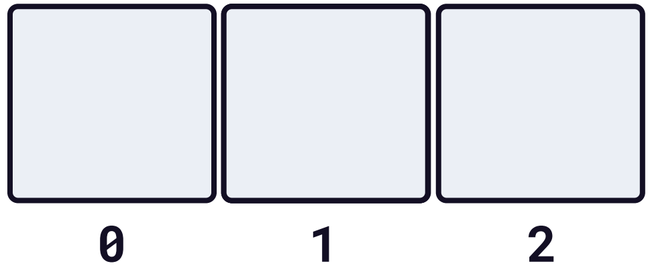
In This Video



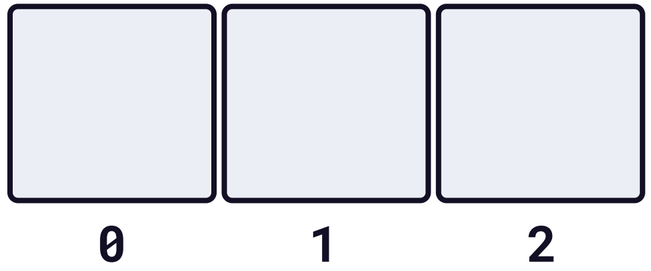
In This Video

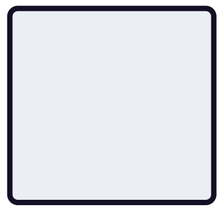


In This Video



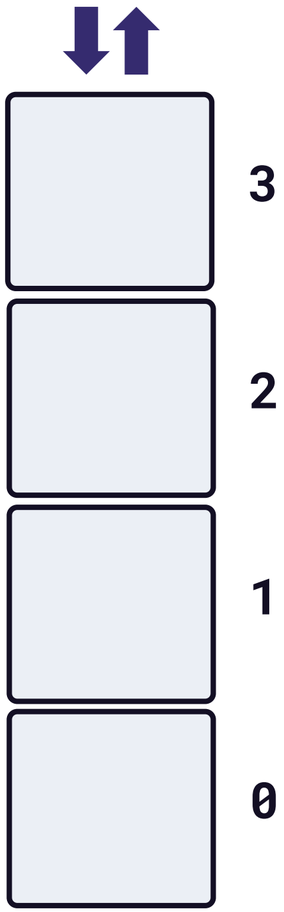
In This Video



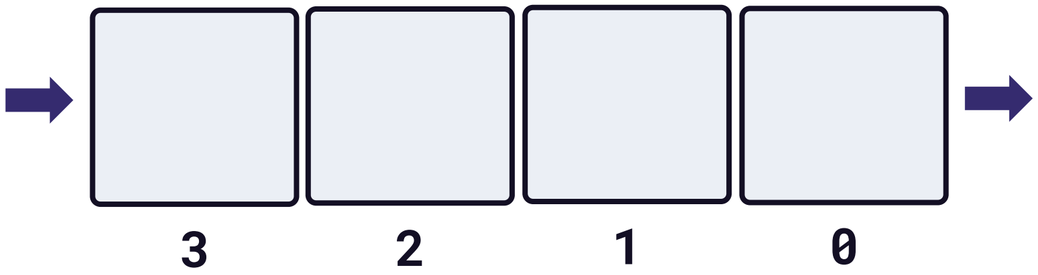


Stacks and Queues

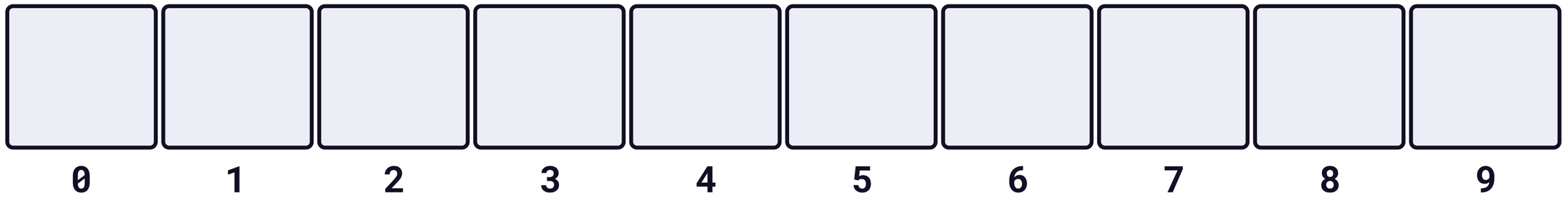
Stack



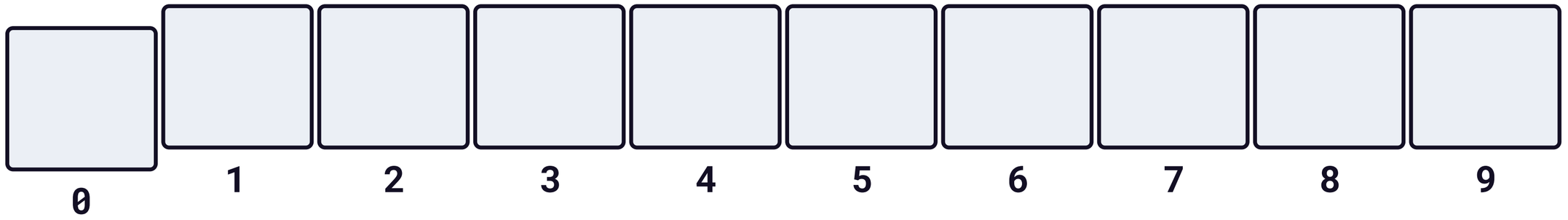
Queue



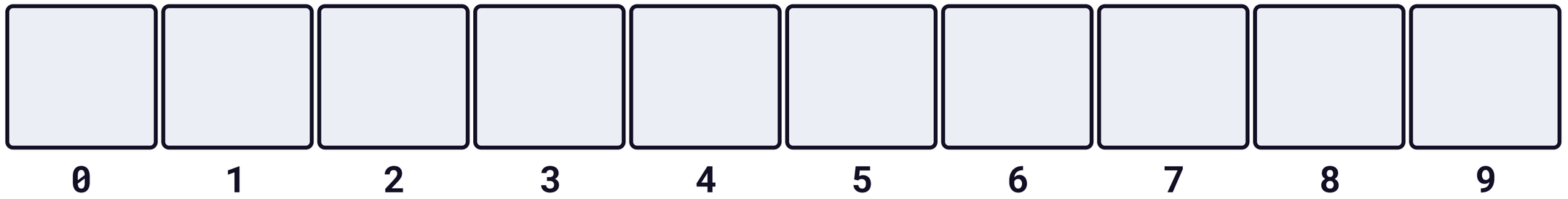
Arrays Use Indexes



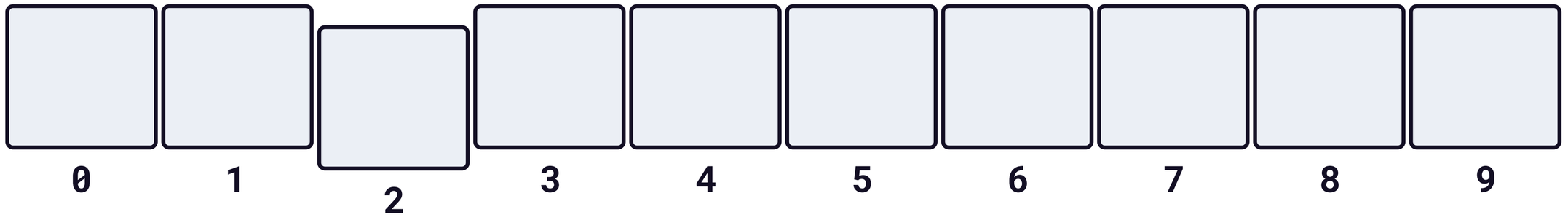
Arrays Use Indexes



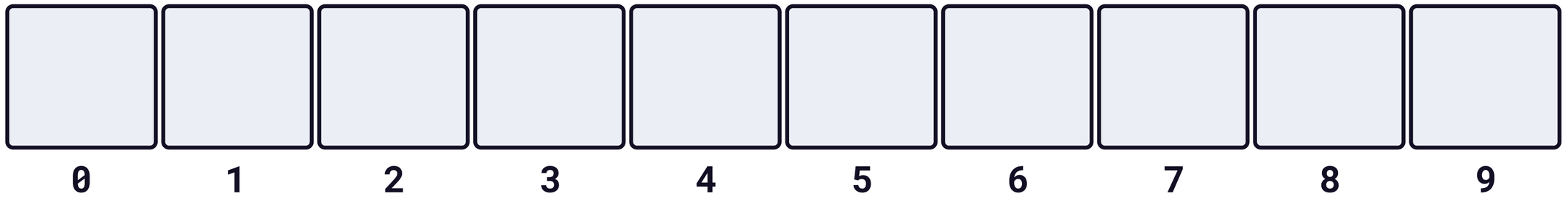
Arrays Use Indexes



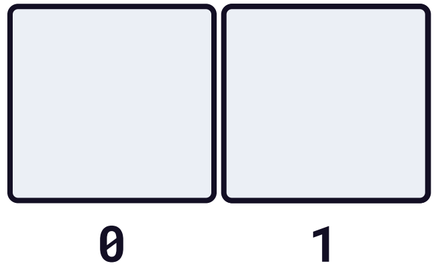
Arrays Use Indexes



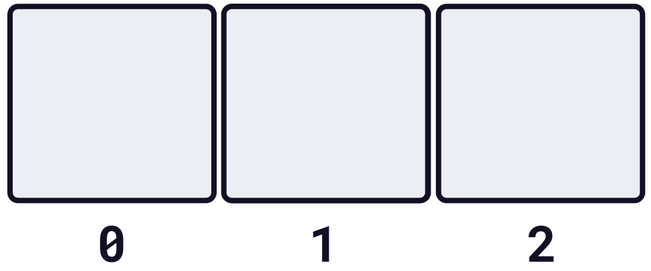
Arrays Use Indexes



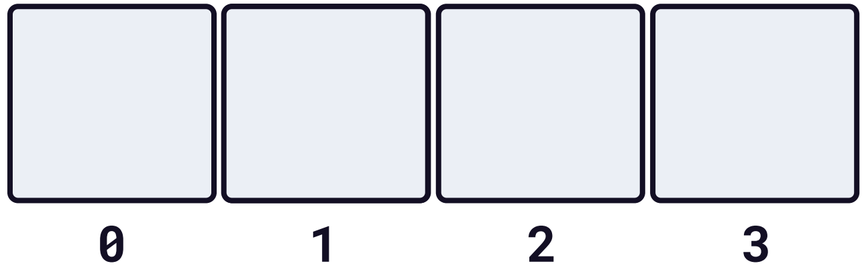
Working with Arrays as a Complete Unit



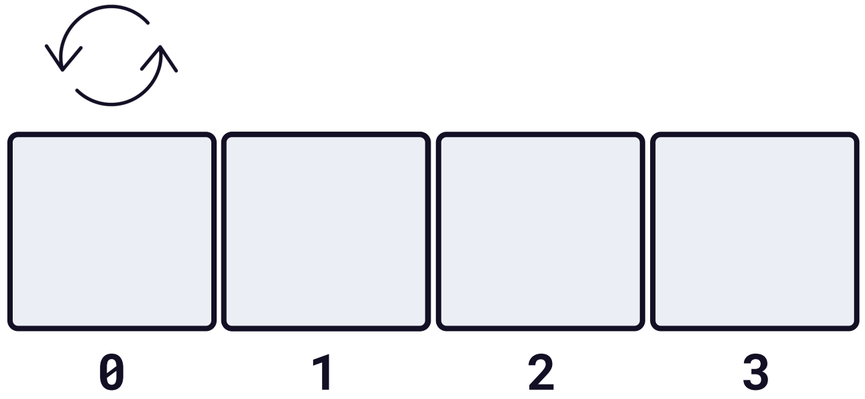
Working with Arrays as a Complete Unit



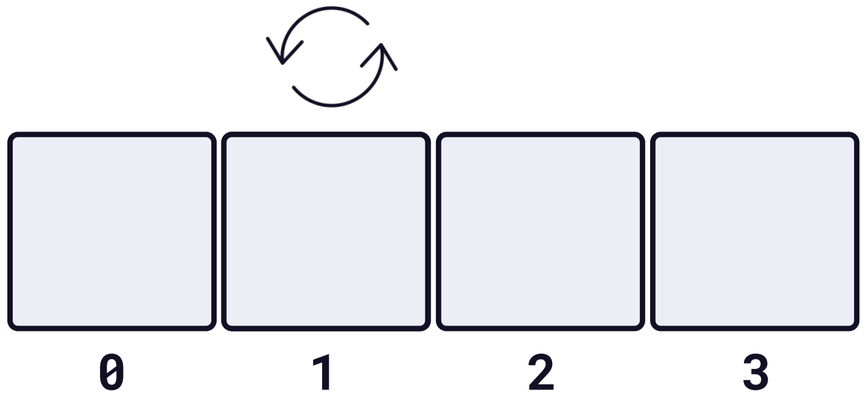
Working with Arrays as a Complete Unit



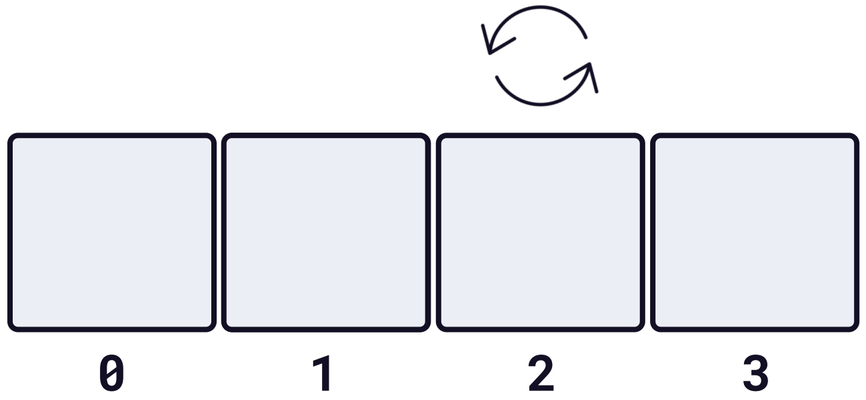
Working with Arrays as a Complete Unit



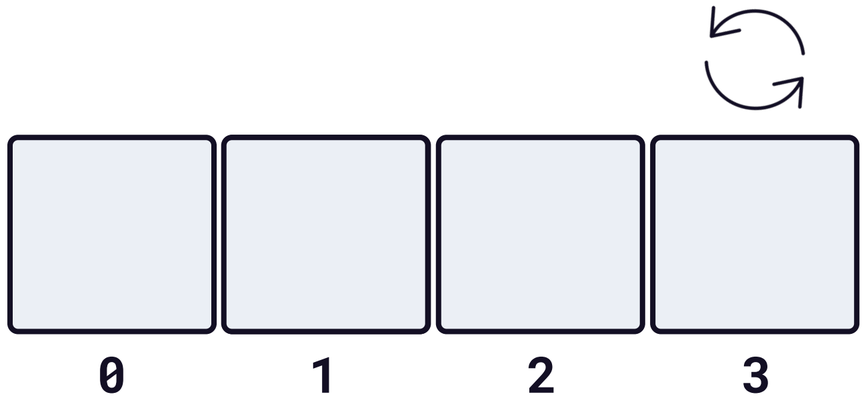
