

Slices in Go

Array	Slice
<ul style="list-style-type: none">● Has a fixed length defined at compile time;● The length of an array is part of its type, defined at compile time and cannot be changed;● By default an uninitialized array has all elements equal to zero;	<ul style="list-style-type: none">● Has a dynamic length (it can shrink or grow);● The length of a slice is not part of its type and it belongs to runtime;● An uninitialized slice is equal to nil (its zero value is nil).

- Both a slice and an array can contain only the same type of elements;
- We can create a keyed slice like a keyed array;

Slice's Backing (Underlying) Array

- When creating a slice, behind the scenes Go creates a **hidden** array called **Backing Array**.
- The backing array stores the elements, not the slice.
- Go implements a slice as a data structure called **slice header**.

Slice Header contains 3 fields:

1. **the address** of the backing array (pointer).
 2. **the length** of the slice. `len()` returns it.
 3. **the capacity** of the slice. `cap()` returns it.
- Slice Header is the runtime representation of a slice.
 - A **nil slice** doesn't have backing array.