## 05. string

19

20

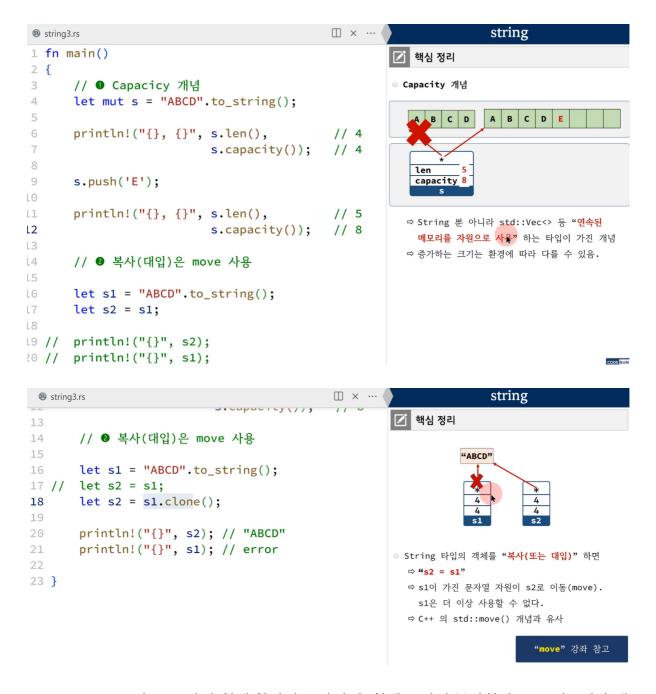
s5.push\_str("OPQ');

```
® string1.rs
                                             □ • … (
                                                                    String
 1 fn main()
                                                      ☑ 핵심 정리
 2 {
       let s1 = "ABCD";
 3
                                     // &str 타입
                                                      Rust 문자열 타입의 종류
       let s2 = String::from("ABCD"); // String 타입
 4
                                                              문자열 자원을 소유.
 5
                                                       String
                                                             C++ std::string 과 유사
 6
 7
       println!("{}, {}", s2.len(), s2.capacity());
                                                              문자열 자원을 소유하지 않음.
                                                        &str
 8
                       // 4, 4
                                                              C++ std::string_view 와 유사
                                                                  "Reference", "&str" 강좌 참고
       println!("{}", std::mem::size_of_val(&s2));
                       // 24
12
                                                      String 객체의 메모리 구조
13
       println!("{:p}", &s2);
14
       println!("{:p}", s2.as_ptr());
                                                               "ABCD"
15
                                                                                     Heap
16 }
17
                                                                        s2 가
                                                         capacity 4
                                                                    "문자열 자원을 소유"
                                                                                    Stack
® string2.rs
                                                                     string
 2 {
 3
       // 1 string 객체를 생성하는 방법
 4
       let s1 = String::new();
       let s2 = String::from("ABCD");
       let s3 = "ABCD".to_string();
       // 2 method
       println!("{}, {}", s1.len(),
                                       s2.len());
                                                              // 0, 4
       println!("{{}}, {{}}", s1.is_empty(), s2.is_empty()); // true, false
10
       println!("{{}}, {{}}", s2.starts_with("AB"), s2.ends_with("AB")); // true, false
11
13
       // ③ 문자(열) 추가, 변경
                 = "ABCD".to_string();
14
       let s4
15
       let mut s5 = "ABCD".to_string();
16
17 // s4.push('X'); // error
       s5.push('X');
18
```

05. string 1

: ® string2.rs // 3 문자(열) 추가, 변경 L3 let s4 = "ABCD".to\_string(); L4 let mut s5 = "ABCD".to\_string(); L5 16 17 // s4.push('X'); // error s5.push('X'); 18 s5.push\_str("OPQ"); 19 20 println!("{}", s5); // ABCDXOPQ 21 22 let s6 = s5.replace("BCD", "----"); 23 println!("{}", s6); // A----XOPQ 24 25 // 4 검색 26 let ret = s5.find("CD"); // Option<T> 27 28 println!("{:?}", ret); // Some(2) 29 30 }

05. string 2



s1.clone() : s1 의 소유 자원(힙에 할당된 문자열)을 힙에 똑같이 복사한다. 소유권 2개가 생긴다고 생각하면 될덧

05. string