Working with Arrays as a Complete Unit



Working with Arrays as a Complete Unit

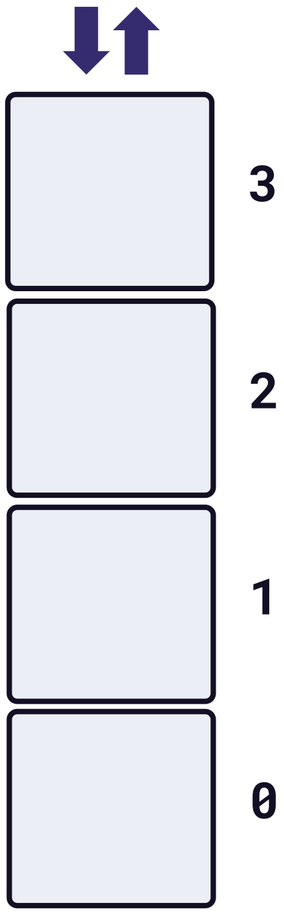


Working with Arrays as a Complete Unit

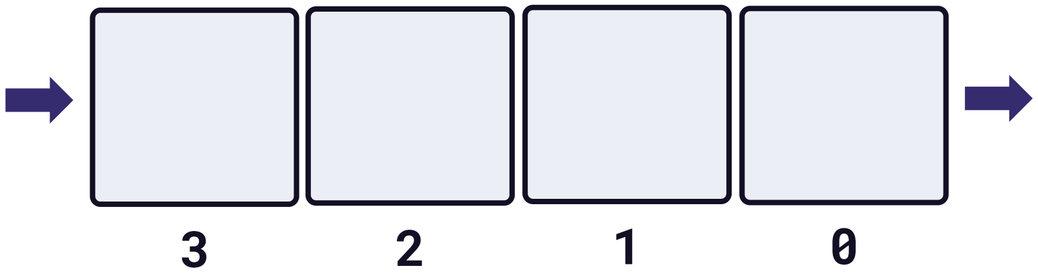


Stacks and Queues

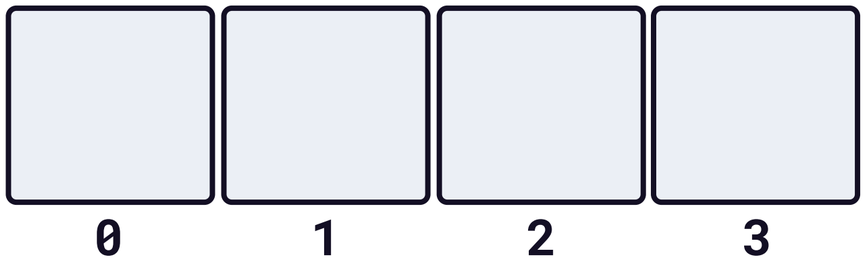
Stack



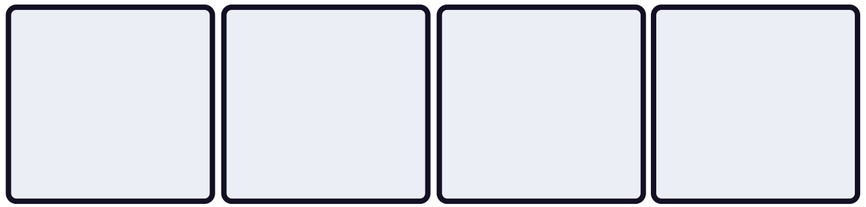
Queue



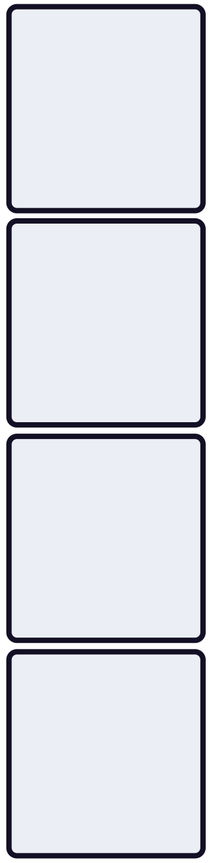
Arrays as Stacks



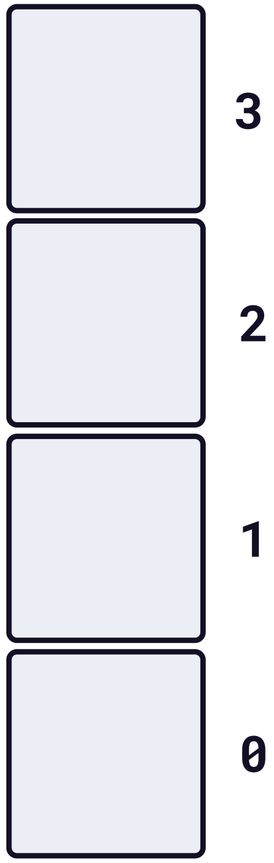
Arrays as Stacks



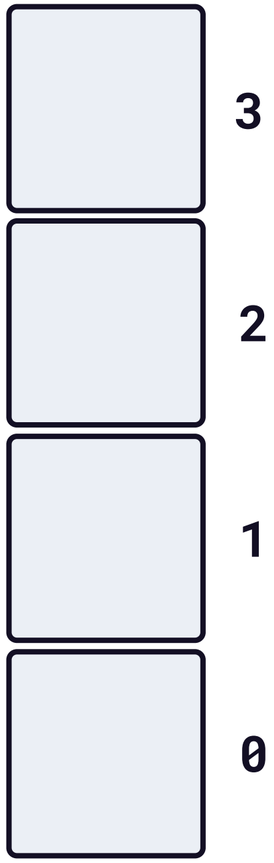
Arrays as Stacks



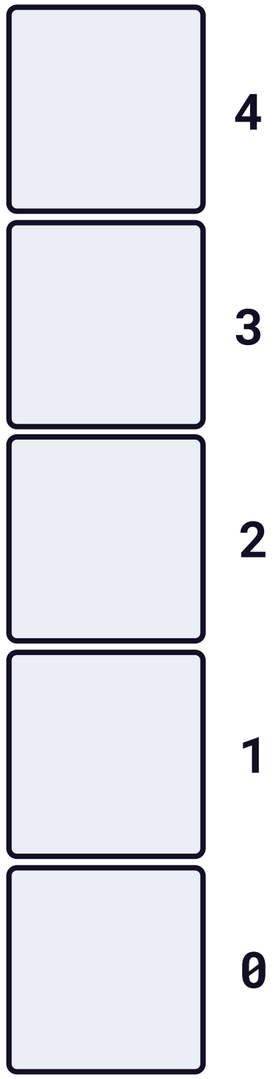
Arrays as Stacks



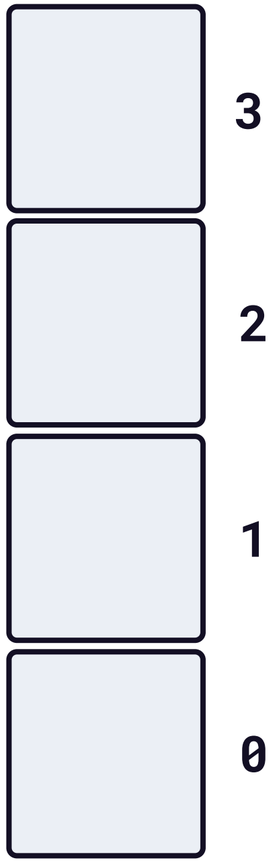
Arrays as Stacks



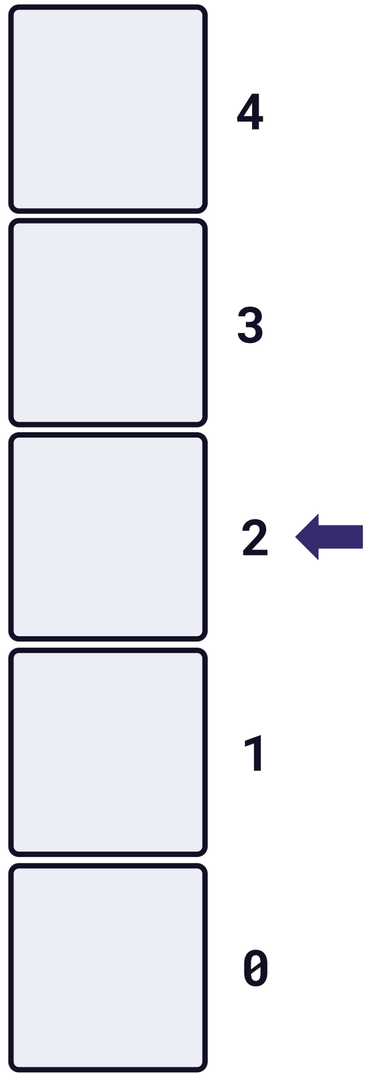
Arrays as Stacks



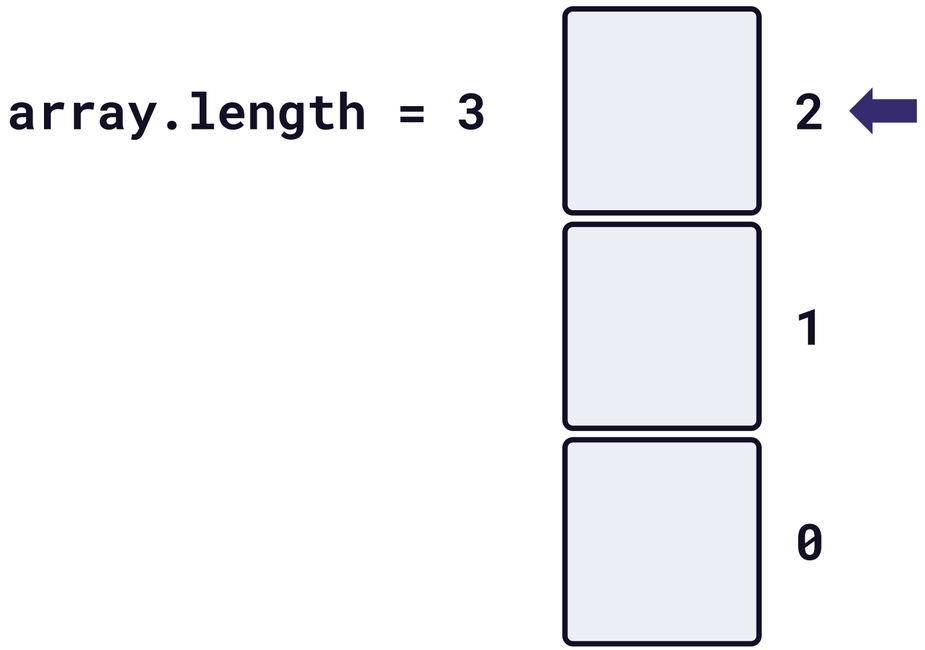
Arrays as Stacks



Arrays as Stacks



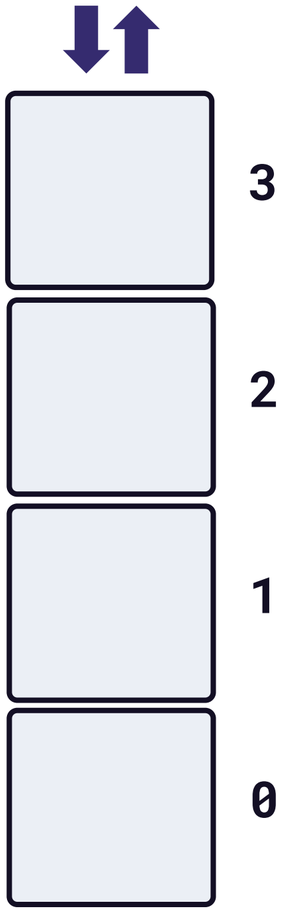
Array Length vs. Last Index



Stacks and Queues

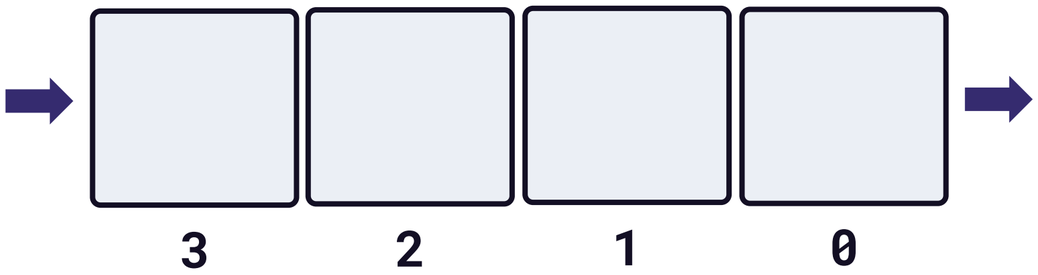
Stack

“Last in, first out” (LIFO)

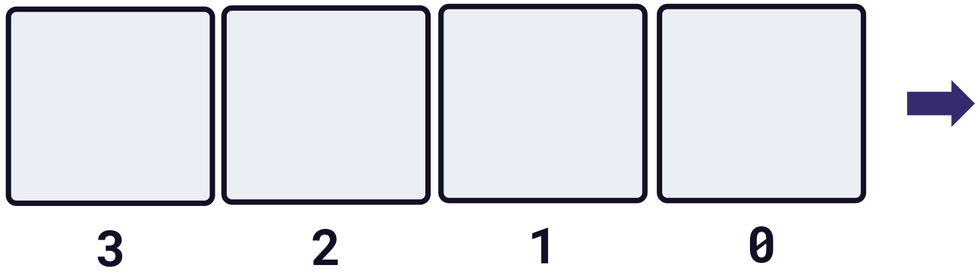


Queue

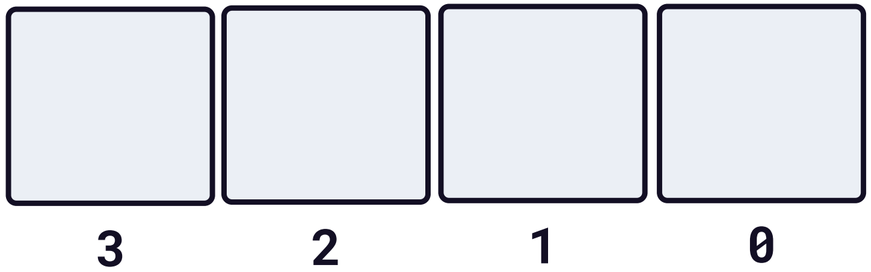
“First in, first out” (FIFO)



