

# lotto

---

## 목표

---

- Python 기본 문법 습득
- 파일 입출력에 대한 이해
- 객체지향 프로그래밍에 대한 이해
- JSON 형태의 데이터를 가공하고 사용하기

## 준비사항

---

- 개발도구
  - Visual Studio Code
  - Python 3.9+
- 필수 라이브러리
  - json
  - random

## 배경

---

**로또(lotto)**란 1부터 45까지의 번호 중 6개의 번호를 구매자가 직접 선택하고, 당첨 번호와 로또 번호가 일치하는 개수에 따라 순위를 결정하는 복권 게임입니다.

**당첨 번호**는 총 7개의 번호(메인 6개 + 보너스 1개)로 이루어져 있으며, 이 중 로또 번호와 일치하는 개수에 따라 1등부터 5등까지 결정됩니다.

**로또 번호**는 한 줄당 6개의 번호이며, A부터 E까지 최대 다섯 줄로 이루어집니다. 각 줄은 독립적으로 당첨 여부를 비교합니다. 즉, 여러 줄이 동시에 당첨될 수 있습니다.

아래는 실제 로또 용지 예시입니다. `회차`, `추첨일`, `로또 번호` 등의 정보가 나타나 있습니다.



## 프로그램 안내

---

위에서 소개한 로또 규칙과 주어진 코드를 활용하여 아래와 같이 동작하는 로또 프로그램을 만들어봅시다.

1. 생성할 로또 번호의 줄 개수를 입력받습니다. 최대 다섯 줄까지 가능합니다.
2. 회차, 추첨일, 로또 번호 정보를 출력합니다.
3. 해당 회차의 당첨 번호 정보를 출력합니다. (메인 6개 + 보너스 1개)
4. 당첨 번호와 로또 번호의 각 줄을 비교하여 각각의 당첨 결과를 출력합니다.
5. 출력 결과는 다음과 같습니다.

```
생성할 로또 줄 개수를 입력하세요.(1줄에서 5줄까지 생성 가능) : 5
=====
                        제 1024 회 로또
=====
추첨일 : 2022/07/16 (토)
=====
A : [8, 12, 26, 32, 42, 43]
B : [2, 10, 17, 18, 19, 43]
C : [4, 10, 28, 38, 40, 42]
D : [4, 12, 15, 27, 39, 43]
E : [2, 3, 8, 19, 27, 40]
=====
당첨 번호 : [9, 18, 20, 22, 38, 44] + 10
=====
A : 0개 일치 (낙첨)
B : 1개 + 보너스 일치 (낙첨)
C : 1개 + 보너스 일치 (낙첨)
D : 0개 일치 (낙첨)
E : 0개 일치 (낙첨)
```

주어진 코드에서 미리 작성된 부분은 수정하지 않습니다.

## 제공 파일 안내

### data/

- data 폴더는 총 3개의 파일을 가지고 있습니다.
- 3개의 파일은 각각 1023, 1024, 1025 회차의 로또 추첨 정보를 json 형식으로 가지고 있습니다.
- 추첨일과 당첨 번호에 대한 정보는 해당 회차 json 파일에서 추출하여 사용합니다.

### main.py

- 로또 프로그램을 실행시키기 위한 메인 파일입니다.
- 터미널에 `python main.py`를 입력하여 로또 프로그램을 실행할 수 있습니다.
- 변수 설명
  - `draw_number` : 회차 번호
  - `lotto` : lotto.py에 선언된 Lotto 클래스의 인스턴스 (Lotto 클래스는 lotto.py로부터 import 하여 사용)
  - `n` : 생성할 로또 번호의 줄 개수 (입력을 통해 할당)
- lotto 인스턴스의 메서드를 단계적으로 호출하는 방식으로 전체 프로그램을 진행합니다.

### lotto.py

- Lotto 클래스가 선언된 파일입니다.

