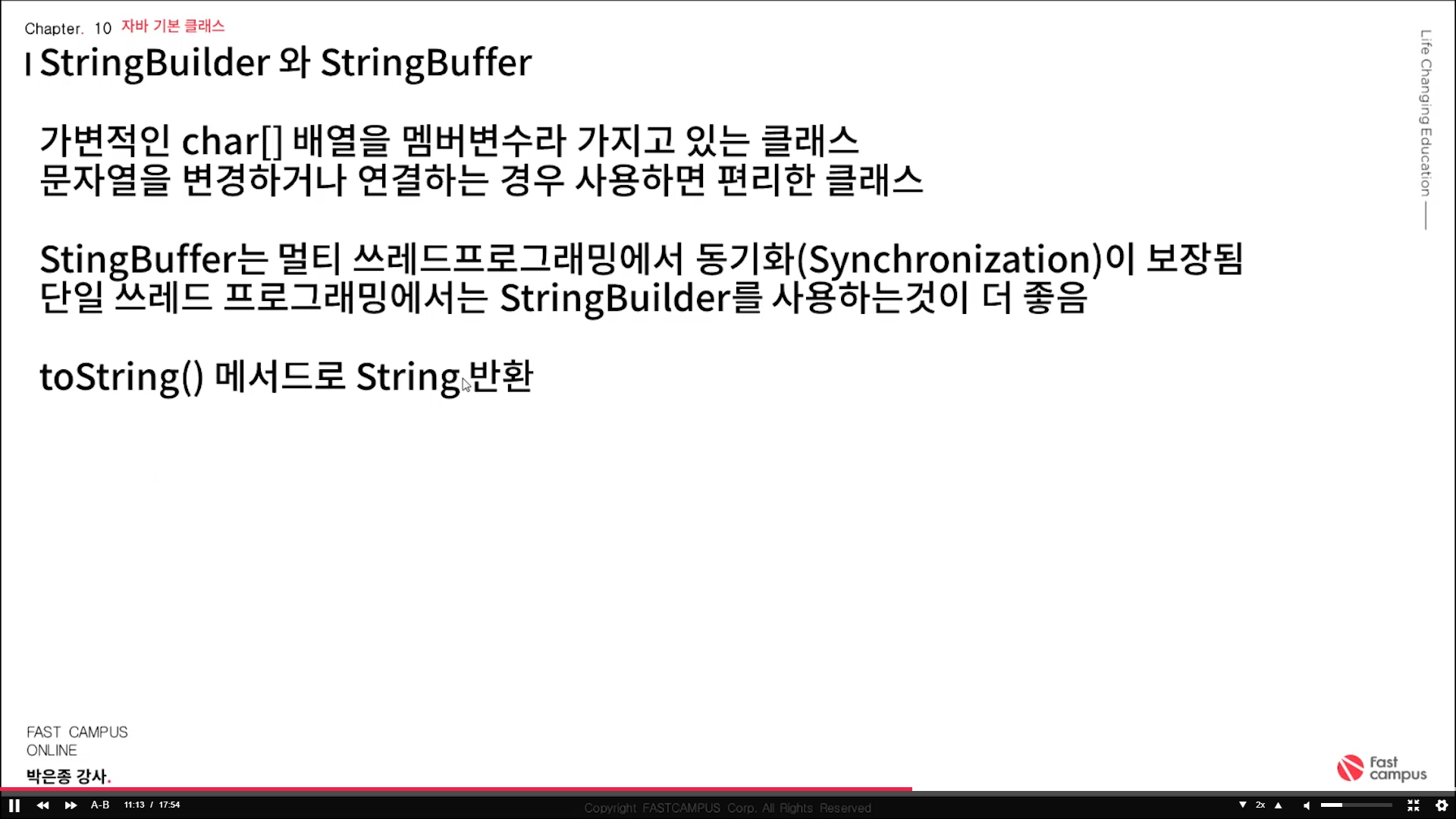