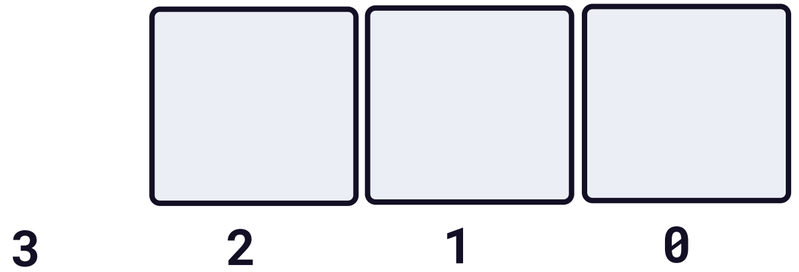
Arrays as Queues



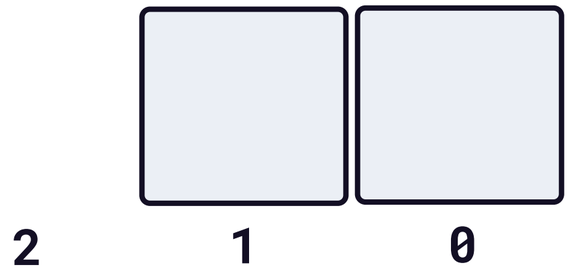
Arrays as Queues



Arrays as Queues

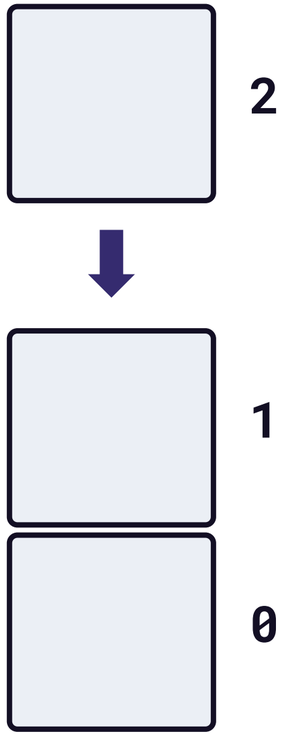


Arrays as Queues

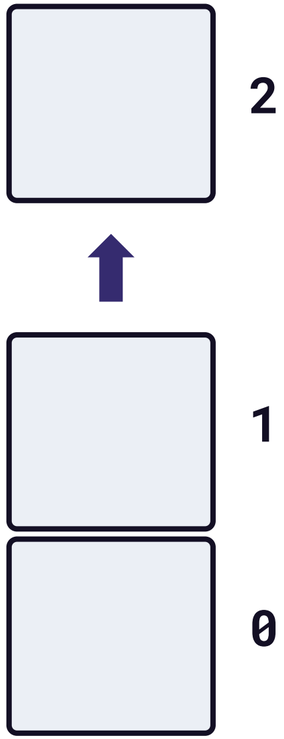


Array Functions Compared

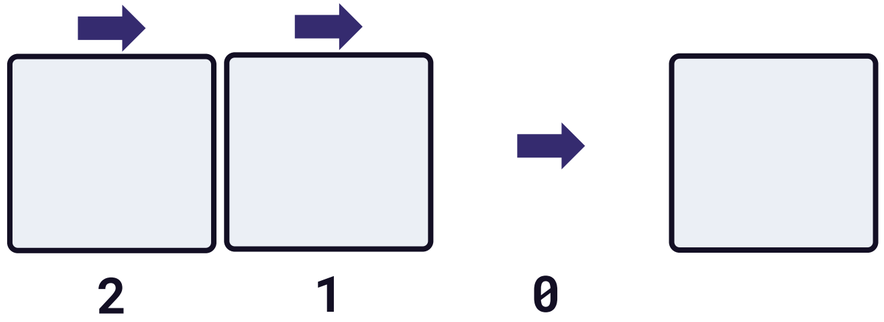
`push()`



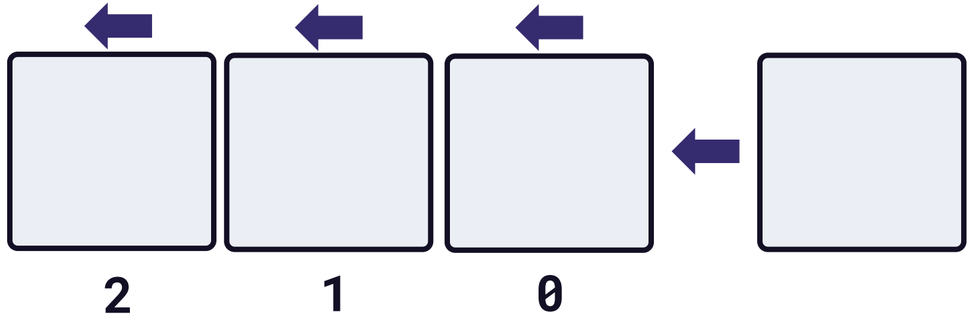
`pop()`



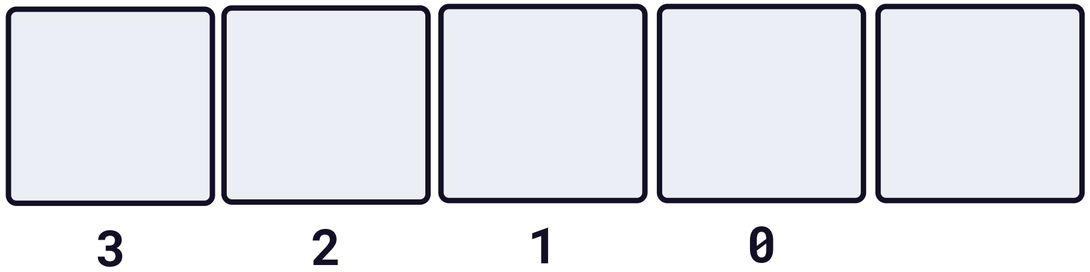
`shift()`



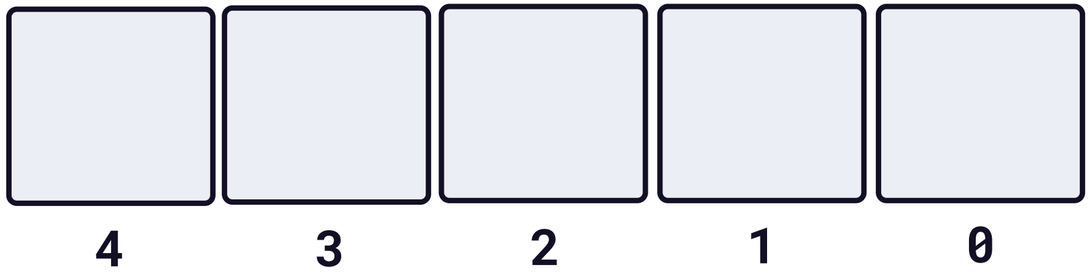
`unshift()`



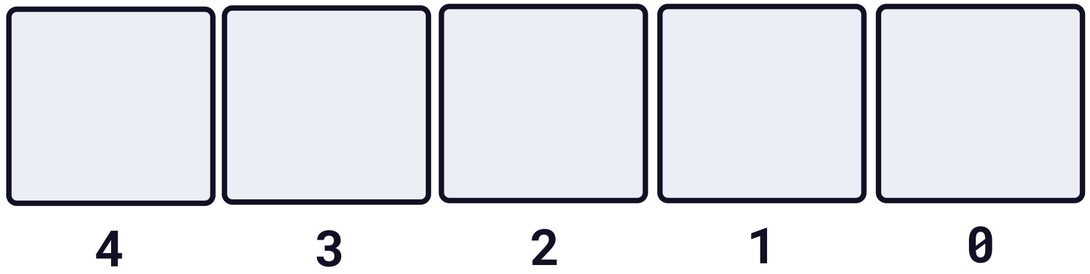
Unshift



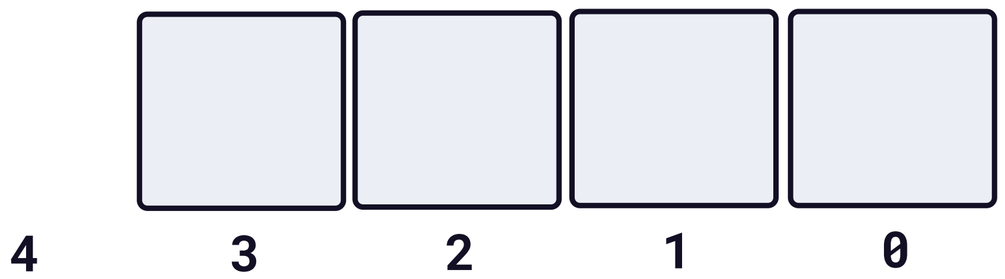
Unshift



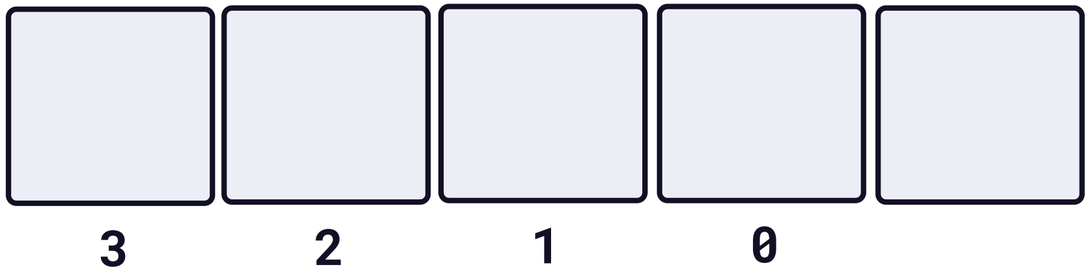
Shift and Unshift



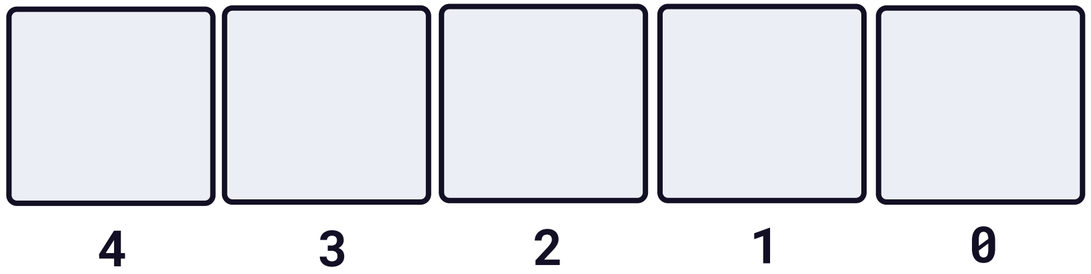
Shift and Unshift



Shift and Unshift



Shift and Unshift



Overview: Array Modules

Module 5: Arrays

- Arrays are Objects
- Creating Arrays
- Identifying Arrays
- Accessing Array Elements
- Cutting Up Arrays
- Putting Together Arrays
- Cloning Arrays
- Sorting Arrays
- Sparse Arrays
- Other Array Types

Module 6: Iteration Deep Dive

- Array Iteration
- For Loops
- Summarizing Arrays
- Searching Through Arrays
- Map

Overview: Array Modules

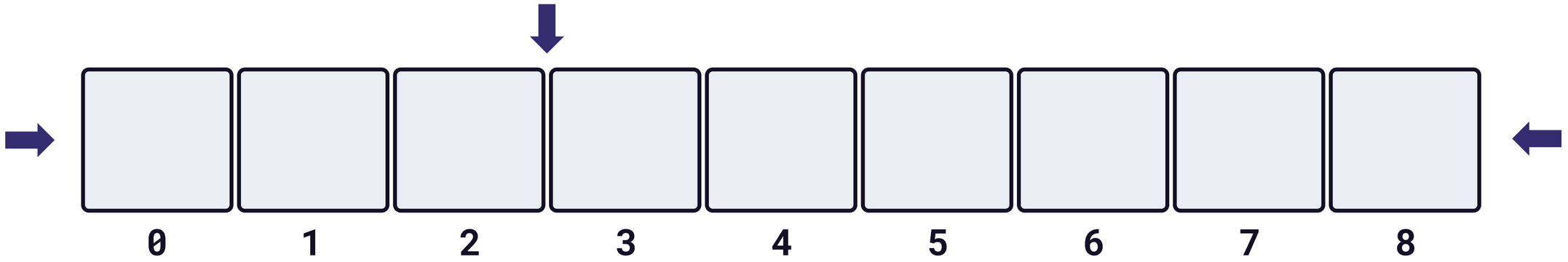
Module 5: Arrays

- Arrays are Objects
- Putting Together Arrays
- Creating Arrays
- Cloning Arrays
- Identifying Arrays
- Sorting Arrays
- Accessing Array Elements
- Sparse Arrays
- Cutting Up Arrays
- Other Array Types

Module 6: Iteration Deep Dive

- Array Iteration
- For Loops
- Summarizing Arrays
- Searching Through Arrays
- Map

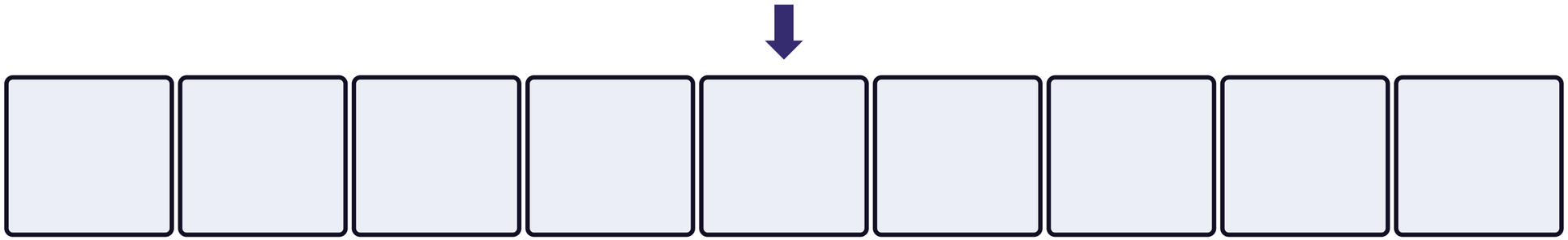
In This Video



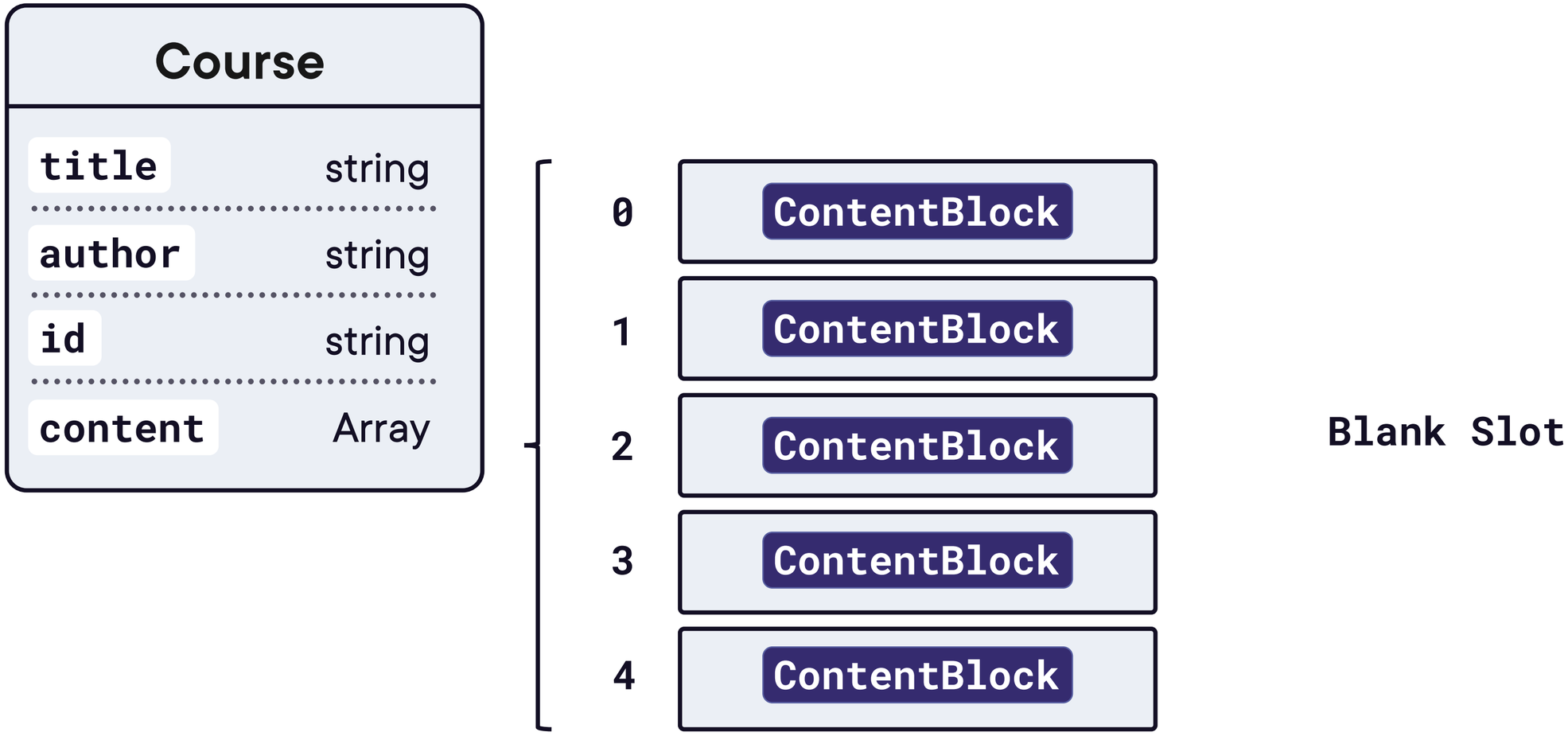
Up Next:

Cutting Up Arrays

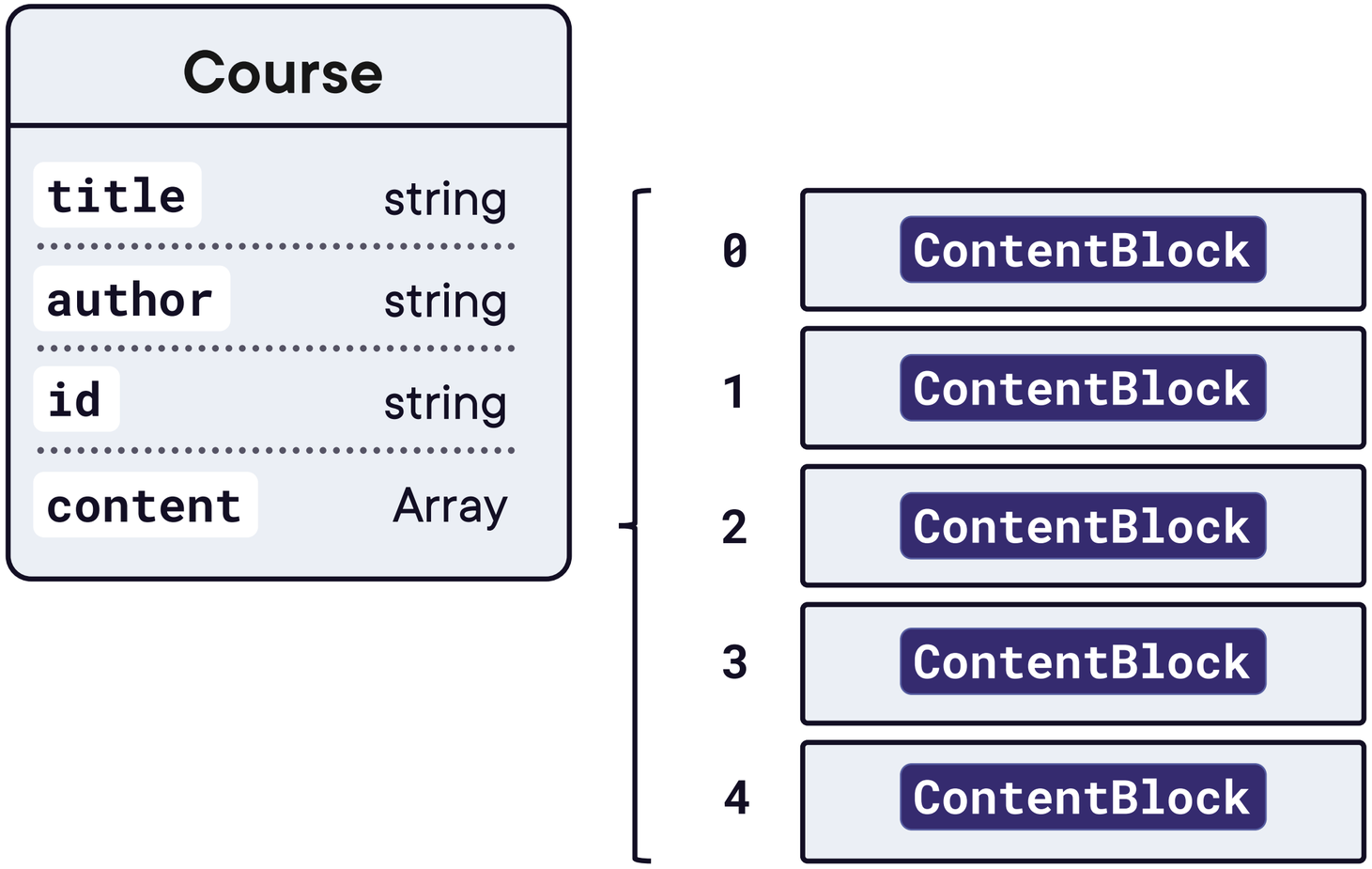
Removing Elements from Arrays



Course Object Structure



Deleting an Array Element



Functions for Cutting Up Arrays

slice

```
myArray.slice()
```

splice

```
myArray.splice()
```

splice

to unite (two ropes or two parts of a rope) by interweaving the strands;

to unite, link, or insert as if by splicing

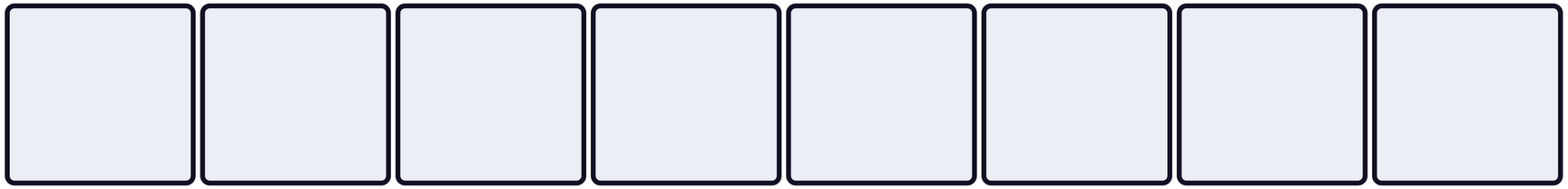
Citation: https://www.merriam-webster.com/dictionary/splice

slice

to cut with or as if with a knife

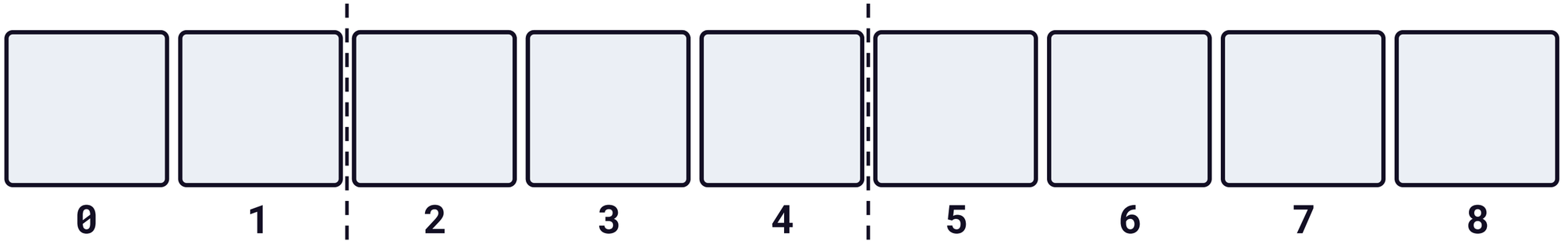
Citation: https://www.merriam-webster.com/dictionary/slice

Splice



Slice

```
myArray.slice(2, 4)
```



Destructive and Non-Destructive Functions

Splice

```
const result = myArray.splice(...)
```

destructive

```
const result = myArray.toSpliced(...)
```

non-destructive

Slice

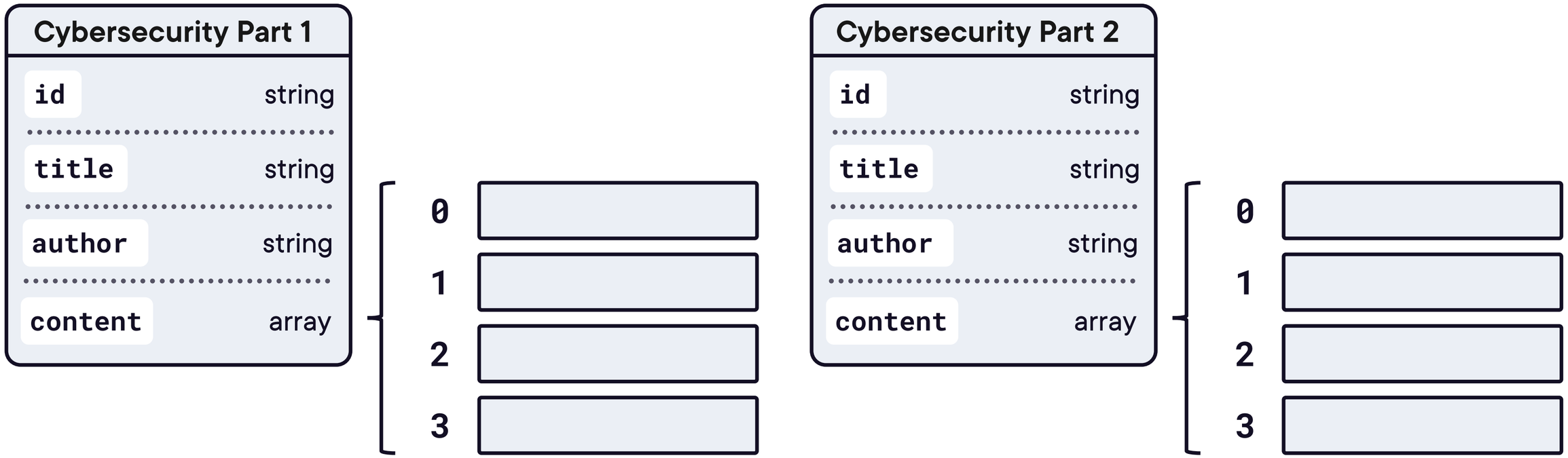
```
const result = myArray.slice(...)
```

non-destructive

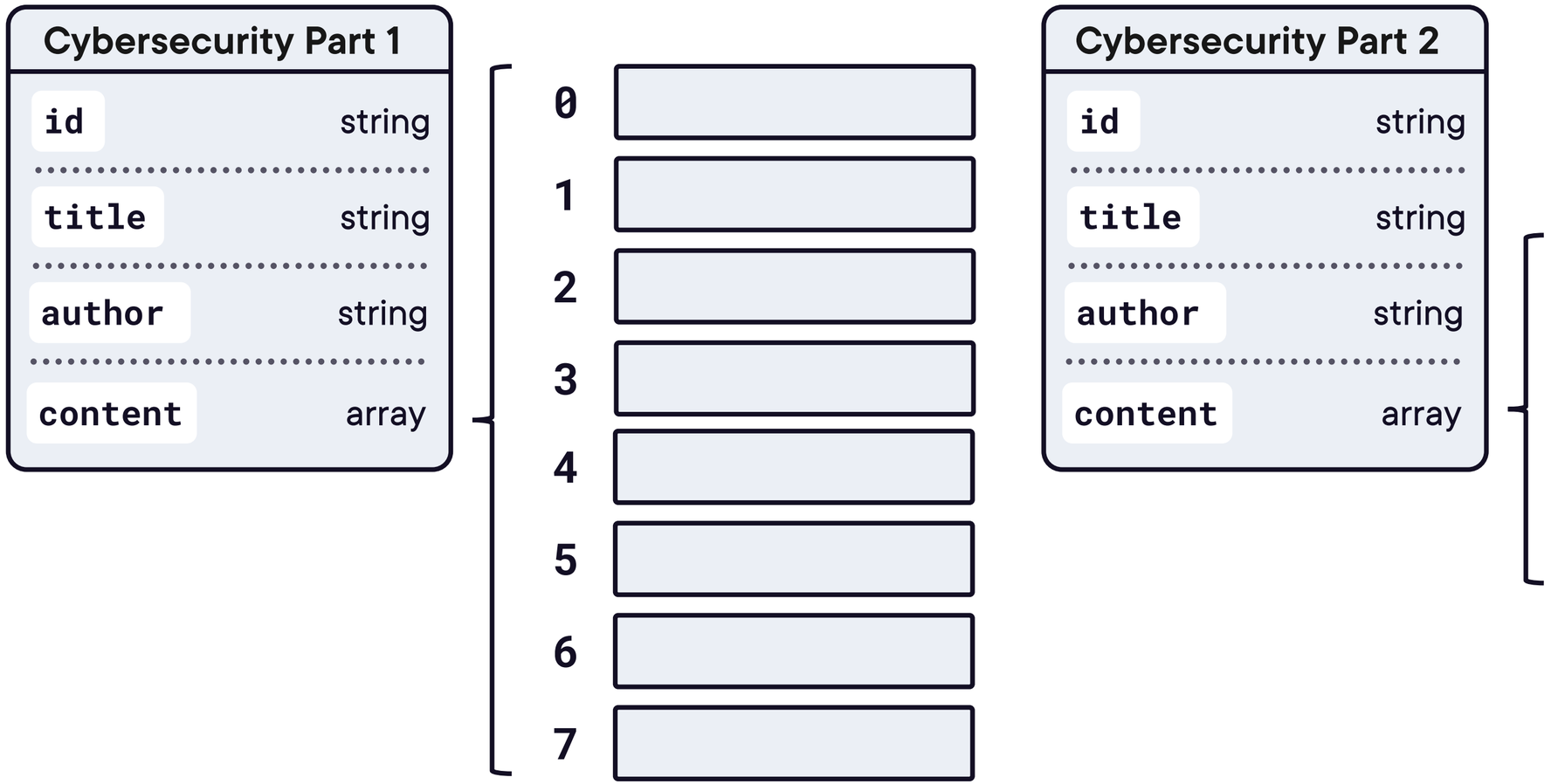
Up Next:

Putting Together Arrays

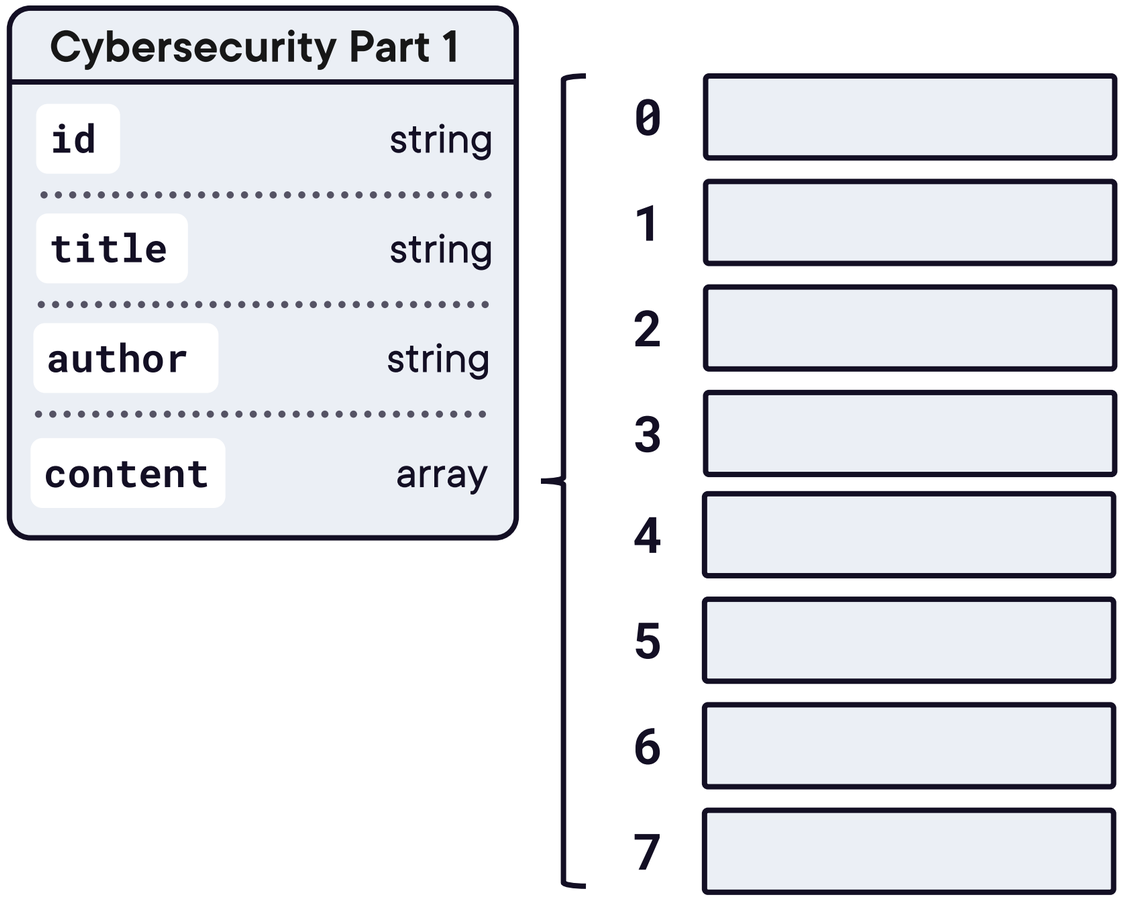
Combining Two Courses



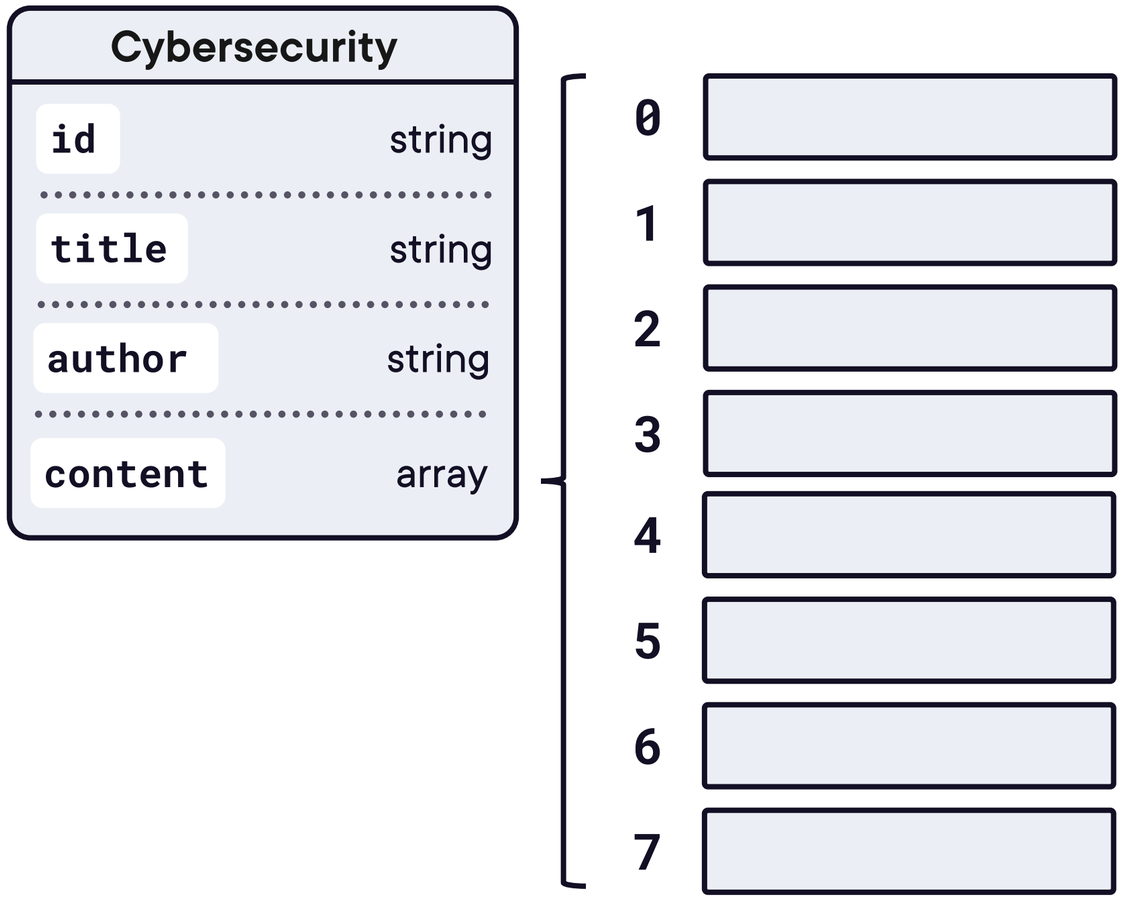
Combining Two Courses



Combining Two Courses



Combining Two Courses



Functions for Combining Arrays

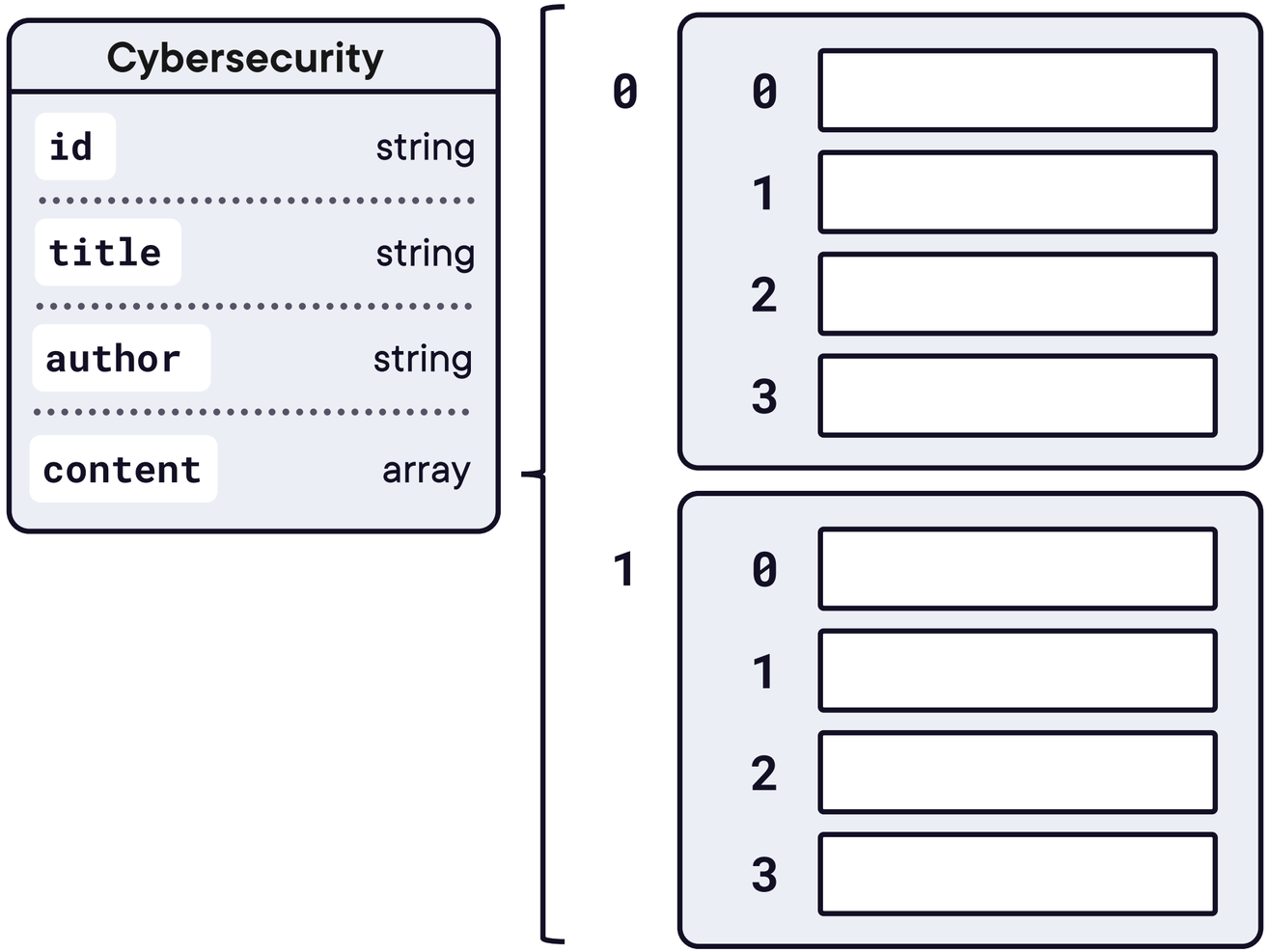
concat

```
array1.concat(array2)
```

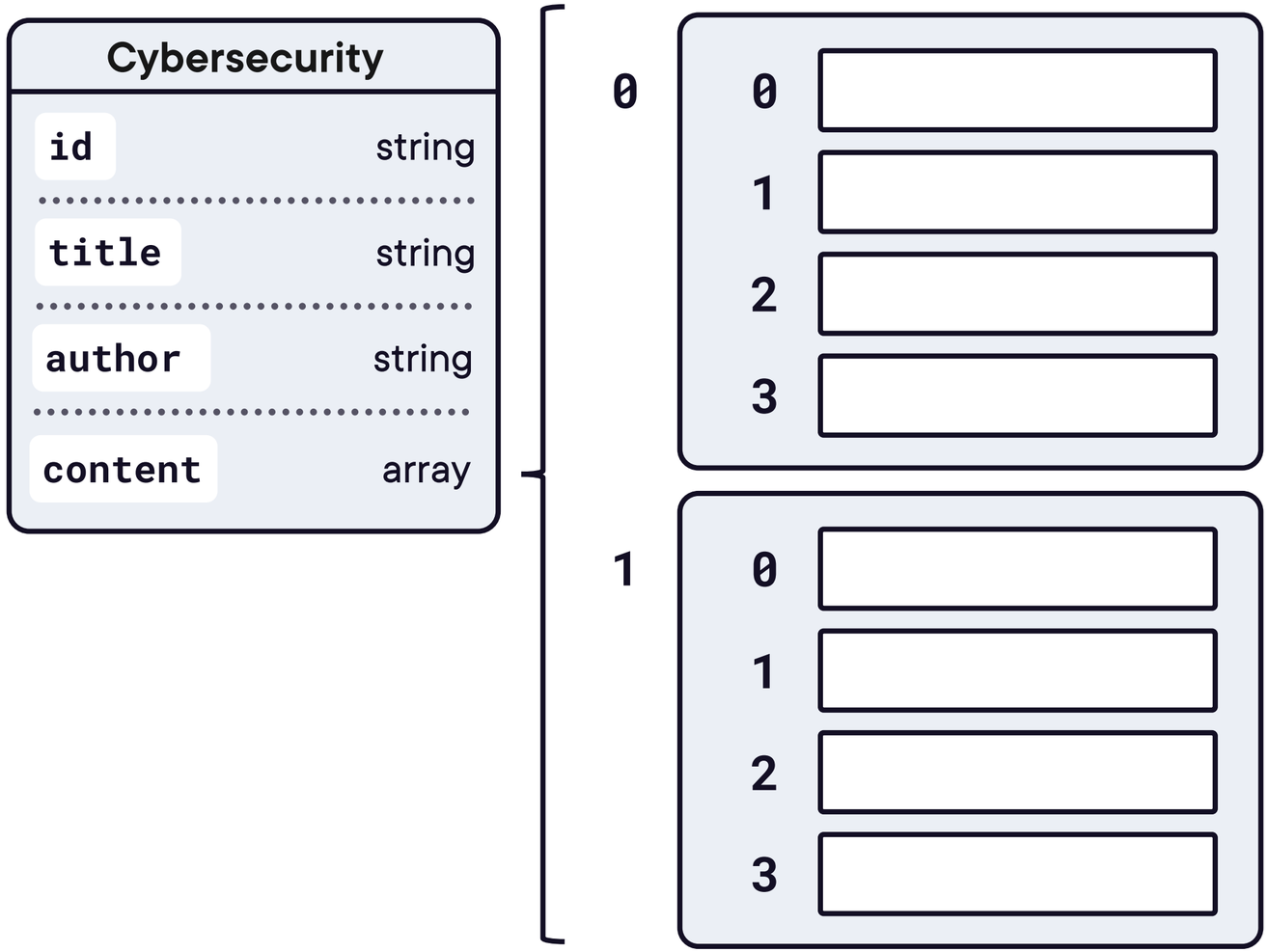
spread operator

```
[...array1, ...array2]
```

