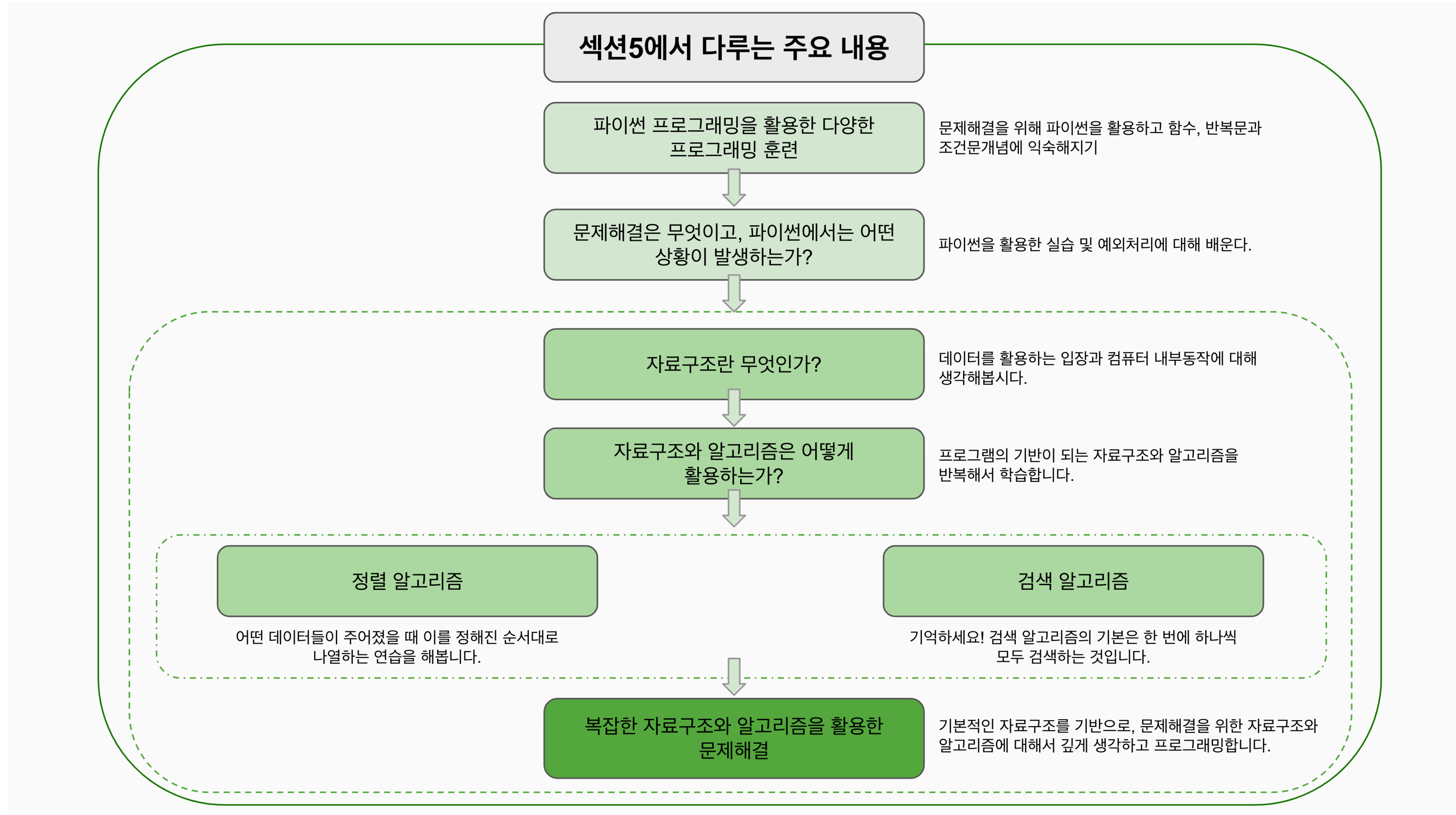


# Section5 Sprint1 Wrap-Up

## Section5 Flow



## Section5 Sprint1을 모두 마치고 나면 여러분은

- ☐ 반복문과 조건문, 함수와 같은 기본적인 것으로 **비정형화된 코드**를 작성할 수 있습니다.
- ☐ **OOP 프로그래밍**에 익숙해짐으로써 **대규모 소스코드를 활용**하실 수 있습니다.
- ☐ **다양한 메소드**에 대해 내부로직을 익힘으로써 상황별 적합한 메소드를 활용하실 수 있습니다.

