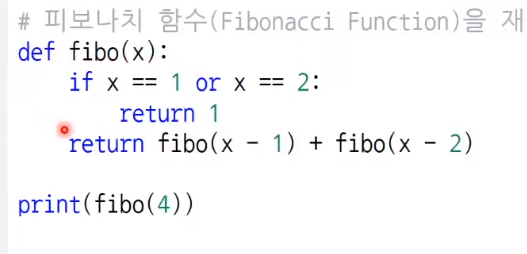
5.다이내믹 프로그래밍

- 한번 계산 한 결과를 다시 이용하는 것

- 탑다운과 바텀업으로 나뉨

- 최적 부분구조, 중복되는 부분 문제

피보나치

* 재귀함수로 구현할 수 있다.
* 
* 같은 것이 여러 번 호출되어 비효율적임

메모리 공간에 한번 구한 값을 저장해놓고 사용하는 것이 효율적

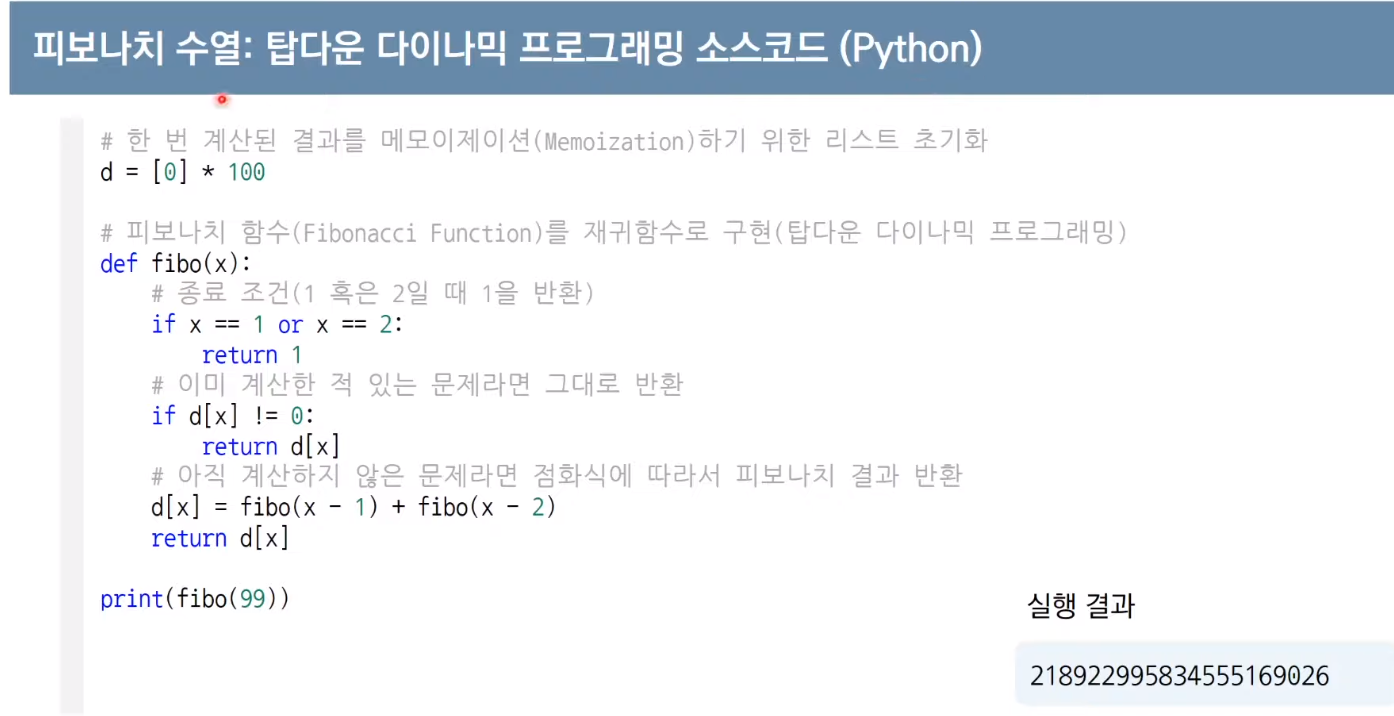
다이내믹 사용조건