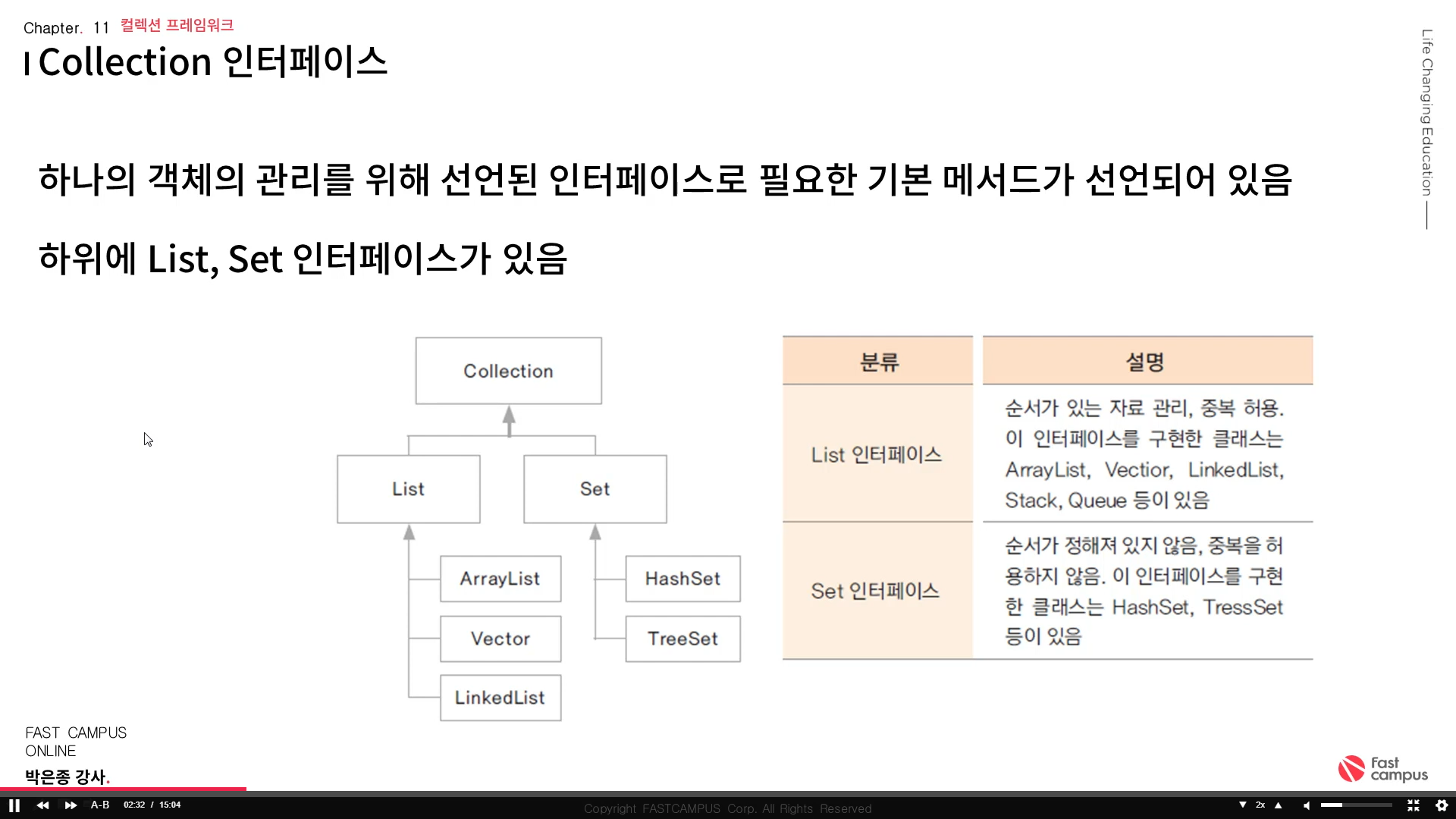


