

# ADV. 대비 파이썬 과정\_입문

김 태 현

# Advanced **검정** 소개

- 3시간 1문제
- 제출 즉시 결과 확인 가능
- 틀린 예제는 표시되지 않음
- 입사 시험 수준 (이하)
- Python 3.7(pypy7.3.4)
- 주요 알고리즘
  - 구현, 시뮬레이션
  - 경우의 수 조합(백트래킹)
  - 2차원 배열 탐색(DFS, BFS)
  - (최근 입사시험에서 priority queue, linked list도 나오는데 설마..?)

# Certi 등급

## Advanced

Python 3.7(pypy7.3.4)

3시간 1문제

일반 문제 형태

귀찮고 까다로운 구현력 요구

시간 복잡도를 크게 고려하지 않는

모든 경우의 수, 배열 탐색

간단한 dataset 관리

제한된 알고리즘

backtracking, dfs, bfs

## Pro

Python 3.7(pypy7.3.4)

4시간 1문제(로 위장한 여러문제)

main<->solution interactive 형태

advanced 수준 이상(혹은 이하)의 구현력 요구

시간복잡도 공간복잡도 최적화 매우 중요

복잡한 dataset 효율적인 관리

다양한 알고리즘

hash, priority queue, union-find

dijkstra, segment tree, dfs, bfs,

binary search, calculator, ..

## Expert

여긴 어디  
나는 누구

# 문제풀이 전략

## 1. 파악, 이해

- 중요한 로직 정보 정리

## 2. 설계

- 로직별 분류, 도식화, 수행방법 정의
- 필요한 dataset 정리

## 3. 검증

- 설계시 정한 내용으로 전체적인 로직을 수행하는데 문제 없는지 검증

## 4. 구현

- 설계 내용 기반으로 간결하고 효율적인 구현

## 5. 완성

- 로깅, 디버깅, 코드 정독
- 문제 제약사항, 예외케이스 꼼꼼히 확인
- 입력, 배열 경계 부분 확인

# 학습 전략

- **이론 학습**

- 파이썬 문법
- 필요 알고리즘

- **설계**

- 이론을 적용
- 사실상, 한정적이어서 크게 어렵지 않다.

- **구현**

- 모든 문제를 직접 구현한다.
- 코드를 깔끔하게 정리한다.
- 다른 사람 코드를 보며 좋은 스타일을 흡수하고 개선해나간다.
- ad문제는 그냥 많이 풀어보는게 답이다. (물론 잘 정리하면서)

# 교육 과정

- ~~문법~~ 필수 문법 과정으로 이동
- ~~Data type~~ 문법 연습문제, 필수 문법 과정으로 이동
- **도형 만들기** 구현의 시작, 규칙 찾기
- **도형 회전** 검정에서 즐겨내는 컨셉
- **구현, 시뮬레이션** AD 그 자체
- **재귀** 백트래킹과 DFS의 도구
- **백트래킹** 모든 경우의 수를 가능성 있는 곳만 탐색하자
- **2차원 배열 탐색-DFS, BFS** pro, expert에서도 중요

# 교육 특징

## pycharm 사용

- Python 3.7(pypy7.3.4)
- 검정 PC에 맞춰 사용
- vs code로 해도 상관 없음

## 교육 수준

- 선수과정 'Python 필수 문법'
- 주제별로 뒤쪽에 배치된 문제들은 난이도가 있습니다.
- 독파시 합격 가능

## 제공 코드 관련

- 코드에 주석이 많지 않을 예정  
코드 읽는 눈 중요
- 구현 스타일은 표준 가이드랑 다를 수 있음

## 코드 작성 네추럴한 모습 추구

틀리고 찾는 과정을 보여드리려고  
최대한 노력했습니다.

## 교육 활용

모르는 개념 => 영상, 자료

문제 먼저 풀어보기

- 쉽다 => 코드만 참고
- 접근이 어렵다 => 해법 설명 듣고 구현해보기
- 구현이 어렵다 => 같이 구현해보고 다시 구현해보기

**감사합니다.**