🔥 중요하지만 도전적인 목표를 하나 더 드리자면

- ☐ **Section5 전체의 최종목적은 자료구조와 알고리즘을 활용하는 것뿐만 아니라, 이해하고 직접 구현해보는 것입니다.**

## Sprint 1

# Python and Problem Solving - Note

N511

### Note 1 목표

- 파이썬 다양한 메소드에 대해 내부로직을 생각하면서 어떤 상황에서 활용되는지 익힙니다.
- 다양한 코드를 활용하여 컬렉션 자료형(리스트, 튜플, 셋, 딕셔너리)에 대해 익힙니다.
- 현실상황을 수학적 기초개념(사칙연산)으로 프로그래밍하는데 익숙해집니다.

자신이 정리했던 목표를 작성해보세요.

### Note 1 키워드

반복문과 조건문  
내장 메소드  
프로그래밍

자신이 많이 활용한 것을 키워드로 정리해보세요.

N512

### Note 2 목표

- 프로그래밍 진행과정을 생각해봅니다.
- 파이썬을 활용한 실습 및 예외처리에 대해 배웁니다.
- 알고리즘을 위한 논리적 방법을 익힙니다.

자신이 정리했던 목표를 작성해보세요.

### Note 2 키워드

문제해결  
의사코드  
컴프리헨션

자신이 많이 활용한 것을 키워드로 정리해보세요.

## Sprint 1

# Python and Problem Solving - Note

N513

### Note 3 목표

- OOP에 대한 기본개념을 숙지합니다.
- 소스코드를 통해 OOP의 기본개념을 파악합니다.
- OOP와 다른 프로그래밍 패러다임의 특징에 대해 알아봅니다.

자신이 정리했던 목표를 작성해보세요.

### Note 3 키워드

OOP 기본컨셉  
클래스 설계

자신이 많이 활용한 것을 키워드로 정리해보세요.

N514

### Note 4 목표

- 데이터를 활용하는 입장과 컴퓨터 내부동작에 대해 생각해봅니다.
- 알고리즘 효율성에 대한 내용인 자료구조의 Big O 및 복잡도 개념에 대해 배웁니다.

자신이 정리했던 목표를 작성해보세요.

### Note 4 키워드

참조와 할당  
성능  
빅오표기법(시간복잡도, 공간복잡도)

자신이 많이 활용한 것을 키워드로 정리해보세요.

# Sprint 1

## Python and Problem Solving

### N511

#### Topic 1

- find와 index의 차이점에 대해 설명해보세요.
- find와 index 메소드 처럼 파이썬에서 비슷한 역할을 하지만 사용방법/의도/상황 등이 다른 메소드에 대해 정리해오고 공유해보세요.

#### Topic 2

- N511 유어클래스 코드에 for문으로 작성된 코드 3가지 이상을 가져와서 while문으로 구현하고 코드를 설명해보세요. (4개 부터는 N511 유어클래스 코드가 아닌 다른 코드를 사용해도 됩니다.)

#### Topic 1

- 💡 Guide
  - 다양한 메소드를 활용하여 문제해결을 하기전 비슷한 기능을 하지만 차이가 있는 메소드들에 대해 알아보기

#### Topic 2

- 💡 Guide
  - 문제해결을 위한 반복문에 대해 익숙해지는 시간

### N512

#### Topic 1

disneyland() 메소드를 참고하여 현실에서 특정 공간에 비용을 받고 입장 처리를 도와주는 키오스크 메소드를 만들어보세요.  
(주의! 모든 코드는 완벽할 수 없습니다. 작은 기능부터 차근차근 버전업하면서 구현해보세요!)

#### Topic 2

로봇청소기의 기능을 의사코드 형태로 최대한 구체적으로 작성해보시오!

