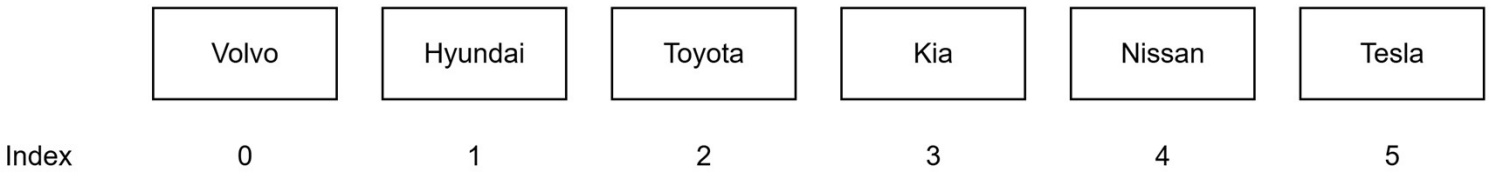
Arrays and Array Lists

Static arrays vs. Dynamic arrays

- There are 2 types of arrays: static arrays and dynamic arrays.
- In Java, static arrays are represented by **arrays**, while dynamic arrays are represented by **array lists**.
- Both types are homogeneous – elements in the array must belong to the same type.
- In Java:
  - Arrays store primitive types and reference types.
  - Array lists store only wrapper, reference types.

Index system of arrays

- Arrays use the zero-based index system.
- Given N is the size of the list:
  - Index of the first element is 0.
  - Index of the last element is $N - 1$.



- Arrays allow random access to elements using their index number → Fast look up, only $O(1)$ time.

Using static arrays

- Static arrays are declared by the element type followed by square brackets.

```
String[] cars;
int[] prices;
```

```
double[] grades;
char[] letters;
```

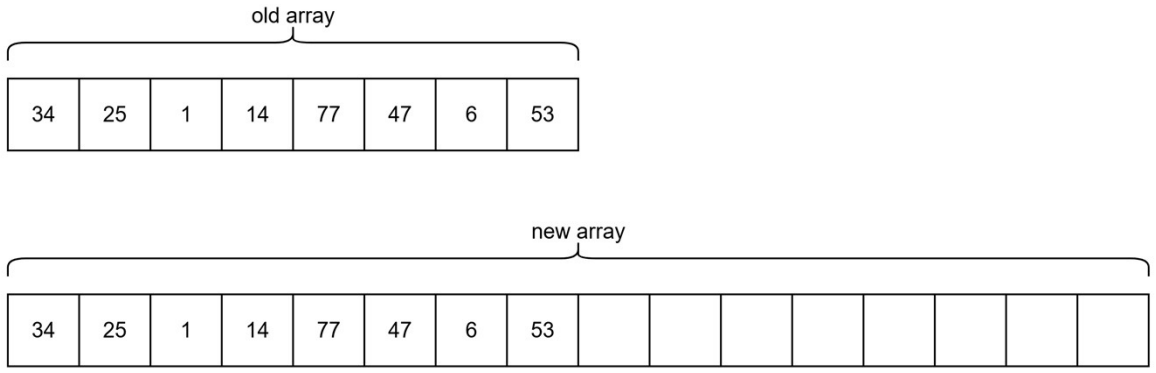
- The list of elements can be placed in a comma-separated list.
- Access the length with the `length` keyword.
- Use the `for` loop to iterate through all the elements in the array.
- Size of static arrays has to be known beforehand and cannot be changed after declaration → May lead to memory waste.

Using dynamic arrays

- A dynamic array is an array with a big improvement: automatic resizing.
- Strengths: fast lookups ($O(1)$), variable size (size grows and shrinks as elements are added), and cache-friendly.
- Weaknesses: slow worst-case appends; insertions and deletions are expensive ($O(N)$).

How are dynamic arrays… **dynamic**?!

- Dynamic arrays automatically make a new, bigger underlying array (usually twice as big) to make room for new elements that need to be appended to it. Each item must be individually copied into the new array. Copying each item over costs $O(N)$ time.



Inserting elements into a dynamic array

- Best case – Adding to the end O(1).
- Worst case:
  - Adding to the front O(N): Every other elements must be shifted one position to the right to make front for this element.
  - Array is at capacity → Requires automatic resizing.