- Lotto 클래스는 해당 로또 프로그램에 사용되는 모든 기능을 가지고 있습니다.
- Lotto 클래스에 있는 메서드 정보는 아래와 같습니다.

| 메서드  | 설명  |
|--|---|
| <code>__init__(self)</code>                                  | Lotto 클래스의 생성자  |
| <code>generate_lines(self, n)</code>                         | n 개의 줄 수만큼 로또 번호 줄을 생성하는 인스턴스 메서드                             |
| <code>print_number_lines(self, draw_number)</code>           | 회차, 추첨일, 로또 번호 정보를 출력하는 인스턴스 메서드                              |
| <code>print_result(self, draw_number)</code>                 | 해당 회차의 당첨 번호와 당첨 결과를 출력하는 인스턴스 메서드                            |
| <code>get_draw_date(draw_number)</code>                      | 해당 회차 추첨일의 년, 월, 일 정보를 튜플로 반환하는 스택틱 메서드                       |
| <code>get_lotto_numbers(draw_number)</code>                  | 해당 회차 당첨 번호의 메인 번호와 보너스 번호가 담긴 튜플을 반환하는 스택틱 메서드               |
| <code>get_same_info(main_numbers, bonus_number, line)</code> | 한 줄의 로또 번호와 메인 번호가 일치하는 개수와 보너스 번호 일치 여부가 담긴 튜플을 반환하는 스택틱 메서드 |
| <code>get_ranking(same_main_counts, is_bonus)</code>         | 당첨 결과를 정수로 반환하는 스택틱 메서드                                       |

## test.py

- 요구사항을 잘 지켜서 코드를 작성했는지 테스트하는 파일입니다.
- 터미널에 `python test.py` 를 입력하여 테스트를 실행할 수 있습니다.
- 테스트를 통과하면 아래와 같이 출력됩니다.

```
$ python test.py
축하합니다! 모든 테스트를 통과하였습니다.
```

- 테스트를 통과하지 못한다면 어느 부분에서 기댓값과 반환값이 다른지 결과가 출력됩니다. 이를 참고로 하여 테스트를 통과할 수 있도록 코드를 수정합니다. (기댓값 : 원래 반환해야 하는 값, 반환값 : 현재 실제로 반환되는 값)

```
$ python test.py
lotto.get_lotto_numbers(1023)
기댓값 : ([10, 14, 16, 18, 29, 35], 25)
반환값 : ([10, 14, 16, 18, 29, 35], 27)

lotto.get_lotto_numbers(1024)
기댓값 : ([9, 18, 20, 22, 38, 44], 10)
반환값 : ([9, 18, 20, 22, 38, 44], 12)

lotto.get_lotto_numbers(1025)
기댓값 : ([8, 9, 20, 25, 29, 33], 7)
반환값 : ([8, 9, 20, 25, 29, 33], 9)

3개의 통과하지 못한 테스트가 있습니다.
```

# 요구사항

순서에 따라 작성하면 최종 프로그램이 완성되는 형태이므로, 반드시 요구사항 순서에 따라 코드를 작성합니다.

이후, test.py를 실행시켜 작성한 메서드들이 정상적으로 동작하는지 반드시 확인 후 제출합니다. 최종 출력 결과는 README의 맨 아래 있는 출력 예시를 참고합니다.

## 1. 생성할 로또 번호의 줄 개수 입력받기

1-1) main.py에서 작성합니다.

1-2) 생성할 로또 번호의 줄 개수 n이 1 미만이거나 5 초과인 경우, 1줄에서 5줄까지만 생성 가능합니다. 다시 입력하세요. 라는 문구를 출력하고 다시 n을 입력받습니다.

```
생성할 로또 줄 개수를 입력하세요.(1줄에서 5줄까지 생성 가능) : 0
1줄에서 5줄까지만 생성 가능합니다. 다시 입력하세요.

생성할 로또 줄 개수를 입력하세요.(1줄에서 5줄까지 생성 가능) : 6
1줄에서 5줄까지만 생성 가능합니다. 다시 입력하세요.

생성할 로또 줄 개수를 입력하세요.(1줄에서 5줄까지 생성 가능) : 
```

1-3) 생성할 로또 번호의 줄 개수 n이 1 이상 5 이하인 경우, 2. 로또 번호 생성하기로 넘어갑니다.

