**파이썬으로 linked list 만들기**

연결되는 방향에 따라 singly linked list, doubly linked list, circular linked list가 있다.

CS에서 배열과 함께 가장 기본이 되는 대표적인 선형 자료구조 중 하나로 다양한 추상 자료형(ADT)구현의 기반이 된다.

동적으로 새로운 노드를 삽입하거나 삭제하기가 간편하다. 그래서 동적인 메모리 할당으로 메모리 관리에 용이하다.

연결구조를 통해 물리 메모리를 연속적으로 사용하지 않아도 되기 때문에 관리가 쉽다. 또한 데이터를 구조체로 묶어서 포인터로 연결한다는 개념은 여러가지 방법으로 다양하게 활용이 가능하다.

데이터를 노드의 형태로 저장

노드에는 데이터와 다음 노드를 가리키는 포인터를 담은 구조로 이루어 져 있다.

이미지 출처: <https://www.geeksforgeeks.org/data-structures/linked-list/>

연결리스트는 데이터 요소의 선형집합으로, 데이터의 순서가 메모리에 물리적인 순서대로 저장되지는 않는다. 실제 컴퓨터의 물리메모리에는 구조체 각각이 서로 연결된 형태로 구성되어 있으며, 메모리 어딘가 여기저기 흩뿌려진 형상을 띤다.

연결리스트는 배열과 다르게 특정 인덱스에 접근하기 위해서는 전체를 순서대로 읽어야 하므로 상수시간에 접근할 수 없다. 즉 탐색에는 최대 O(n)이 소요된다. (탐색할 때, 첫 노드부터 탐색을 시작해야 하므로 배열보다는 상대적으로 느림)( 사실상 삽입과 삭제가 왼쪽에서(Head에서) 이루어지지 않을 경우, 결국 탐색을 먼저 해야 하기 때문에 삽입과 삭제 모두 적게는 O(k)부터 최악의 경우 O(N)까지 걸릴 가능성이 있다.) 반면 시작 또는 끝지점에 아이템을 추가하거나 삭제, 추출하는 작업은 O(1)에 가능하다.

파이썬에서는 리스트로 연결리스트를 구현한다.

장점: Linked List의 길이를 동적으로 저절 가능/ 데이터의 삽입과 삭제가 쉬움.

단점: 임의의 노드에 바로 접근 불가./ 다음 노드의 위치를 저장하기 위한 추가공간이 필요함/ 연결리스트를 거꾸로 탐색하기 어려움

파이썬으로 singly linked list 만들기

소스코드 참조