Concat()이나 +를 이용하여 문자열을 연결할 경우, 새로운 메모리에 새로운 문자열이 저장되어 보여지는 것이므로, 메모리의 낭비가 심하다.



문자열을 계속 연결해서 사용해야 할 경우, StringBuilder랑 StringBuffer를 써야한다.





크게 Collection 인터페이스와 Map 인터페이스 두 가지가 있다. Collection 하위에 List인터페이스와 Set 인터페이스가 있고, List인터페이스에 속한 것들은 add하고 api에 제시된대로 썼고, hashSet쓸 때 중복인지 아닌지 알기위해 hashCode와 equals 오버라이딩했다. 이 오버라이딩 함수는 되도록 외워두기. TreeSet에서는 정렬을위해 comparable, comparator를 구현했다.

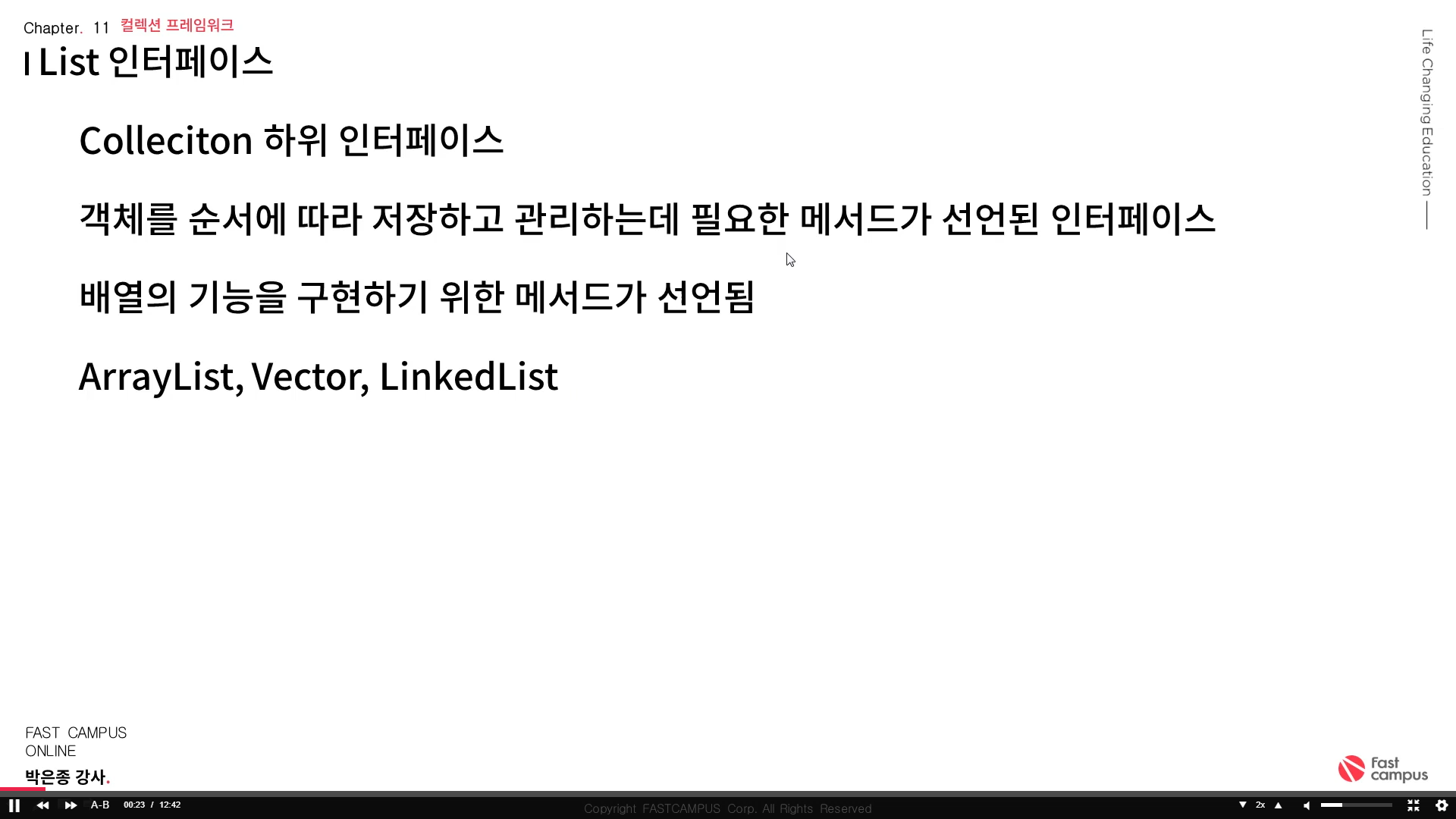