## 2. 로또 번호 생성하기

2-1) lotto.py에서 작성합니다.

2-2) 아래 조건을 만족하도록 생성자 를 작성합니다.

- number\_lines 라는 인스턴스 변수에 빈 리스트가 할당됩니다.
- 이후 number\_lines 는 로또 번호에 해당하는 모든 줄을 담는 용도로 사용됩니다.

2-3) 아래 조건을 만족하도록 generate\_lines(self, n) 를 작성합니다.

- 매개변수는 아래와 같습니다.

| 매개변수 | 타입  | 설명              |
|------|-----|-----------------|
| n    | int | 생성할 로또 번호의 줄 개수 |

- 반환값은 없습니다.
- n 줄의 로또 번호를 생성하는 인스턴스 메서드입니다.
- 한 줄의 로또 번호는 random 모듈을 이용하여 1부터 45까지의 번호 중 랜덤하게 6개를 추출하여 생성합니다.
- 한 줄의 로또 번호는 오름차순으로 정렬되어야 합니다.

- 생성한 n 줄의 로또 번호를 `number_lines`에 저장합니다.
- 저장된 `number_lines`은 아래와 같이 이차원 리스트 형태로 로또 번호 줄을 가집니다.

```
# n = 5 일 때의 number_lines

[
    [6, 11, 26, 28, 33, 41],
    [4, 10, 28, 29, 34, 36],
    [1, 22, 30, 31, 36, 37],
    [7, 10, 11, 12, 32, 45],
    [2, 6, 8, 23, 38, 39]
]
```

### 3. 회차, 추첨일, 로또 번호 출력하기

3-1) `lotto.py`에서 작성합니다.

3-2) 아래 조건을 만족하도록 `print_number_lines(self, draw_number)`를 작성합니다.

- 매개변수는 아래와 같습니다.

| 매개변수                     | 타입               | 설명    |
|--------------------------|------------------|-------|
| <code>draw_number</code> | <code>int</code> | 회차 번호 |

- 반환값은 없습니다.
- 회차, 추첨일, 로또 번호 정보를 출력하는 인스턴스 메서드입니다.
- 3-3을 참고하여 `get_draw_date(draw_number)`를 작성하고, 그 반환값을 `year`, `month`, `day` 세 변수에 할당합니다.
- 아래 화면처럼 회차와 추첨일을 출력합니다.

```
=====
              제 1024 회 로또
=====
추첨일 : 2022/07/16 (토)
=====
```

- 아래 화면처럼 `number_lines`에 저장된 로또 번호의 각 줄을 출력합니다. A : [1, 2, 3, 4, 5, 6]의 형식으로 출력합니다.

```
=====
              제 1024 회 로또
=====
추첨일 : 2022/07/16 (토)
=====
A : [5, 8, 19, 25, 26, 37]
B : [2, 3, 21, 24, 30, 41]
C : [3, 4, 10, 11, 18, 34]
D : [12, 20, 24, 34, 38, 45]
E : [6, 20, 27, 35, 43, 44]
```

3-3) 아래 조건을 만족하도록 `get_draw_date(draw_number)`를 작성합니다.

- 매개변수는 아래와 같습니다.

| 매개변수        | 타입  | 설명    |
|-------------|-----|-------|
| draw_number | int | 회차 번호 |

- 반환값은 아래와 같습니다.

| 반환값   | 타입  | 설명      |
|-------|-----|---------|
| year  | str | 추첨일의 년도 |
| month | str | 추첨일의 월  |
| day   | str | 추첨일의 일  |

- 해당 회차 추첨일의 년, 월, 일 정보를 튜플로 반환하는 스택 메서드입니다.
- 내장 함수 `open` 과 `json` 모듈을 이용하여 해당 회차의 추첨일을 추출합니다. (추첨일에 해당하는 키는 `drwNoDate` 입니다.)
- 추첨일에서 `-` 를 기준으로 년, 월, 일을 분리합니다. (년, 월, 일은 문자열입니다.)