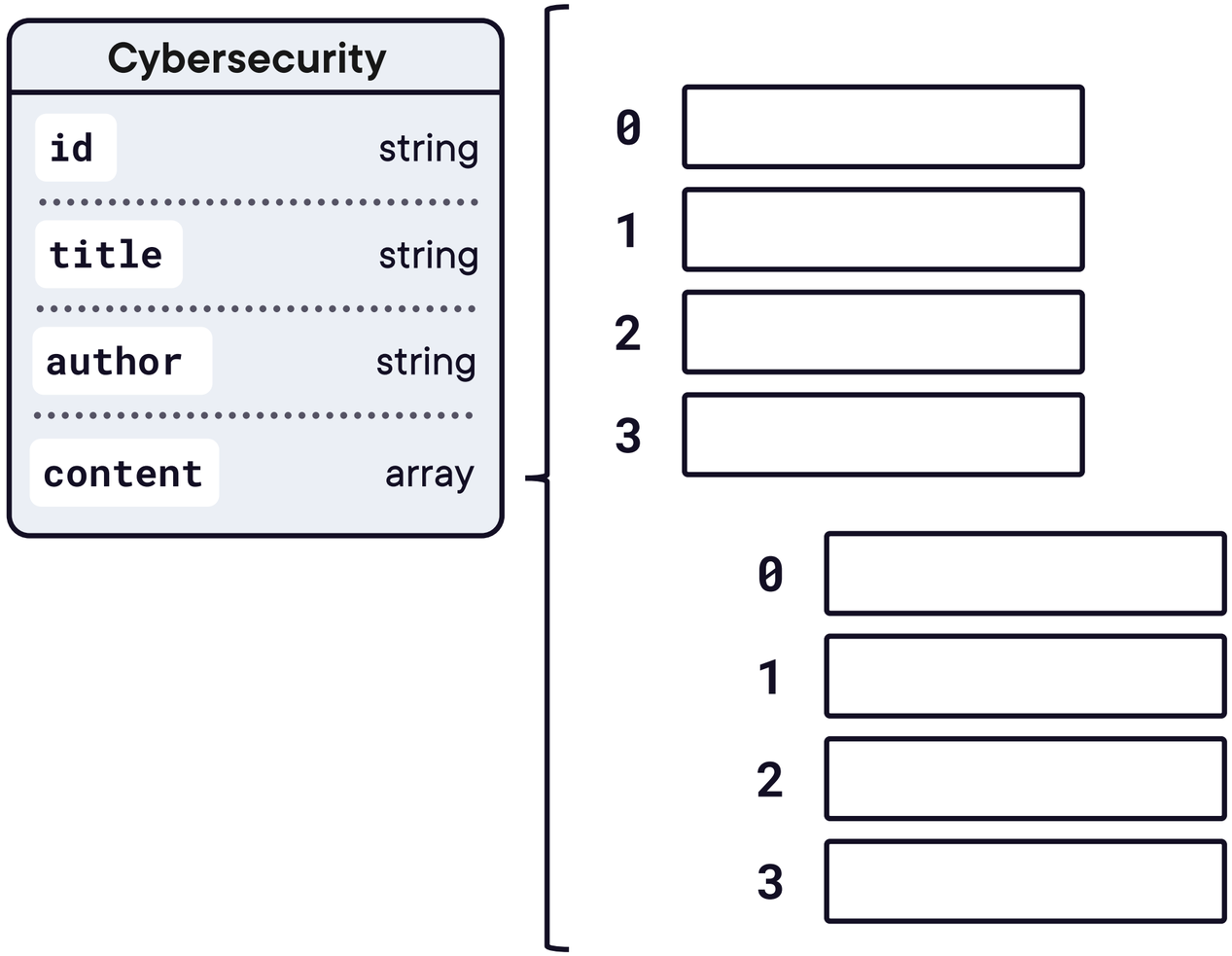
Accidental Two-Dimensional Arrays



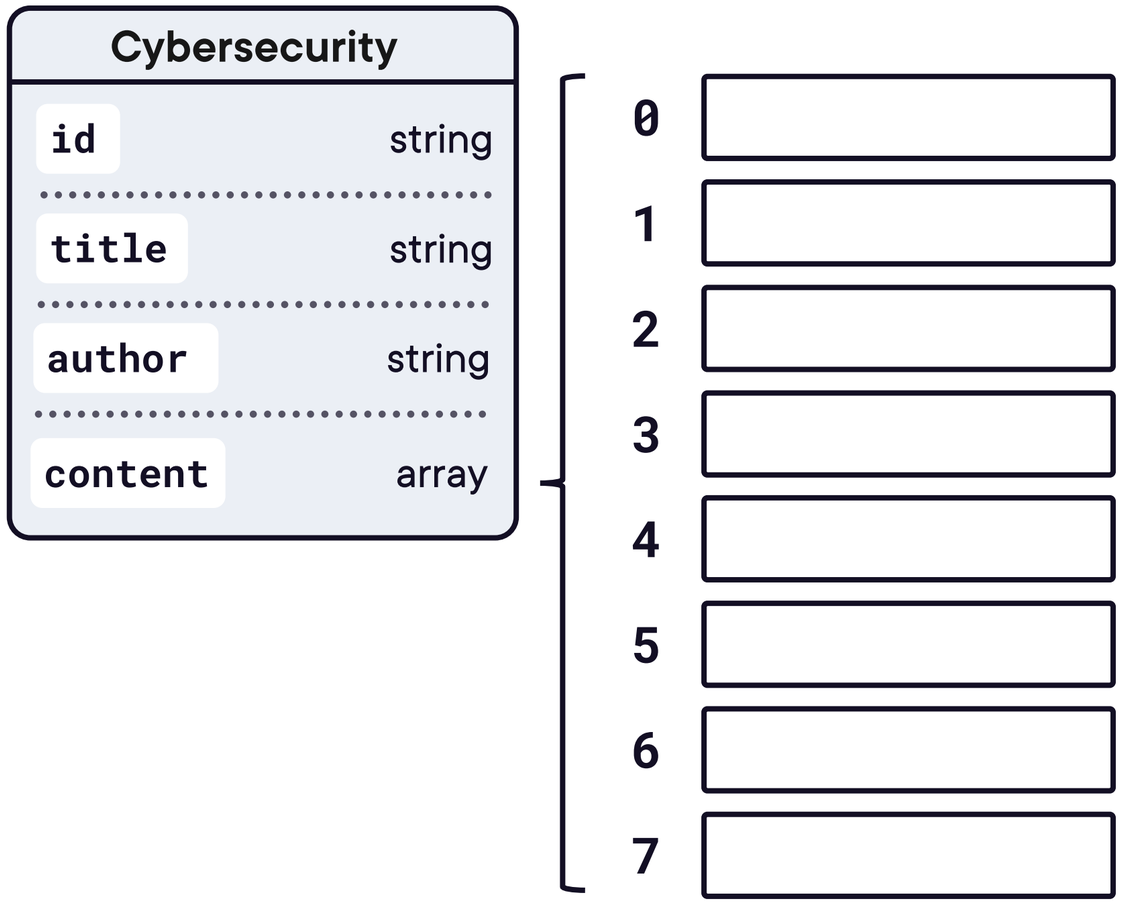
The Spread Operator Unpacks Arrays



The Spread Operator Unpacks Arrays



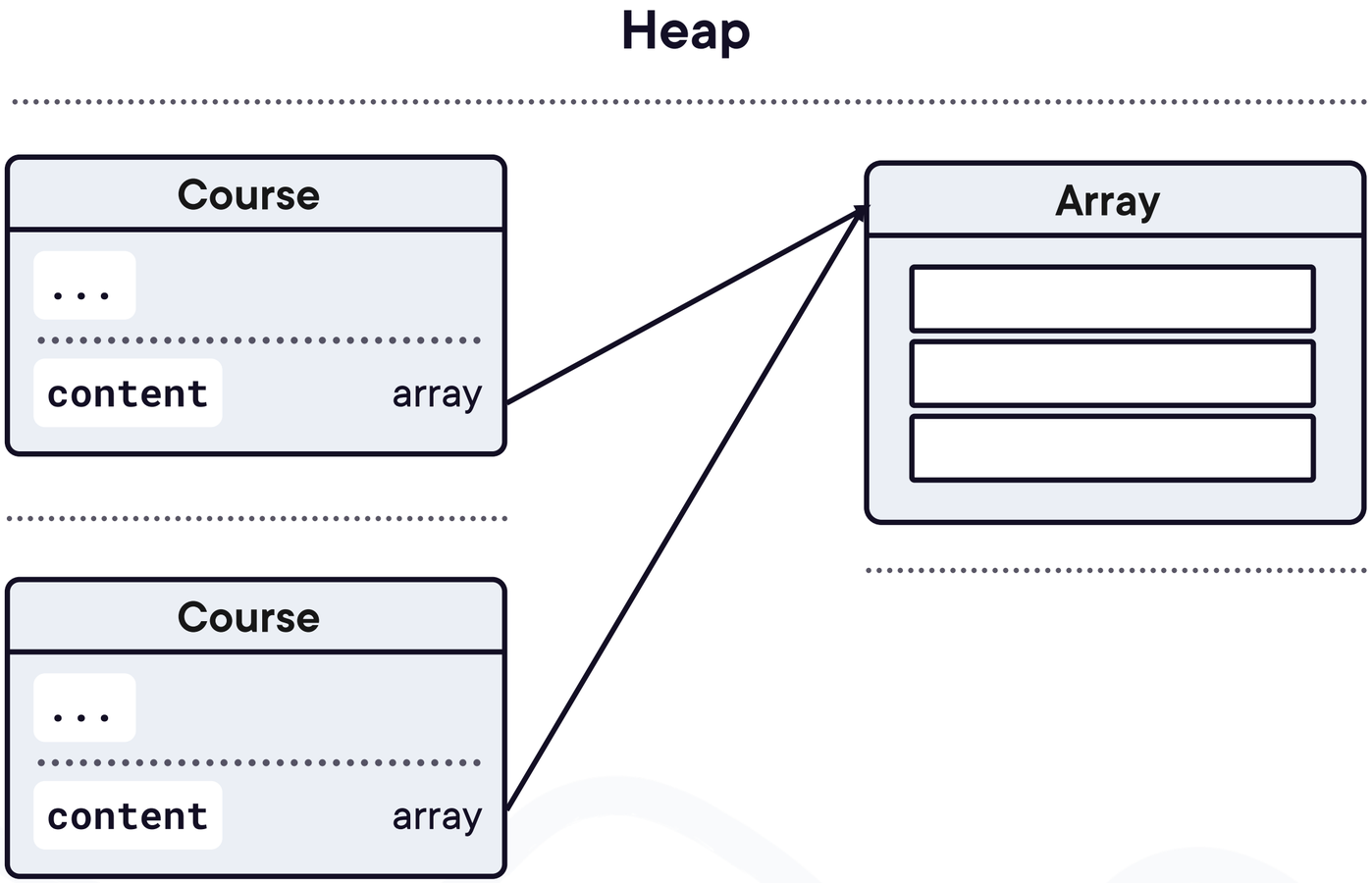
The Spread Operator Unpacks Arrays



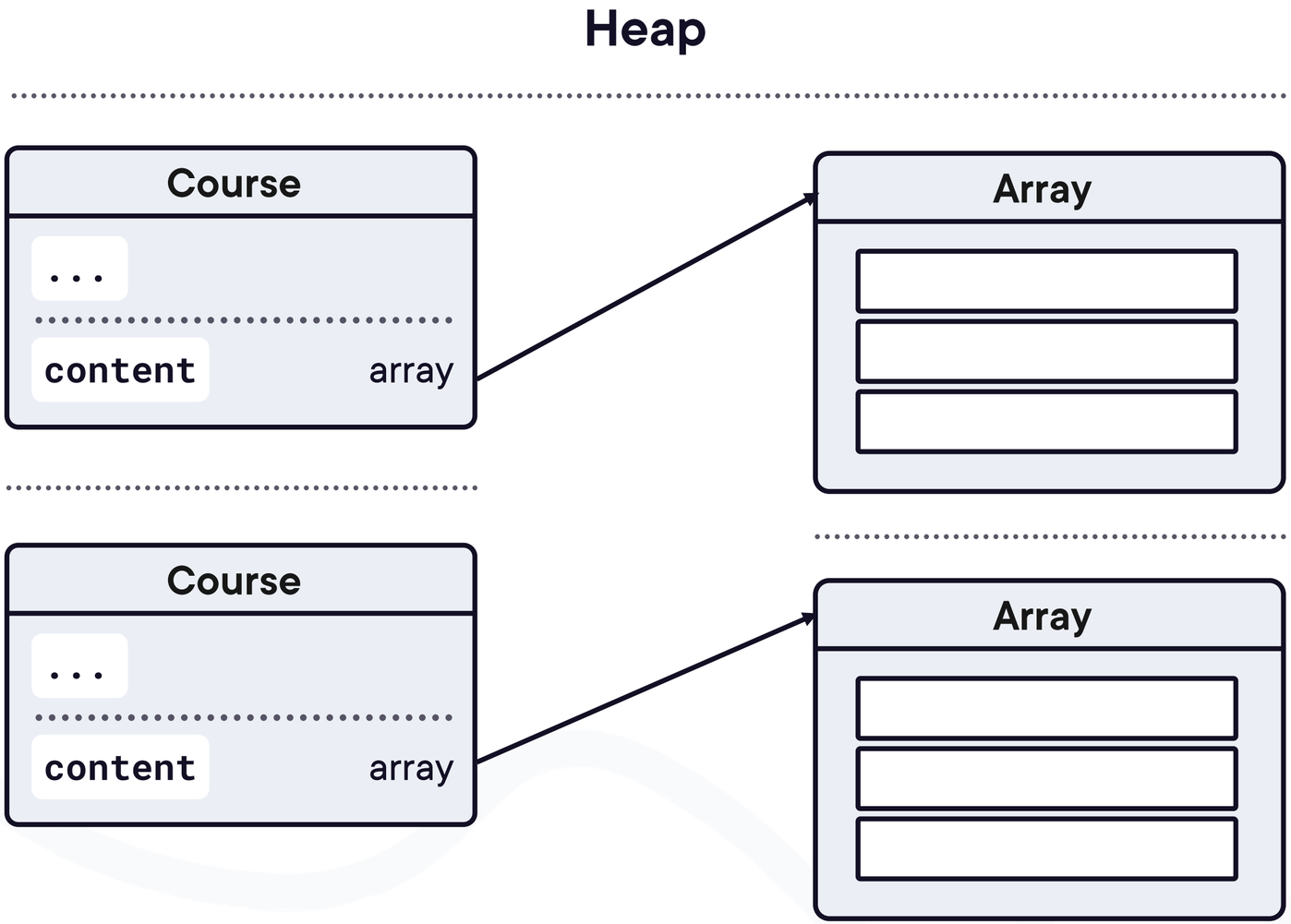
Up Next:

Cloning Arrays

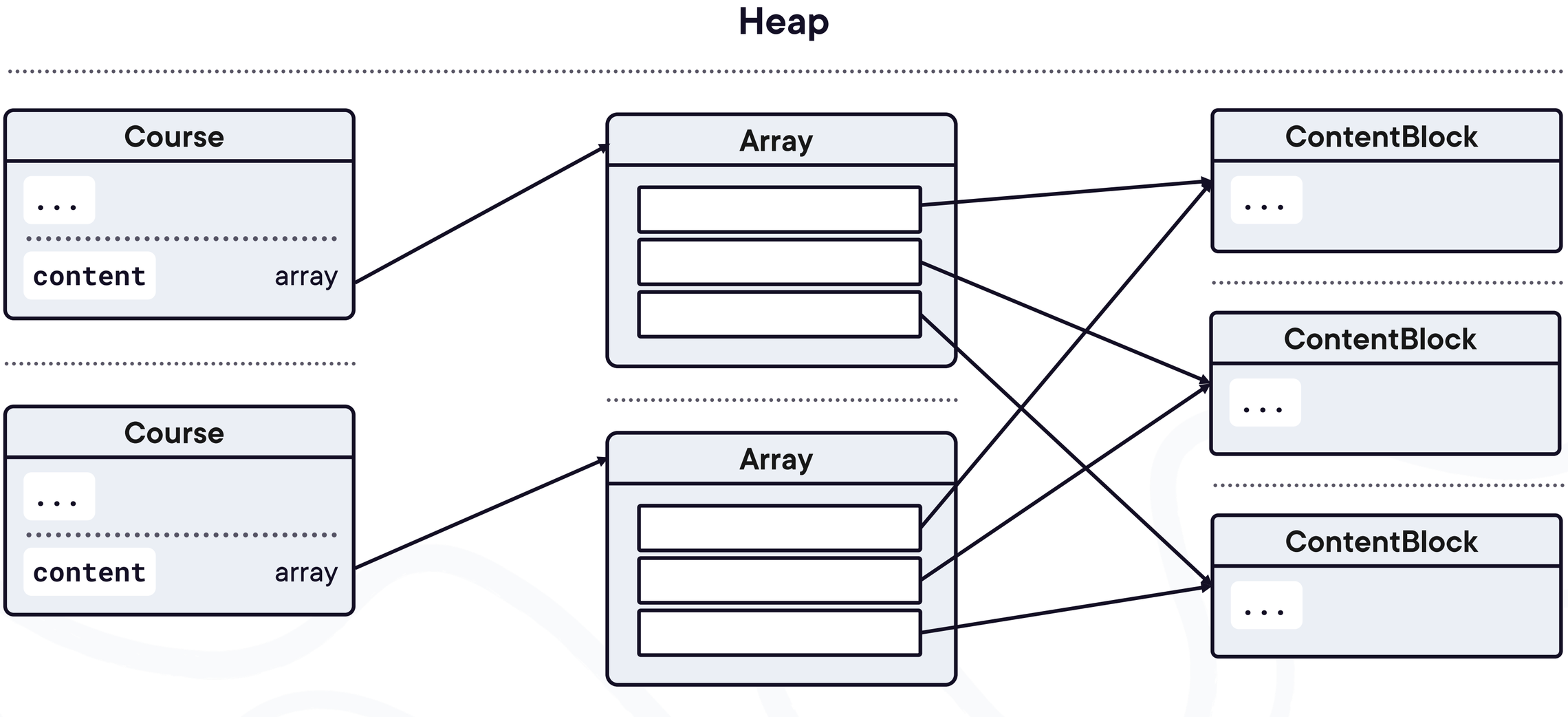
Dealing With References When Cloning Objects



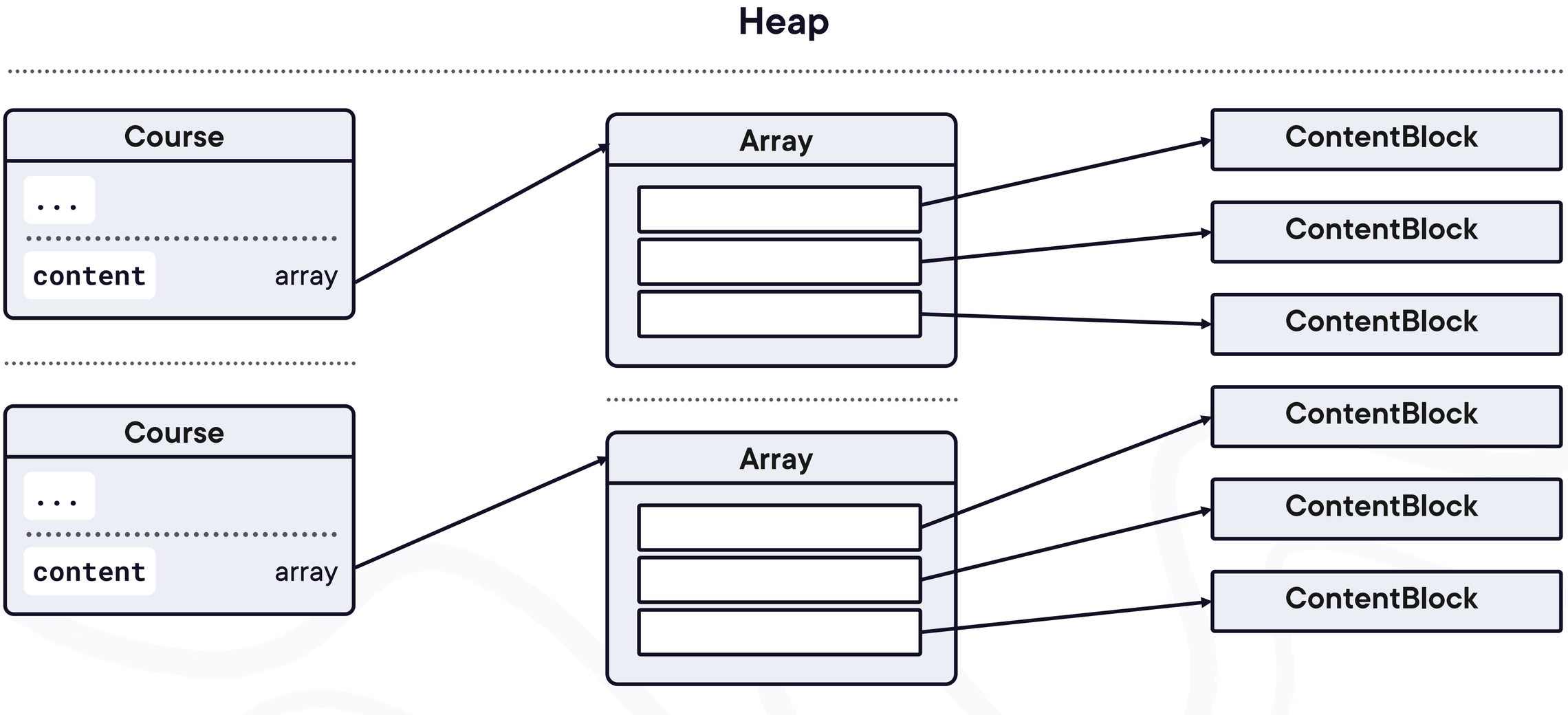
Dealing With References When Cloning Objects



Dealing With References When Cloning Objects



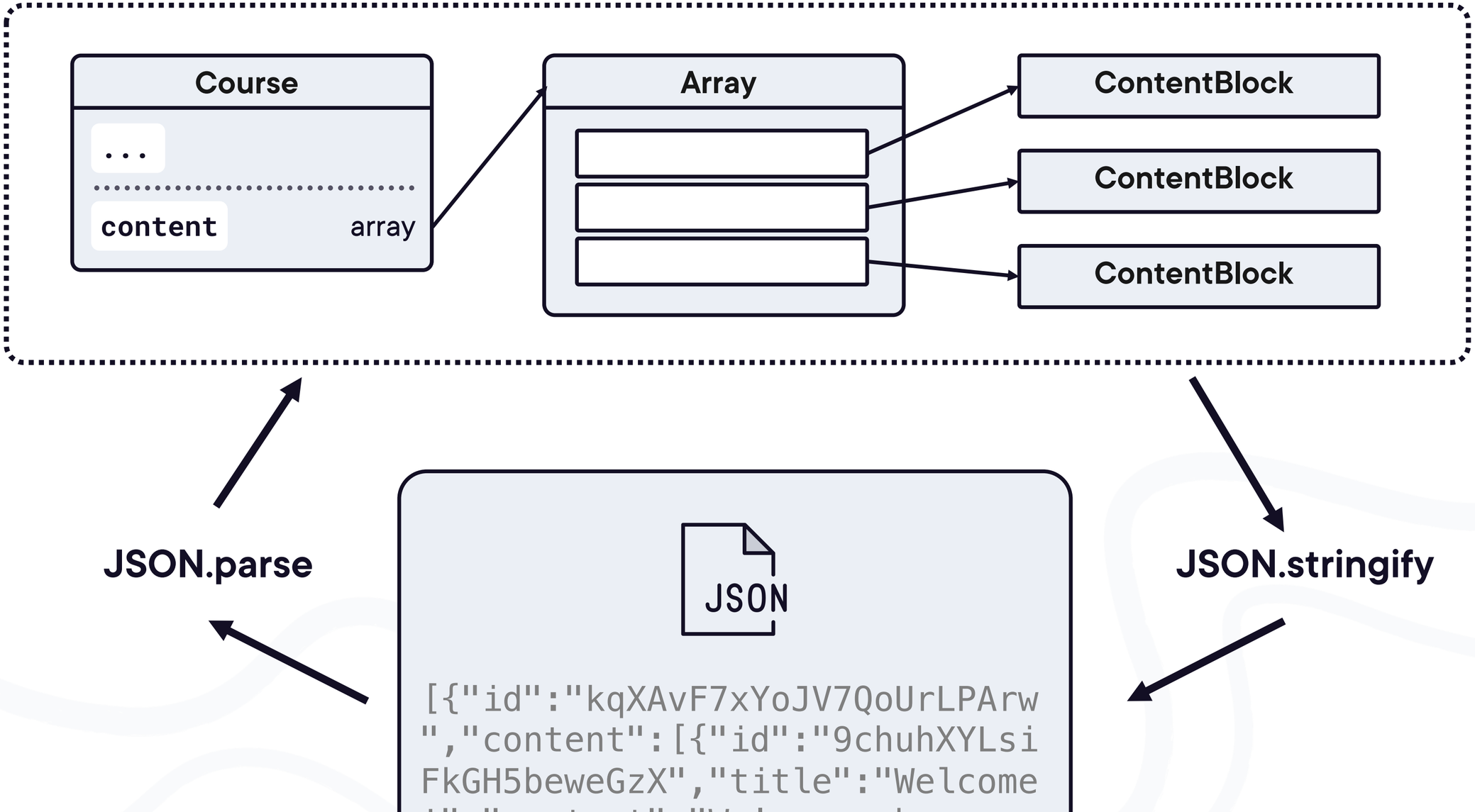
Dealing With References When Cloning Objects



