03. vector #1

vector1.rs

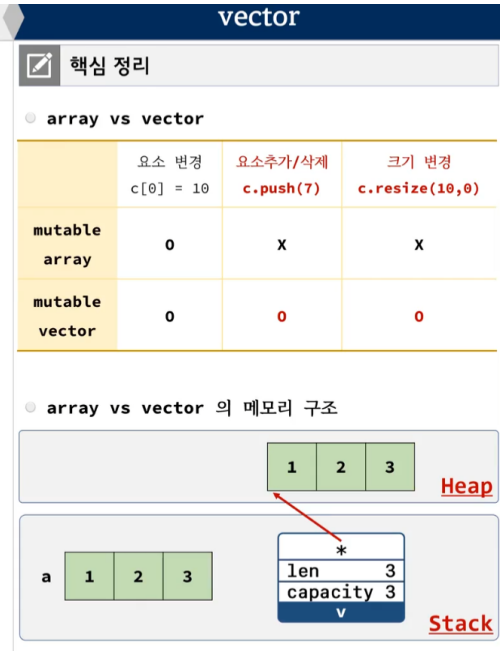
```rust
fn main()
{
    let mut a = [1,2,3];
    let mut v = vec![1,2,3];

    // ❶ 요소 변경 - 배열, vector 모두 가능
    a[0] = 0;  // ok
    v[0] = 0;  // ok

    // ❷ 항목의 추가
    a.push(20); // error
    v.push(20); // ok

    // ❸ 크기 변경
    a.resize(10, 0); // error
    v.resize(10, 0); // ok

    println!("{:?}", v);
}
```



vector1.rs

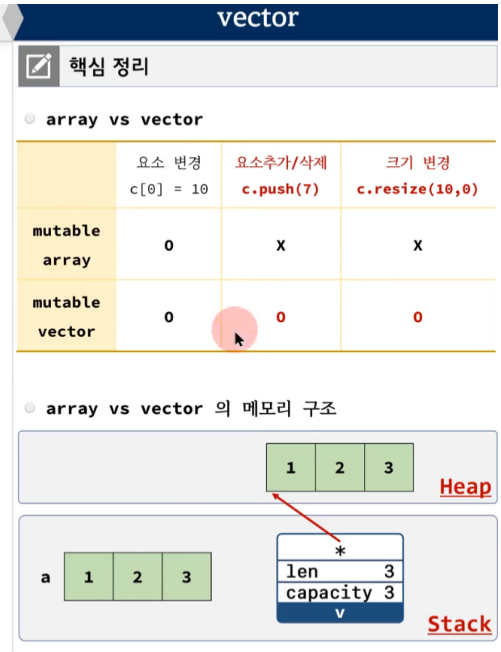
```rust
    // ❷ 항목의 추가
//  a.push(20); // error
    v.push(20); // ok

    // ❸ 크기 변경
//  a.resize(10, 0); // error
    v.resize(10, 0); // ok

    println!("{:?}", v);

    println!("{:?}", std::mem::size_of_val(&a));
    println!("{:?}", std::mem::size_of_val(&v));

    println!("{:p}      ", &a);
    println!("{:p}, {:p}", &v, v.as_ptr());
}
```



std::mem::size_of_val(&a) : 스택에 할당된 a의 배열 크기

std::mem::size_of_val(&v) : 스택에 할당된 v의 벡터 정보(구조체?) 크기

println!("{:p}", &a) : 스택에 할당된 a의 배열 주소

println!("{:p}", &v) : 스택에 할당된 v의 벡터 정보(구조체?) 주소

println!("{:p}", v.as_ptr()) : 스택에 할당된 v의 벡터 정보(구조체?)가 가리키는 힙에 할당된 배열 주소. (위 이미지에서 (*))