1. 큰문제를 작은 문제로 나눌 수 있고
2. 동일한 작은 문제를 반복적으로 호출하는 경우 사용

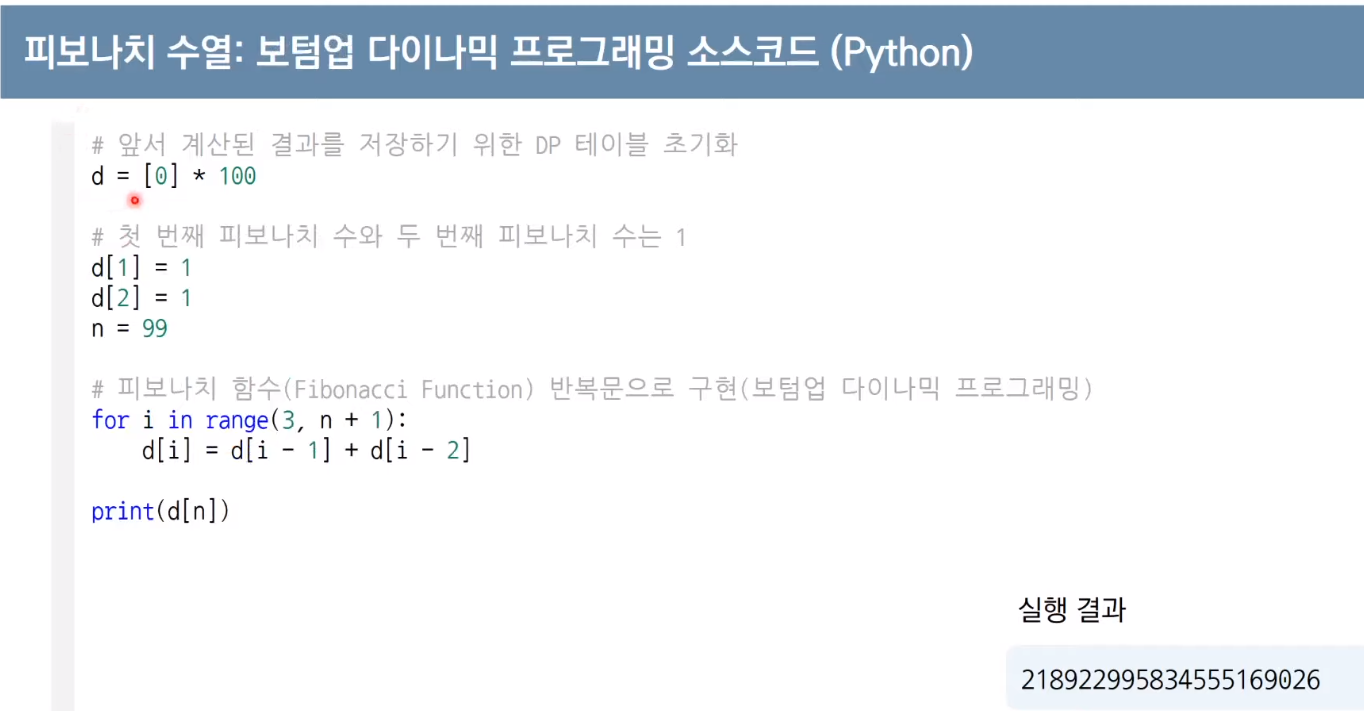
메모이제이션(탑다운) – 결과를 저장하는 테이블 사용

* 한번 계산한 결과를 메모리에 저장하는 기법
* 캐싱이라고도 표현함

바텀업은 반복문 사용



탑다운은 테이블 이용하여 재귀식 사용

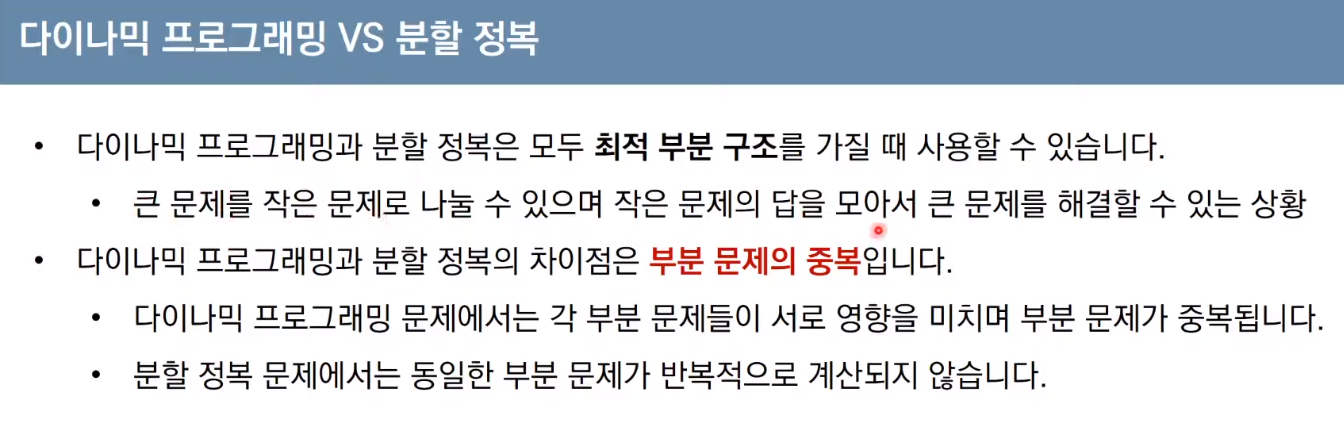