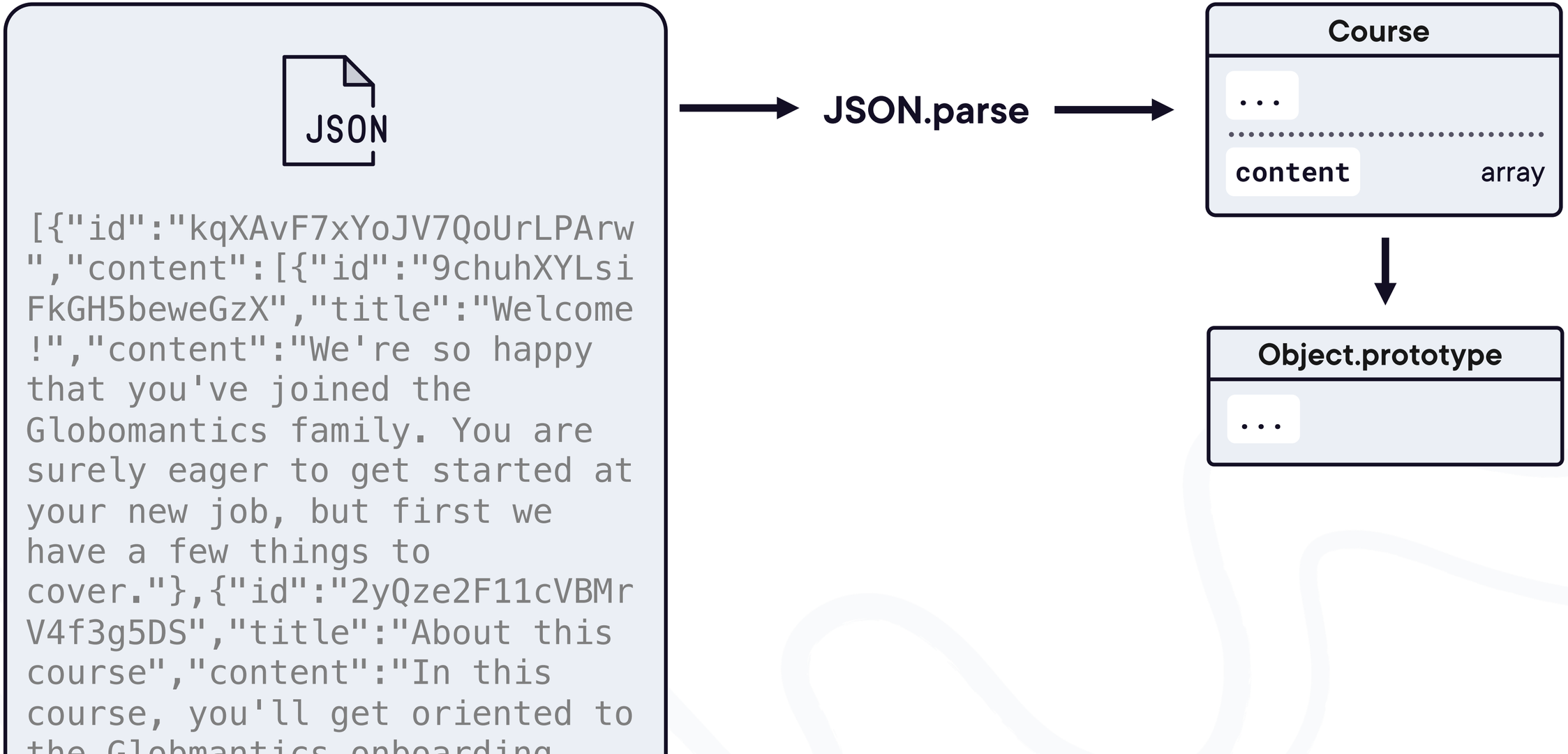
A full discussion of deep copying is outside the scope of this course

A full discussion of deep copying is outside the scope of this course*

JSON Utilities



JSON Utilities



Up Next:

Sorting Arrays

Functions for Sorting Arrays

reverse

```
array.reverse()
```

toReversed

```
array.toReversed()
```

sort

```
array.sort()
```

toSorted

```
array.toSorted()
```

Expectations of Array.sort Comparison Function

```
array.sort(comparisonFunction);
```

Two arguments
Convention: a & b

Compare a to b
Two random entries

Return 1, -1, or 0
+1: a is higher; -1: b is higher

```
const comparisonFunction = function(a, b) {
   if (a.title > b.title) return 1;
   if (b.title > a.title) return -1;
   return 0;
};
```

Overview: Array Modules

Module 5: Arrays

- Arrays are Objects
- Creating Arrays
- Identifying Arrays
- Accessing Array Elements
- Cutting Up Arrays
- Putting Together Arrays
- Cloning Arrays
- Sorting Arrays
- Sparse Arrays
- Other Array Types

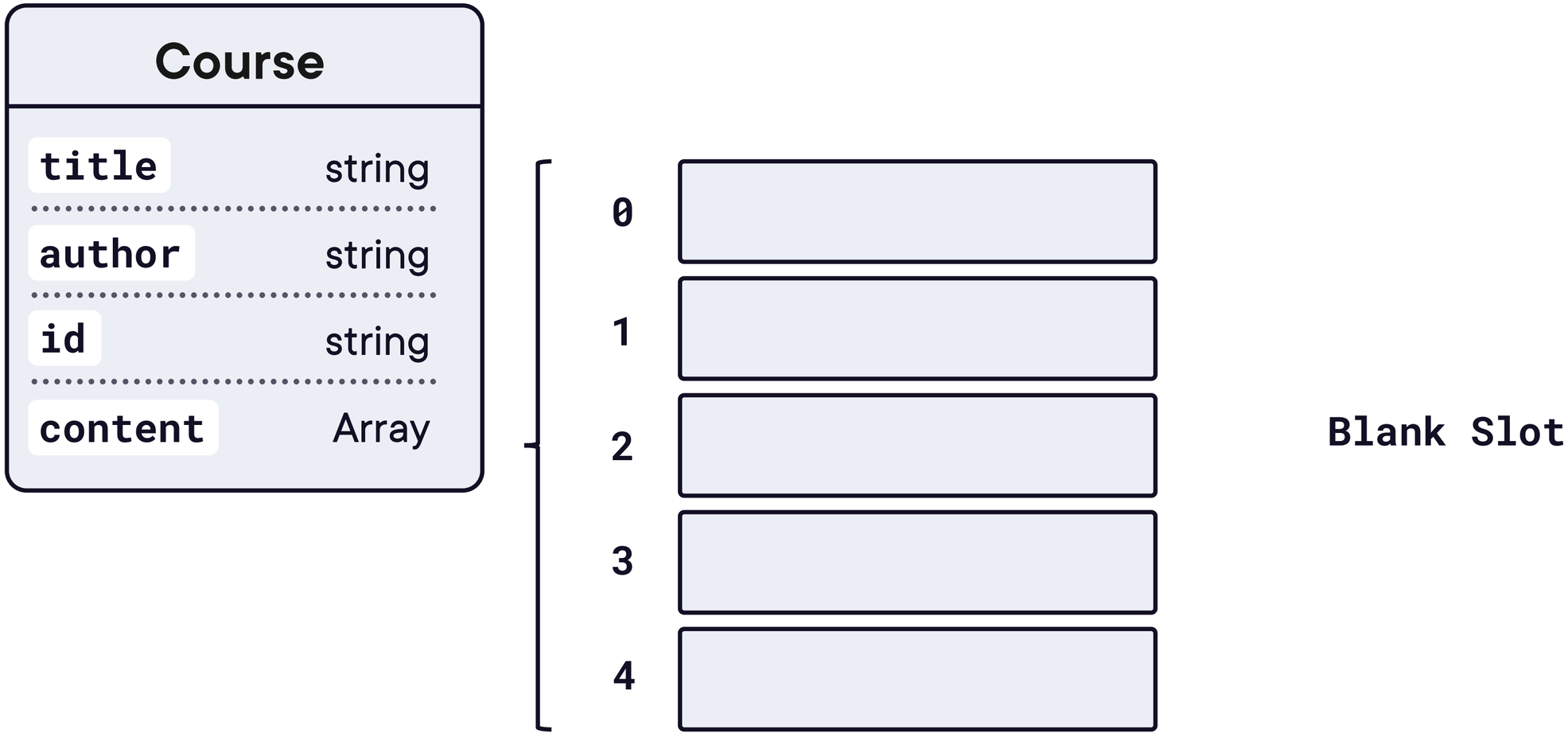
Module 6: Iteration Deep Dive

- Array Iteration
- For Loops
- Summarizing Arrays
- Searching Through Arrays
- Map

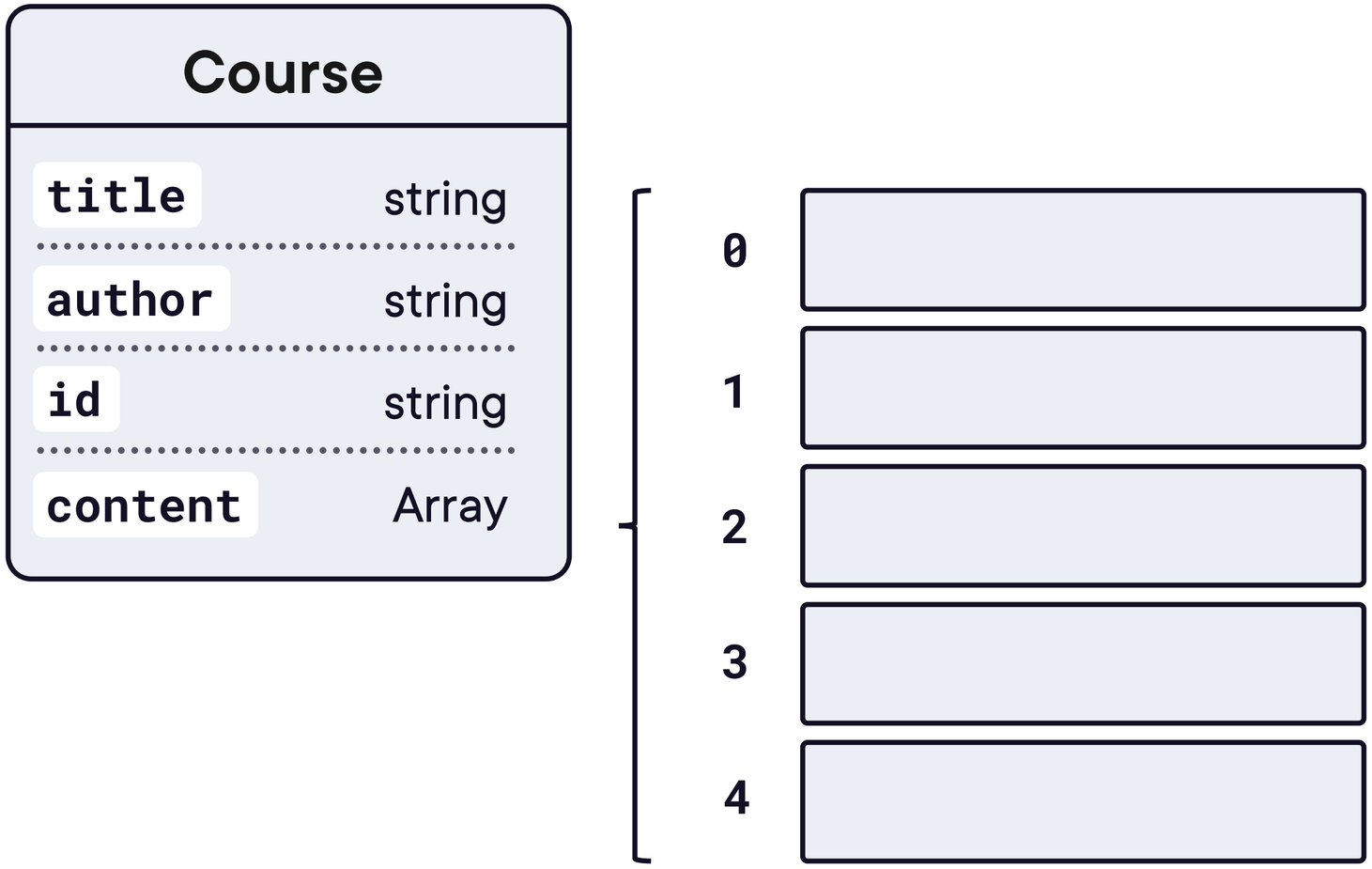
Up Next:

Sparse Arrays

Course Object Structure



Deleting an Array Element



Ways to Make a Sparse Array

Deleting an array element

```
delete array[3];
```

Initializing an array with a length but not filling it

```
let array = [3];
array[1] = "banana";
```

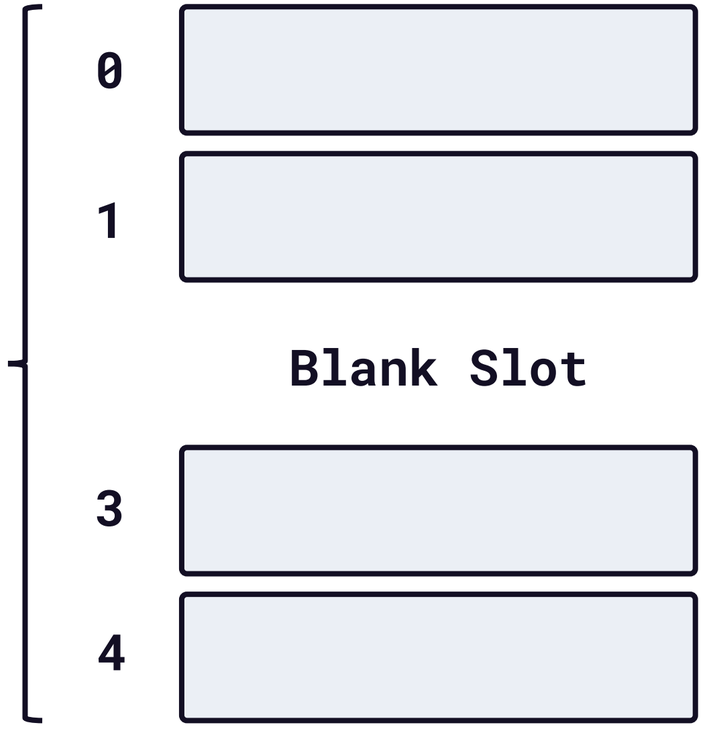
Adding to an array at an index greater than its length

```
let fruits = ["apple", "banana”];
fruits[5] = ”cherry";
```

