

# Table of Contents

1	Array Package Documentation
1.1	Creating Arrays
1.2	Array Operations
1.3	Sorting
1.4	Array Views
1.5	Fixed Arrays
1.6	Utilities

# Array Package Documentation

This package provides array manipulation utilities for MoonBit, including fixed-size arrays (FixedArray), dynamic arrays (Array), and array views (ArrayView/View).

## Creating Arrays

There are several ways to create arrays in MoonBit:

```
1
2  test "array creation" {
3
4      let arr1 = [1, 2, 3]
5      inspect(arr1, content="[1, 2, 3]")
6
7
8      let arr2 = Array::makei(3, i => i * 2)
9      inspect(arr2, content="[0, 2, 4]")
10
11
12     let arr3 = Array::from_iter("hello" iter())
13     inspect(arr3, content="['h', 'e', 'l', 'l', 'o']")
14 }
```

## Array Operations

Common array operations include mapping, filtering, and folding:

```
1
2  test "array operations" {
3      let nums = [1, 2, 3, 4, 5]
4
5
6      let neg_evens = nums filter_map(x => if x % 2 == 0 { Some(-x) } else {
7      inspect(neg_evens, content="[-2, -4]")
8
9
10     let sum = nums fold(init=0, (acc, x) => acc + x)
11     inspect(sum, content="15")
12
13
14     let last = nums last()
15     inspect(last, content="Some(5)")
16 }
```

## Sorting

The package provides various sorting utilities:

```

1
2  test "sorting" {
3    let arr = [3, 1, 4, 1, 5, 9, 2, 6]
4
5
6    let sorted1 = arr copy()
7    sorted1 sort()
8    inspect(sorted1, content="[1, 1, 2, 3, 4, 5, 6, 9]")
9
10
11   let strs = ["aa", "b", "ccc"]
12   let sorted2 = strs copy()
13   sorted2 sort_by((a, b) => a length() compare(b length()))
14   inspect(
15     sorted2,
16     content=(
17       #|["b", "aa", "ccc"]
18     ),
19   )
20
21
22   let pairs = [(2, "b"), (1, "a"), (3, "c")]
23   let sorted3 = pairs copy()
24   sorted3 sort_by_key(p => p 0)
25   inspect(
26     sorted3,
27     content=(
28       #|[(1, "a"), (2, "b"), (3, "c")]
29     ),
30   )
31 }

```

## Array Views

Array views provide a lightweight way to work with array slices:

```

1
2  test "array views" {
3    let arr = [1, 2, 3, 4, 5]
4    let view = arr[1:4]
5    inspect(view, content="[2, 3, 4]")
6
7
8    let doubled = view map(x => x * 2)
9    inspect(doubled, content="[4, 6, 8]")
10 }

```

## Fixed Arrays

Fixed arrays provide immutable array operations:

```

1
2  test "fixed arrays" {
3      let fixed : FixedArray[_] = [1, 2, 3]
4
5
6      let combined = fixed + [4, 5]
7      inspect(combined, content="[1, 2, 3, 4, 5]")
8
9
10     let has_two = fixed contains(2)
11     inspect(has_two, content="true")
12
13
14     let starts = fixed starts_with([1, 2])
15     inspect(starts, content="true")
16     let ends = fixed ends_with([2, 3])
17     inspect(ends, content="true")
18 }

```

## Utilities

Additional array utilities for common operations:

```

1
2  test "utilities" {
3
4      let words = ["hello", "world"]
5      let joined = words join(" ")
6      inspect(joined, content="hello world")
7
8
9      let nums = [1, 2, 3, 4, 5]
10
11
12     let shuffled = nums shuffle(rand=_ => 1)
13     inspect(shuffled, content="[1, 3, 4, 5, 2]")
14 }

```