

[Algorithm] 11 강 : 자주 사용하는 라이브러리(유용한 라이브러리) — 나무늘보의 개발 블로그

노트북: 첫 번째 노트북

만든 날짜: 2020-10-23 오후 10:29

URL: <https://continuous-development.tistory.com/155>

Python

[Algorithm] 11 강 : 자주 사용하는 라이브러리(유용한 라이브러리)

2020. 10. 23. 22:29 수정 삭제 공개

자주 사용하는 라이브러리

- 내장 함수 : 기본 입출력 함수부터 정렬 함수까지 기본적인 함수 제공
- itertools : 파이썬에서 반복되는 형태의 데이터를 처리하기 위한 유용한 기능 제공
- heapq : 힙(Heap) 자료구조 제공(우선순위 큐 기능 구현)
- bisect : 이진 탐색 기능을 제공
- collections : 덱(deque) , 카운터(Counter)등 자료구조 포함
- math : 필수적인 수학적 기능 제공 (팩토리얼, 제곱근, 최대공약수(GCD), 삼각 함수 관련 함수부터 파이 같은 상수)

```
# sum()
result = sum([1,2,3,4,5])
print(result)
=> 15

# min() , max()
min_result = min(7,3,5,2)
max_result = max(7,3,5,2)
print(min_result, max_result)
=> 2 7

# eval()
```

```

result = eval("(3+5)*7")
print(result)
=>56

# sorted()
result = sorted([9,1,8,5,4])
reverse_result = sorted([9,1,8,5,4],reverse = True)
print(result)
=>[1,4,5,8,9]
print(reverse_result)
=>[9,8,5,4,1]

# sorted() with key
array = [('홍길동',35),('이순신',75),('아무개',50)]
result = sorted(array,key=lambda x : x[1],reverse=True)
print(result)
=>[('이순신',75),('아무개',50),('홍길동',35)]

```

순열과 조합

모든경우의 수를 고려해야 할 때 사용

순열 : 서로 다른 n 개에서 서로 다른 R 개를 선택하여 일렬로 나열하는 것

{A, B, C} 중 3개 => ABC, ACB, BAC, BCA, CAB, CBA

조합 : 서로 다른 n 개에서 순서에 상관없이 서로 다른 r 개를 선택하는 것

{A, B, C}에서 순서 고려 않고 두 개 뽑는 경우 => AB, AC, BC

순열의 수: $nPr = n * (n - 1) * (n - 2) * \dots * (n - r + 1)$

조합의 수: $nCr = \frac{n * (n - 1) * (n - 2) * \dots * (n - r + 1)}{r!}$

```

# 순열
# A,B,C 에서 두개를 선택하여 나열하는 경우

from itertools import permutations

data = ['A','B','C']

```

```
result = list(permutations(data,3)) # 모든 순열 구하기
print(result)

=>[(A,B,C), (A,C,B), (B,A,C), (B,C,A), (C,A,B), (C,B,A)]
```

```
# 조합
# A,B,C 에서 순서를 고려 하지 않고 두개를 뽑는 경우

from itertools import combinations

data = ['A','B','C']

result = list(combinations(data,2))
print(result)

=> [(A,B), (A,C), (B,C)]
```

Counter

횟수를 세는 기능 제공

리스트와 같은 반복 가능한 객체가 주어졌을 때 내부의 원소가 몇 번씩 등장했는지 세준다.

```
from collections import Counter

counter = Counter(['red','blue','green','blue','blue'])

print(counter['blue']) # blue가 등장한 횟수 출력
=>3

print(counter['green']) # green 이 등장한 횟수 출력
=>1

print(dict(counter)) # 사전 자료형으로 반환
=>{'red':2,'blue':3,'green':1}
```

최대 공약수와 최소 공배수

```
import math

#최소 공배수(LCM)를 구하는 함수
def lcm(a,b):
    return a*b // math.gcd(a,b)

a = 21
b = 14

print(math.gcd(21,14)) # 최대 공약수(GCD) 계산
=>7

print(lcm(21,14)) # 최소 공배수(LCM) 계산
=>42
```

'Python' 카테고리의 다른 글

[Algorithm] 11 강 : 자주 사용하는 라이브러리(유용한 라이브러리)

[Python] 시각화 사용법 - seaborn을 통한 그래프 만들기

[Python] 시각화 사용법 - folium 을 통한 지도 시각화 및 마커(marker) 찍기

[Python] 시각화 사용법 - matplotlib을 통한 box plot 그리기

[Python] matplotlib을 통한 bar plot 그리기

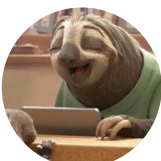
[Python] matplotlib 한글 폰트 깨짐 현상 , 마이너스 기호(폰트)가 깨지는 현상 ...

python counter

python 순열과 조합

python 최대공약수

python 최소 공배수



나무늘보스

혼자 끄적끄적하는 블로그 입니다.