#### Topic 1

- 💡 Guide
  - 예시 메소드를 참조하여 현실에서 특정 공간에 비용을 받고 입장 처리를 도와주는 메소드를 만들어봅시다.
  - 키오스크 기능을 만든다
  - 조건문 우선순위를 어떻게 지정할지

#### Topic 2

- 💡 Guide
  - 오늘 배운 의사코드 작성법을 활용하여 로봇청소기를 제작해 봅시다.

## Sprint 1

# Python and Problem Solving

N513

### Topic 1

- 노트 [OOP와 의견들 > OOP와 일상생활]의 Bus, Car 클래스에 `__init__()` 메소드를 사용하지 않고 각각의 속성(speed, color, model 등)을 부여하는 코드를 작성해보세요.
- 그리고 `__init__()` 메소드의 역할에 대해 정의해보세요. 어떤 경우에 `__init__()` 메소드를 사용하는지에 대해 생각해보세요.

### Topic 2

- 오전에 만들었던 고릴라 클래스를 기반으로 Card 클래스를 생성하세요. 그리고 상속기능과 클래스 변수, 인스턴스 변수를 사용하여 아래의 기능을 구현해보세요.
- 클래스 구현 및 기능활용해보기

### Topic 1

- 💡 Guide
- 클래스만 선언한 경우
  - `__init__()` 메소드의 역할

### Topic 2

- 💡 Guide
- 상속 기능을 활용하여 카드 클래스 만들어보기

N514

### Topic 1

크기가 1,000,000이상인 리스트에서 모든 요소들을 하나씩 제거하는 연산인 `pop(0)`과 `pop()`을 사용해서 각각 어느 정도의 시간이 소요되었는지 `time` 모듈을 사용하여 시간을 측정해보고, 두 연산의 시간이 차이가 나는 이유에 대해서 설명해보세요.

### Topic 2

k진수에서 소수 개수 구하기 문제를 풀어보고 서로의 속도를 비교해보세요. 그리고 왜 속도가 빠른 코드는 빨랐을 지에 대해서 토론해보세요.

### Topic 1

- 💡 Guide
- `pop()`과 `pop(0)`의 실행시간 차이를 보고 차이가 발생하는 이유에 대해 알아봅시다.

### Topic 2

- 💡 Guide
- 해당 문제에서 속도 비교를 진행할 때 어느 부분(n진수 변경)에서 속도차이가 발생하는 지 속도를 알아보기

## Sprint 1

# Python and Problem Solving - FAQ

N511

### Question

`[1,2,3,4] < [1,2,4] == True` 입니다. 왜 그럴까요?

### Answer

- 리스트에서 인덱스에 대한 이해를 해야합니다.
- 리스트별 비교를 진행할 때 인덱스 순서대로 값을 비교합니다.
- 이에 따라 2번째 인덱스 값이 `[3] < [4]` 로 비교되므로 질문과 같은 결과값이 나옵니다.

N512

### Question

컴프리헨션과 기존 반복문과의 속도 차이가 있을까요?

### Answer

- 프로그래밍 문법의 사용범위를 생각해봅니다.
- 컴프리헨션은 반복문과 조건문을 모두 포함하는 복합적인 문법이고, 일반 반복문은 말 그대로 반복문의 역할만 하는 문법입니다.
- 이에 따라 컴프리헨션에서 조건이 많아질 수록 속도가 느려질 수 있습니다.



## Sprint 1

## Python and Problem Solving - FAQ

N513

### Question

다형성

### Answer

- OOP에서 다형성은 계층 구조의 상속 관계에서 상속받은 기능 외, 다른 기능을 추가적으로 제공하고자 할 때 사용합니다.
- 사용 시 장점
  - 프로그램 작성 코드량을 줄여줍니다.
  - 다형성을 잘 쓰면 if/else 문을 많이 줄일 수 있습니다.
- 코드의 가독성을 높여줍니다.