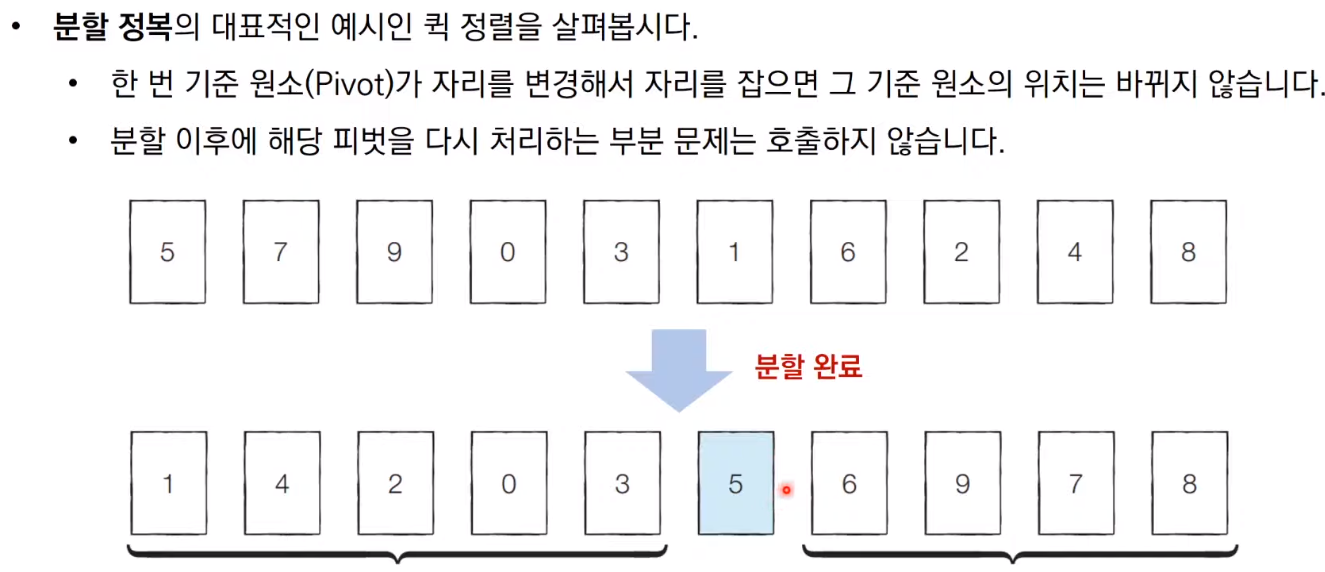
바텀업은 테이블 이용하여 반복문 사용

다이내믹 VS 분할 정복

둘다 큰문제를 작은 문제로 나눌수 있을 때 사용

차이는 부분 문제의 중복이다.





분할 정복은 한번 계산된 결과가 다시 이용되지 않음

\*주어진 문제가 어떤 유형인지 파악하는 것이 중요

\*다른 풀이(그리디,구현,완전탐색)으로 풀리지 않을거 같으면 다이내믹 사용

\* 재귀적으로 어설프게 구현한 다음 탑다운 방식 사용

