Ex32\_더\_해보기

# 문제 : range를 사용하는 방법이 먼가요?

Range는 for루프를 내가 원하는 개수 만큼 도리기 위해, 리스트를 만들기에 좋아요.

Range(start = 0(defaults), end = ? ( <?), 등차 )

이죠.

제가 만약, 어떤 리스트 인덱스 처음 부터 8개 정도 하고 싶다고 하면

Range(8)이라고 하면

0인덱스 ~ 7인덱스 까지 돌면서 들어가는 거죠. 현재 인덱부터 8개를 딱 만들어줘요. 홀수 번째 꺼만 하는 거도 쉽죠. 등차를 2로 두면 되요 .. 짝수는

Range(1,10,2) 식으로 만들면 2,4,6,8,10번 째의 거가 블록으로 들어가서 함수마냥 어떤 오퍼레이션에 참여해서 어떤것을 리턴하게 되죠.

참 유용하네요 !!

# 문제 : for 문 없이 원소들에 range(0,6)을 넣을 수 있냐?

리스트 = [ ]

Range(0,6) = [0, 1, 2, 3, 4, 5] 리스트 이다.

어떤 리스트에, 어떤 리스트를 그대로 뒤로 원소대로 넣으려면, .extend를 사용한다. 리스트.Extend(range(0.6))

리스트에게 먹히는 일종의 함수다.

# 문제 : list 자료형에게는 어떤 메소드가 있나요.

I will try to read pydoc directly and try my best to understand.

List => 클래스

**append**(...)

L.[append](http://localhost:53551/get?key=list#list-append)(object) -> None -- append object to end

**clear**(...)

L.[clear](http://localhost:53551/get?key=list#list-clear)() -> None -- remove all items from L

**copy**(...)

L.[copy](http://localhost:53551/get?key=list#list-copy)() -> list -- a shallow copy of L

**count**(...)

L.[count](http://localhost:53551/get?key=list#list-count)(value) -> integer -- return number of occurrences of value

**extend**(...)

L.[extend](http://localhost:53551/get?key=list#list-extend)(iterable = sequence) -> None -- extend list by appending elements from the iterable

**index**(...)

L.[index](http://localhost:53551/get?key=list#list-index)(value, [start, [stop]]) -> integer -- return first index of value.  
Raises ValueError if the value is not present.

**insert**(...)

L.[insert](http://localhost:53551/get?key=list#list-insert)(index, object) -- insert object before index

**pop**(...)

L.[pop](http://localhost:53551/get?key=list#list-pop)([index]) -> item -- remove and return item at index (default last).  
Raises IndexError if list is empty or index is out of range.

**remove**(...)

L.[remove](http://localhost:53551/get?key=list#list-remove)(value) -> None -- remove first occurrence of value.  
Raises ValueError if the value is not present.

**reverse**(...)

L.[reverse](http://localhost:53551/get?key=list#list-reverse)() -- reverse \*IN PLACE\*

**sort**(...)

L.[sort](http://localhost:53551/get?key=list#list-sort)(key=None, reverse=False) -> None -- stable sort \*IN PLACE\*

A screenshot of a social media post

Description automatically generated

Reverse = False 디폴트 => 오름차순이 디폴트

Reverse = True => 내림차순

m.sort(key=len \* what is standard for that sort?! => if you put func name here, we sort according to the resulted value produced by the element in list going thru that function to let out the value that could be used for the sorting to attempt )

A screenshot of a cell phone

Description automatically generated

리스트. 메소드를 알면 유용한 점이 많음

.append

.count

.index(object, startingpoint(<=조건), endpoint(>조건) )

.insert(index, object) => 굴러들어온 돌이 빼낸돌 양쪽으로 밀어내고 해당 인덱스 차지

List.remove(value) => 발류에 해당하는 요소들중 처음 오는 것들 부터 없앰.

# English Words to learn ( => dev dictionary )