N514

### Question

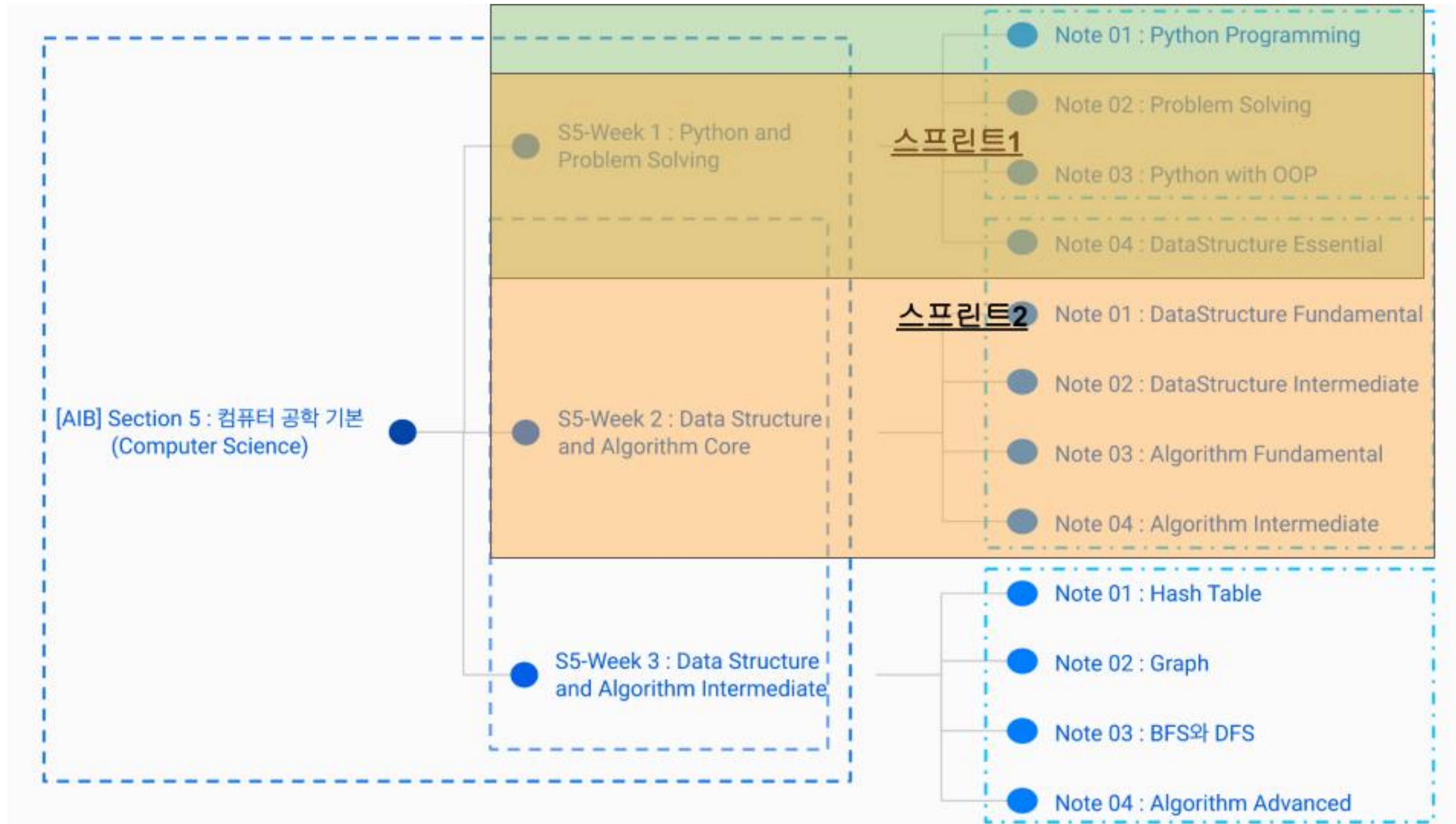
시간복잡도를 판단하는 기준은 어떤 것이 있을까요?

### Answer

- 반복문과 내부로직, 그리고 입력값에 따라 달라질 수 있는 소프트웨어 성능
- 1차적으로 가장 빠른 방법은 예를 들어 상수형태로 반복되면  $O(n)$ 으로 되는 것입니다.
  - 입력값 $n$ 에 따라 반복되는 로직을 함께 중점적으로 살펴봐야 합니다.
  - 특히 고려하셔야 될 부분은 입력되는 파라미터 $n$ 과 반복문이라고 보면 됩니다.
  - 중요한 것은  $O(n)$ ,  $O(n^2)$ 을 맞추는 것보다 내부로직에 영향을 주는 것이 무엇인지 살펴보는 것입니다.
  - 참고)알고리즘 평가기준 - 시간복잡도와 점근표기법- $O(n)$  : <https://nittaku.tistory.com/495>