## 4. 당첨 번호와 당첨 결과 출력하기

4-1) `lotto.py` 에서 작성합니다.

4-2) 아래 조건을 만족하도록 `print_result(self, draw_number)` 를 작성합니다.

- 매개변수는 아래와 같습니다.

| 매개변수        | 타입  | 설명    |
|-------------|-----|-------|
| draw_number | int | 회차 번호 |

- 반환값은 없습니다.
- 해당 회차의 당첨 번호와 당첨 결과를 출력하는 인스턴스 메서드입니다.
- 4-3을 참고하여 `get_lotto_numbers(draw_number)` 를 작성하고, 그 반환값을 `main_numbers`, `bonus_number` 두 변수에 할당합니다.
- 아래 화면처럼 당첨 번호를 출력합니다. 당첨 번호는 메인 번호 6개와 보너스 번호 1개로 구성됩니다.

```
=====
당첨 번호 : [9, 18, 20, 22, 38, 44] + 10
=====
```

- 아래 화면처럼 `number_lines` 에 들어있는 각 로또 번호 줄에 대해 당첨 결과를 출력합니다.
  - 4-4를 참고하여 `get_same_info(main_numbers, bonus_number, line)` 를 작성하고, 그 반환값을 `same_main_counts`, `is_bonus` 두 변수에 할당합니다.
  - 당첨 번호와 로또 번호가 일치하는 개수에 따라 등위를 결정합니다. 등위에 들지 못하면 `낙첨` 입니다. (아래 참고)  
 등위는 4-5를 참고하여 `get_ranking(same_main_counts, is_bonus)` 를 작성하고, 그 반환값을 `ranking` 변수에 할당합니다.

- 아래 화면처럼 각 로또 번호 줄의 메인 번호 일치 개수, 보너스 번호 일치 여부, 등위를 형식에 맞게 출력합니다.

■ 예시 1

```
=====
당첨 번호 : [9, 18, 20, 22, 38, 44] + 10
=====
A : 0개 일치 (낙첨)
B : 0개 일치 (낙첨)
C : 1개 + 보너스 일치 (낙첨)
D : 2개 일치 (낙첨)
E : 2개 일치 (낙첨)
```

■ 예시 2

```
=====
당첨 번호 : [8, 9, 20, 25, 29, 33] + 7
=====
A : 3개 일치 (5등 당첨!)
B : 1개 일치 (낙첨)
C : 1개 일치 (낙첨)
D : 3개 일치 (5등 당첨!)
E : 0개 일치 (낙첨)
```

4-3) 아래 조건을 만족하도록 `get_totto_numbers(draw_number)` 를 작성합니다.

- 매개변수는 아래와 같습니다.

| 매개변수        | 타입  | 설명    |
|-------------|-----|-------|
| draw_number | int | 회차 번호 |

- 반환값은 아래와 같습니다.

| 반환값          | 타입   | 설명        |
|--------------|------|-----------|
| main_numbers | list | 메인 번호 리스트 |
| bonus_number | int  | 보너스 번호    |

- 해당 회차 당첨 번호의 메인 번호와 보너스 번호가 담긴 튜플을 반환하는 스택틱 메서드입니다.
- 내장 함수 `open` 과 `json` 모듈을 이용하여 해당 회차의 메인 번호와 보너스 번호를 추출합니다.  
(메인 번호에 해당하는 키는 6개이며 `drwt`로 시작합니다. 문자열 메서드 `startswith`를 활용합니다.)  
(보너스 번호에 해당하는 키는 `bnusNo` 입니다.)
- 메인 번호 6개는 하나의 리스트에 담기며 오름차순으로 정렬되어야 합니다.