Hash는 검색을 위한 자료구조

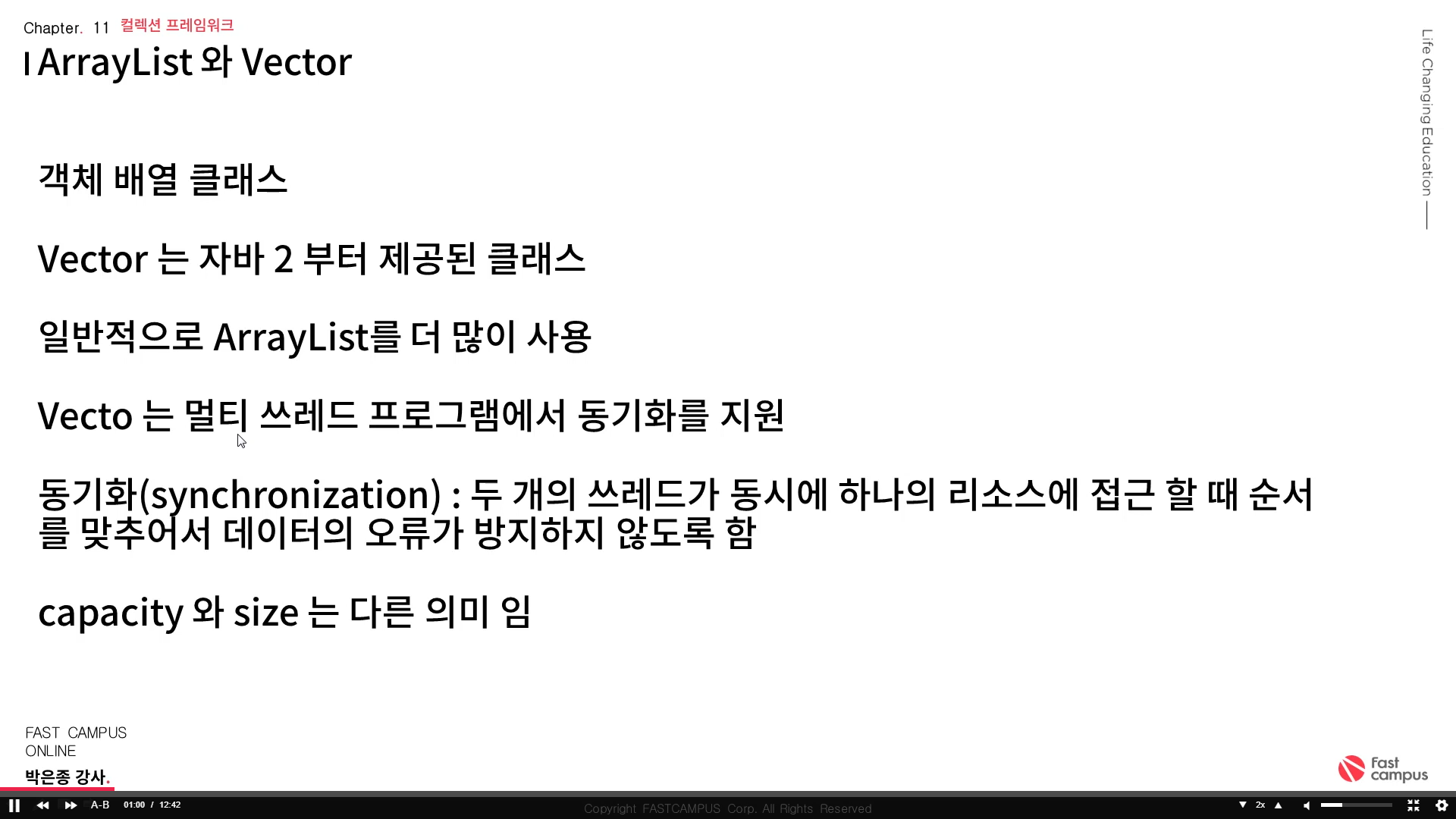
Index의 위치를 key를 주면 알려준다. Index=hash(key)

Binary Tree는 parent 하위에 binary tree가 2개이하인 경우를 말한다.

Binary Search Tree는 검색을 위한 Binary Tree인데, 기존 값과 비교를 해서 새로 들어온 값이 작으면 왼쪽으로 저장되고, 크면 오른쪽으로 저장된다.

List interface는 하나의 자료를 위한 interface이다.





Vector는 모든 메서드에 synchronize단어가 들어있다.

Capacity는 10개짜리 배열이 만들어진다. 즉 배열의 용량

Size는 몇 개의 자료가 들어있나를 리턴해줌