## Sprint 1

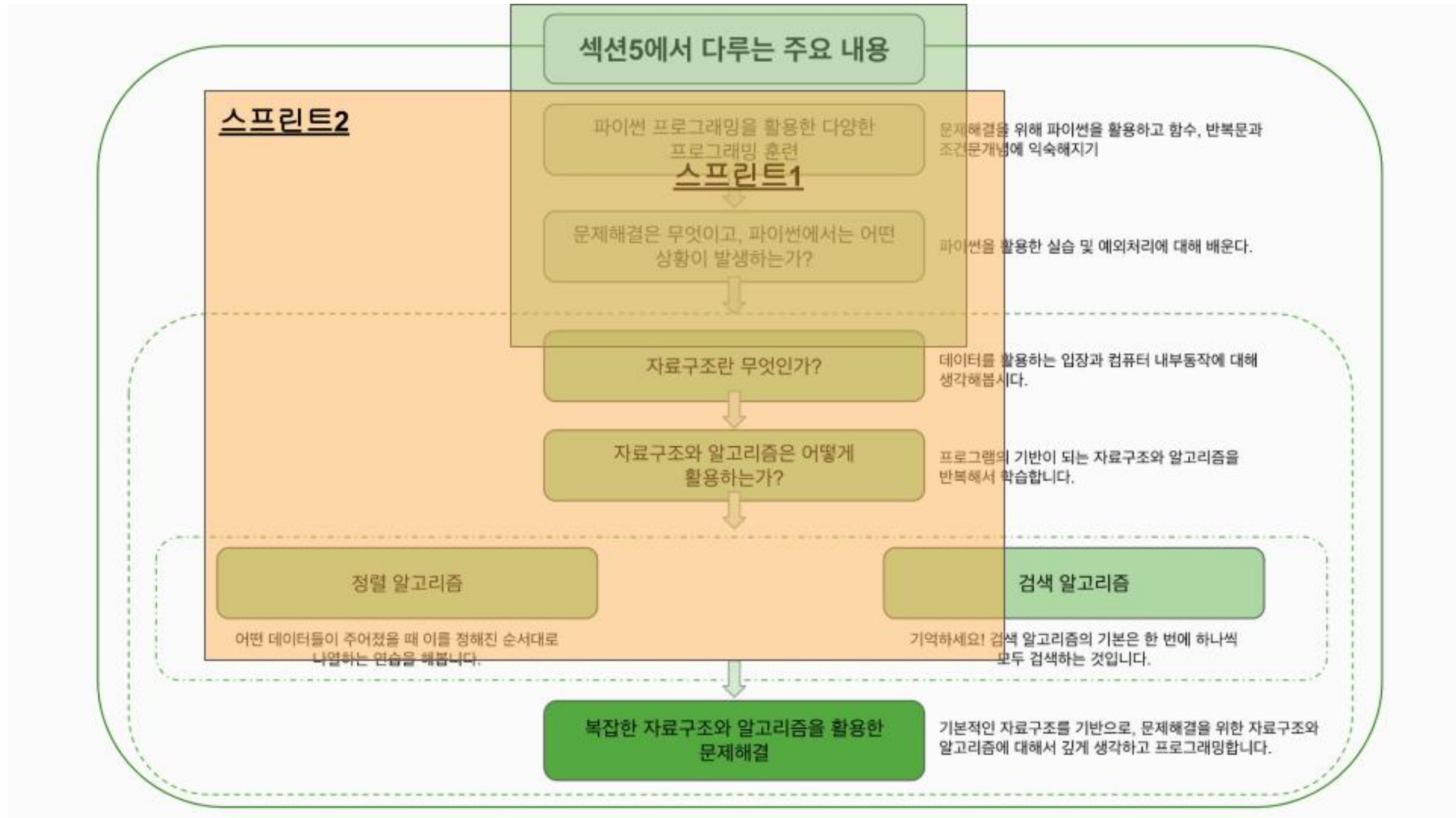
## Python and Problem Solving - Outro





# Sprint 1

## Python and Problem Solving - Outro



# 수고하셨습니다!