

# 파이썬 CT

## ▼ 유튜브

### ▼ 처음 기초 100제 코드업

- 다음 백준, 코드포스 온라인 저지
- 백준 - 그리디부터 풀어라, 가장 쉬운 난이도이므로 - 많이 출제됨

그다음은 탐색(삼전에서 많이나옴)

그리디와 탐색만 탄탄히 해도 떨어질일 없다 - 코테 2문제 중 1문제만 제대로 풀어도 합 가능성 높음

고급 다이나믹 필요 없음

+ 기본 동적 프로그램 - 그래프 이론 - 중고급 동적 프로그래밍 - 문자열

어느 대기업이든 기초만 충실히해도

코드포스 블루면 , 삼성 역량 테스트 b형 이면 코테 합

연구직, 대학원이면 그 이상이지

- 삼성전자 역량 테스트 기출은 백준에서 확인 가능
- 카카오 기출도 프래그래머스에 있음
- 나동빈 알고리즘 유튜브만 좀 보라 - 그리디, 동적 기초, 탐색 50개씩 풀어보고 삼성 전자 기출 이후 여러 문제집 풀어보라
- 2015 기준 코테로 뽑는 곳이 점점 늘어남
- 대회나 성능 고려한다면 c++, 웬만하면 파이썬

<https://hkim-data.tistory.com/30?category=1016184> 기본 파이썬 정리

## ▼ 코드업100

- `print(" " " ")`
- `print(' " " ')`
  - 프린트 내 "" 쓰고 싶으면 전체를 "로 감싸도됨

- 리스트와 어레이

```
# 리스트 연산

a = [1, 3, 5]
b = [2, 4, 6]

a + b # [1, 3, 5, 2, 4, 6]

# 허용하지 않음 : a - b, a * b, a / b

a * 3 # [1, 3, 5, 1, 3, 5, 1, 3, 5]
```

```
# 넘파이 어레이 연산

a = np.array([1, 3, 5])
b = np.array([2, 4, 6])

a + b # array([ 3,  7, 11])
a * b # array([ 2, 12, 30])

a + 2 # array([3, 5, 7])
a * 3 # array([ 3,  9, 15])
```

```
a = [1, 3, 5]
b = np.array([1, 3, 5])

# 리스트 메소드(b.append(7) 등은 불가능)
a.append(7)
a.extend([9])
a.remove(9)

# 어레이 메소드(a.mean() 등은 불가능)
b.mean()
b.argmax()
b.round()
```

- 기준점 기준 따로 저장, 공백없이 출력하기

```
y,m,d = input().split('.')
print(d,"-",m,"-",y, sep="")
```

- 당연히 인풋으로 받은 건 모두 string으로 인식

```
s = input()
print(s[0])
print(s[2])
```

- 스트링 이어붙일때

```
print(b,a) 중간에 공백
print(b+a) 공백 없음
```

- 10진수 → 16진수

```
a = input()
a = int(a)
print('%X' %a) #X면 대문자, x면 소문자
```

hex(a)

format(100, '#x')

두가지 모두 가능, 반환값은 string

n = int(a, 16) #입력된 a를 16진수로 인식해 변수 n에 저장

- ○유니코드값 변환

```
a = input()
b = ord(a)
print(b)
```

ord( ) 는 어떤 문자의 순서 위치(ordinal position) 값을 의미한다.

실제로 각각의 문자들에는 연속된 정수 값이 순서에 따라 부여 되어 있다. A:65, B:66, C:67 ....

ord(c) : 문자 c 를 10진수로 변환한 값

chr( )는 정수값->문자, ord( )는 문자->정수값 형태로 바꿔주는 서로 반대 방향으로 바뀌어 주는 기능을 한다.

- 스트링 반복

```
a = input()
b = int(input())
for i in range(b):
    print(a)
-----동일
a, b = input().split()
print(a*int(b))
```

문자열 \* 정수 또는 정수 \* 문자열은 그 문자열을 여러 번 반복한 문자열

- 거듭제곱

c = a\*\*b

- 소수점 자리 조절

```
a = float(input())
print(format(a, ".2f"))
-----
print("소수 두번째 자리까지 표기: ", round(num, 2))
print("소수 두번째 자리까지 표기: {:.2f}".format(num1))
print("소수 두번째 자리까지 표기: ", format(num, ".2f"))
```

- format의 다른 이용

a=1, b=2

"{0}이면 {1}이고 {0}인데".format(a,b)

•