



## Vector

- A vector is contiguous growable array type.
- Vectors are re-sizable arrays.
- Their size is not known at compile time, but they can grow or shrink at any time.
- Vectors allow you to store more than one value in a single data structure that puts all the values next to each other in memory. Vectors can only store values of the same type.
- The data of the vector is allocated on the heap.
- Vectors ensure they never allocate more than `isize::MAX` bytes.
- A simple way of explaining a vector is that it is a container that stores the values like an array, but it has more advantages than an array data structure. It can be increase size dynamically during runtime.

## Creating vector in rust

1. Using `Vec::new()` Method:

```
let v : Vec<i64> = Vec::new();
```

2. Using macro in rust

```
let v = vec!['G','E','E','K','S'];
```

## Accessing a vectors

### Using subscript operator

Similar to the concept of indexing in other languages, subscript operators can be used to directly access the values in a vector through their index.

### Iterating on the vector

To access the vector we can also iterate through a vector-like we do in other programming languages. We can use for loop to iterate through a vector.

## Updating a vector

we can update the vector using the `push()` method.