Sparse arrays are handled inconsistently

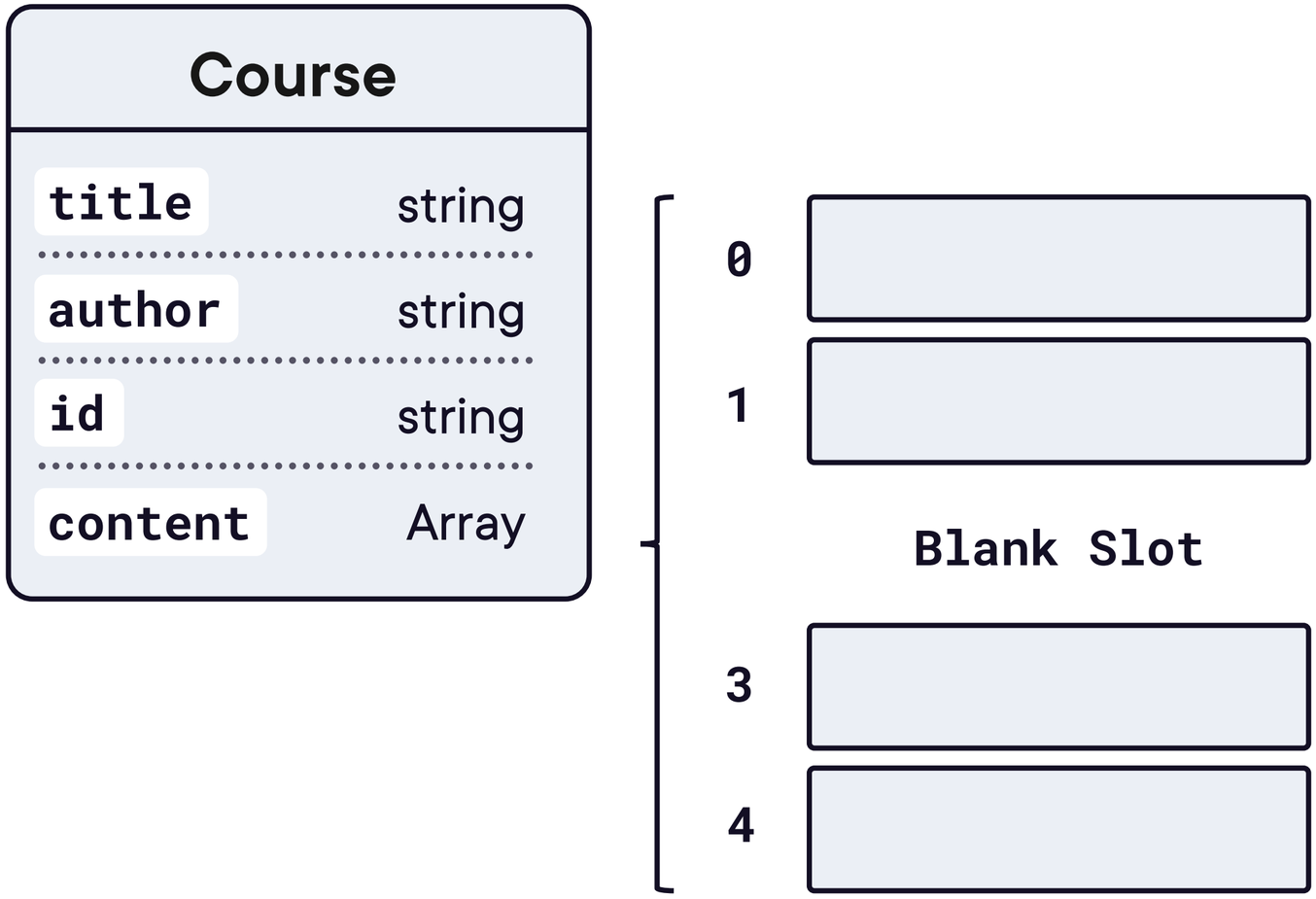
Deleting an Array Element

Course	
`title`	string
`author`	string
`id`	string
`content`	Array




```
course.content.forEach();
```

Deleting an Array Element

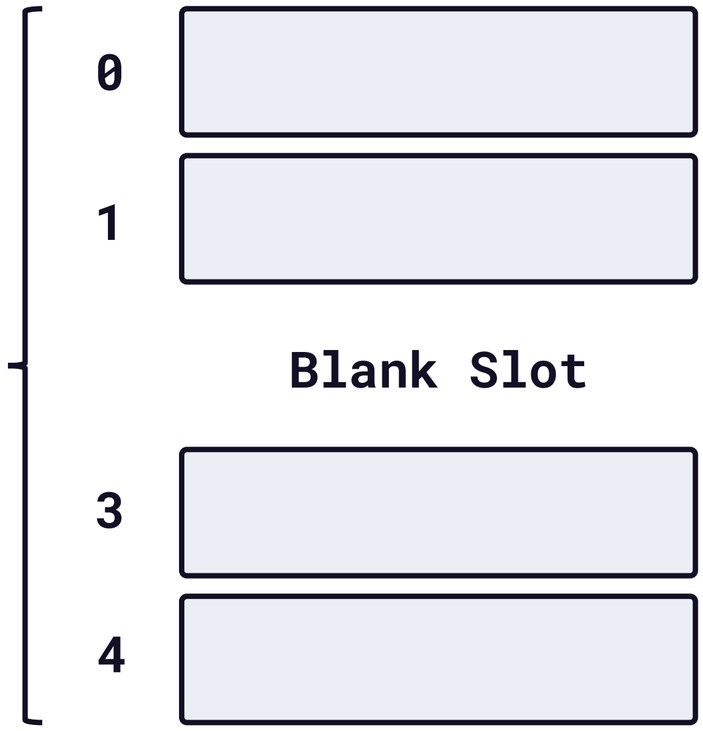



```
course.content.forEach();
```

ContentBlock

Deleting an Array Element

Course	
`title`	string
`author`	string
`id`	string
`content`	Array

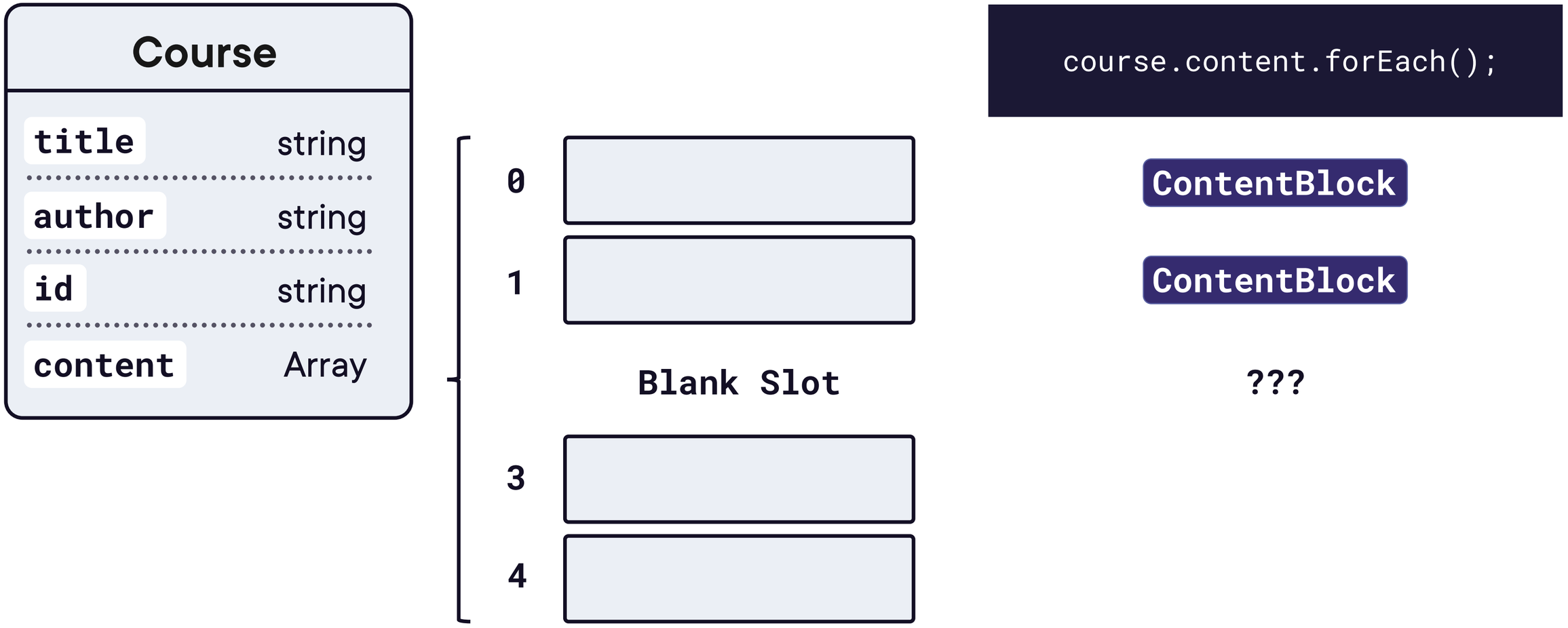



```
course.content.forEach();
```

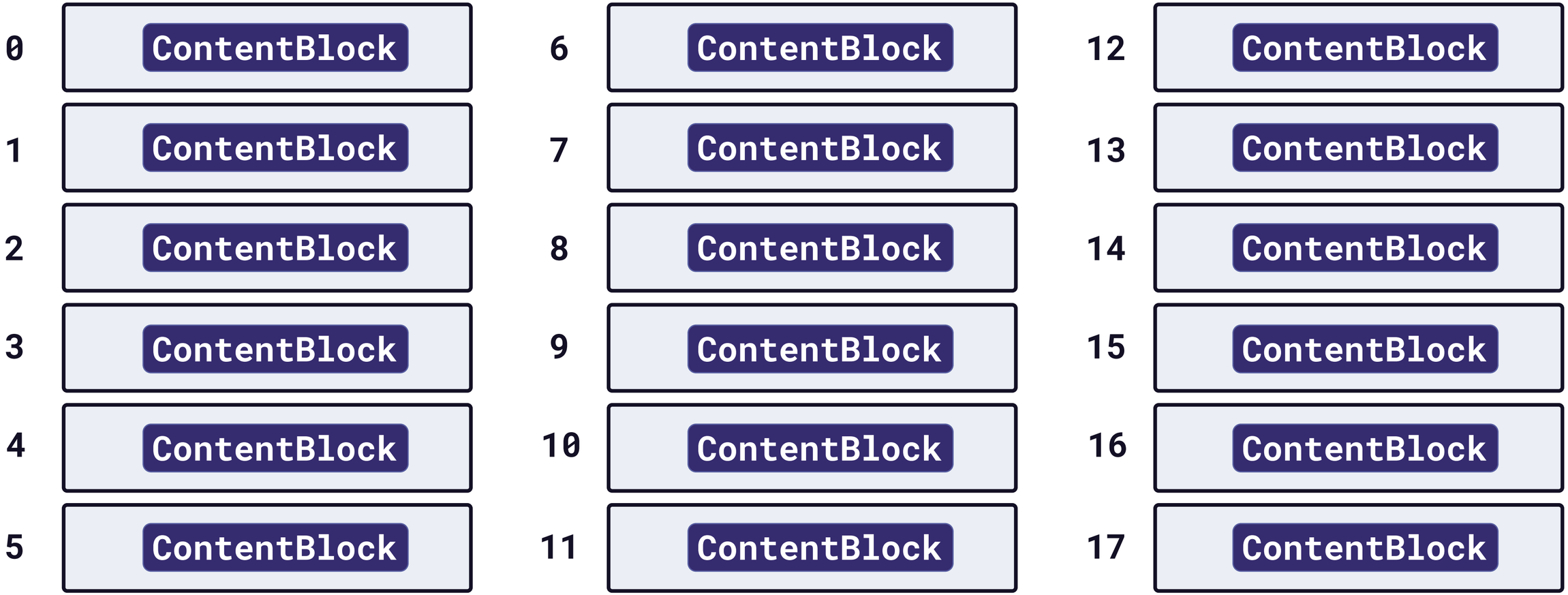
`ContentBlock`

`ContentBlock`

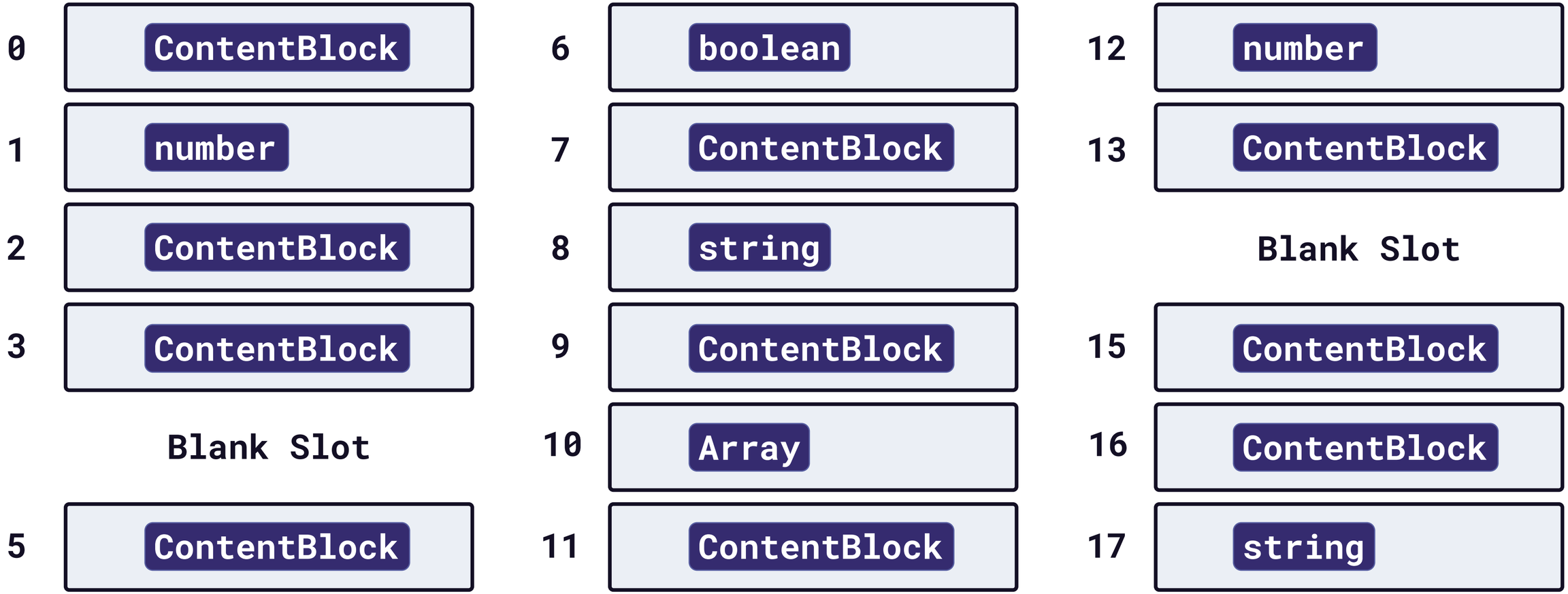
Iterating Through a Sparse Array



Effective Arrays Contain the Same Type of Content



Inconsistent Arrays are Chaotic to Use



Up Next:

Maps, Sets, and Typed Arrays

Maps, Sets, and Typed Arrays

Comparison of JavaScript Collection Types

	Arrays	Maps	Sets	Typed Arrays
Comprised of	• keys • values	• keys • values	unique values	• keys • values
Are ordered	in any way	insertion order	insertion order	in any way
Keys are	zero-indexed numbers	anything	n/a	zero-indexed numbers
Values are	anything	anything	anything	only specified type

Types of Typed Arrays

Int8Array

Float64Array

Uint8ClampedArray

Int32Array

BigUint64Array

BigInt64Array

Float32Array

Float64Array

Uint8Array

Uint16Array

Uint32Array

Int16Array

Overview: Array Modules

Module 5: Arrays

- Arrays are Objects
- Creating Arrays
- Identifying Arrays
- Accessing Array Elements
- Cutting Up Arrays
- Putting Together Arrays
- Cloning Arrays
- Sorting Arrays
- Sparse Arrays
- Other Array Types

Module 6: Iteration Deep Dive

- Array Iteration
- For Loops
- Summarizing Arrays
- Searching Through Arrays
- Map

Overview: Array Modules

Module 5: Arrays

- Arrays are Objects
- Putting Together Arrays
- Creating Arrays
- Cloning Arrays
- Identifying Arrays
- Sorting Arrays
- Accessing Array Elements
- Sparse Arrays
- Cutting Up Arrays
- Other Array Types

Module 6: Iteration Deep Dive

- Array Iteration
- For Loops
- Summarizing Arrays
- Searching Through Arrays
- Map