4-4) 아래 조건을 만족하도록 `get_same_info(main_numbers, bonus_number, line)` 를 작성합니다.

- 매개변수는 아래와 같습니다.

| 매개변수         | 타입   | 설명        |
|--------------|------|-----------|
| main_numbers | list | 메인 번호 리스트 |
| bonus_number | int  | 보너스 번호    |

| 매개변수 | 타입   | 설명             |
|------|------|----------------|
| line | list | 한 줄의 로또 번호 리스트 |

- 반환값은 아래와 같습니다.

| 반환값              | 타입   | 설명             |
|------------------|------|----------------|
| same_main_counts | int  | 메인 번호와 일치하는 개수 |
| is_bonus         | bool | 보너스 번호 일치 여부   |

- 한 줄의 로또 번호와 메인 번호가 일치하는 개수와 보너스 번호 일치 여부가 담긴 튜플을 반환하는 스택 메서드입니다.

4-5) 아래 조건을 만족하도록 `get_ranking(same_main_counts, is_bonus)` 를 작성합니다.

- 매개변수는 아래와 같습니다.

| 매개변수             | 타입   | 설명             |
|------------------|------|----------------|
| same_main_counts | int  | 메인 번호와 일치하는 개수 |
| is_bonus         | bool | 보너스 번호 일치 여부   |

- 반환값은 아래와 같습니다.

| 반환값     | 타입  | 설명    |
|---------|-----|-------|
| ranking | int | 당첨 순위 |

- 당첨 결과를 정수로 반환하는 스택 메서드입니다.
- 순위 산정 방식은 아래 표를 참고합니다. 1등에서 5등안에 속하지 않으면 낙첨입니다.

| 순위 | 당첨내용                 |
|----|----------------------|
| 1등 | 6개 번호 일치             |
| 2등 | 5개 번호 일치 + 보너스 번호 일치 |
| 3등 | 5개 번호 일치             |
| 4등 | 4개 번호 일치             |
| 5등 | 3개 번호 일치             |
| 낙첨 | 2개 이하 번호 일치          |

## 출력 예시



### 1. n이 1인 경우

```
생성할 로또 줄 개수를 입력하세요.(1줄에서 5줄까지 생성 가능) : 1
=====
제 1023 회 로또
=====
추첨일 : 2022/07/09 (토)
=====
A : [2, 29, 35, 40, 41, 44]
=====
당첨 번호 : [10, 14, 16, 18, 29, 35] + 25
=====
A : 2개 일치 (낙첨)
```

### 2. n이 3인 경우

```
생성할 로또 줄 개수를 입력하세요.(1줄에서 5줄까지 생성 가능) : 3
=====
제 1024 회 로또
=====
추첨일 : 2022/07/16 (토)
=====
A : [10, 11, 17, 27, 32, 42]
B : [10, 11, 14, 23, 33, 40]
C : [5, 9, 15, 17, 29, 42]
=====
당첨 번호 : [9, 18, 20, 22, 38, 44] + 10
=====
A : 0개 + 보너스 일치 (낙첨)
B : 0개 + 보너스 일치 (낙첨)
C : 1개 일치 (낙첨)
```

### 3. n이 5인 경우

```
생성할 로또 줄 개수를 입력하세요.(1줄에서 5줄까지 생성 가능) : 5
=====
제 1025 회 로또
=====
추첨일 : 2022/07/23 (토)
=====
A : [4, 6, 9, 20, 29, 43]
B : [4, 12, 14, 24, 32, 33]
C : [1, 6, 8, 27, 32, 44]
D : [8, 11, 20, 26, 33, 37]
E : [1, 23, 26, 28, 36, 38]
=====
당첨 번호 : [8, 9, 20, 25, 29, 33] + 7
=====
A : 3개 일치 (5등 당첨!)
B : 1개 일치 (낙첨)
C : 1개 일치 (낙첨)
D : 3개 일치 (5등 당첨!)
E : 0개 일치 (낙첨)
```