

프로그래밍 패러다임 : 함수형 프로그래밍

01 프로그래밍 패러다임

개념

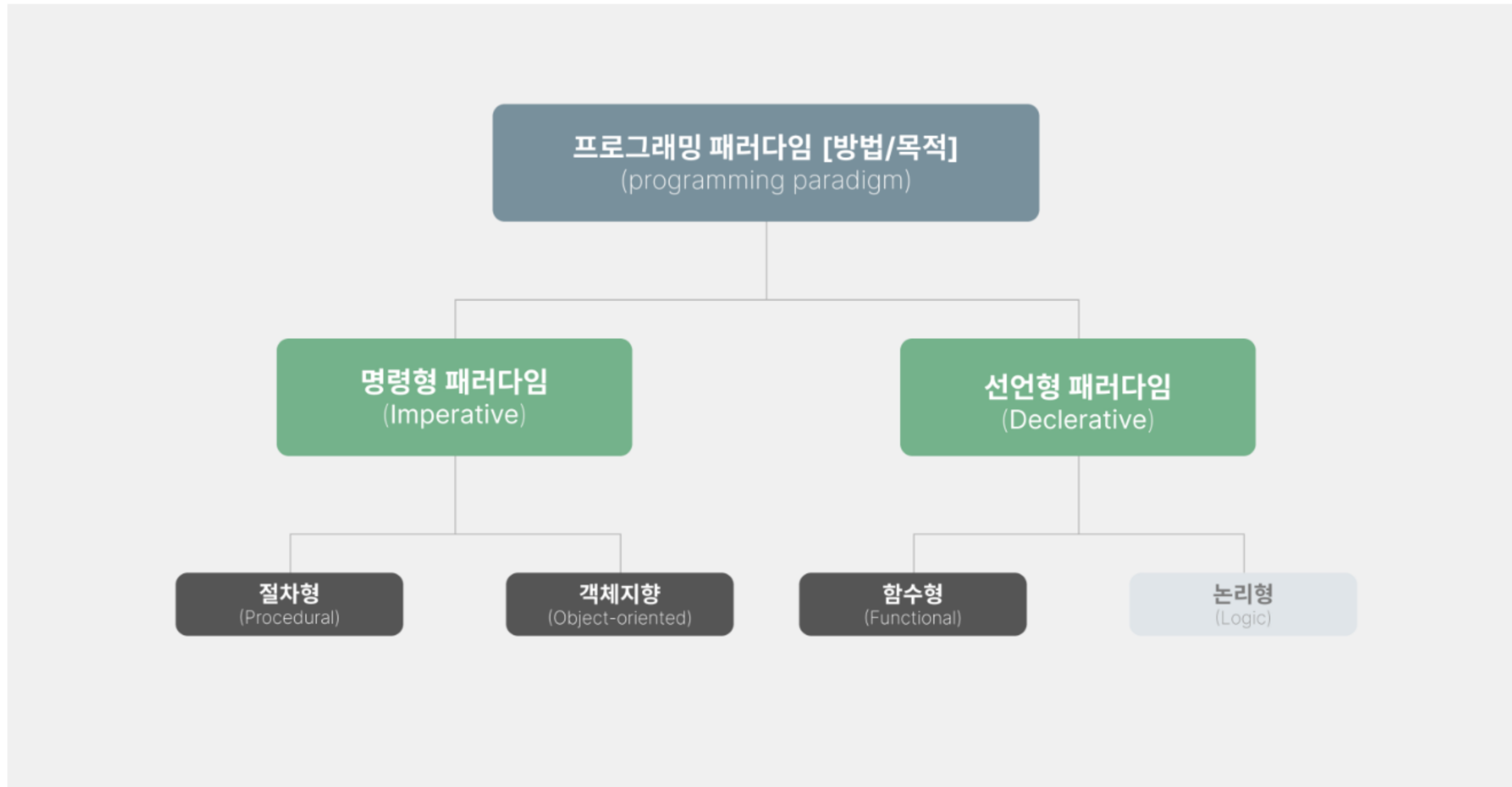
프로그래밍: 컴퓨터 프로그램을 작성하는 일

패러다임: 어떤 한 시대 사람들의 견해나 사고를 근본적으로 규정하고 있는 테두리로서의 인식 체계

프로그램을 개발하기 위해서 어떤 언어를 사용할지 어떤 것을 지양하거나 지향할지 등 다양한 방법으로 전략을 수립하는 데 필요한 모든 것들을 의미한다.

01 프로그래밍 패러다임

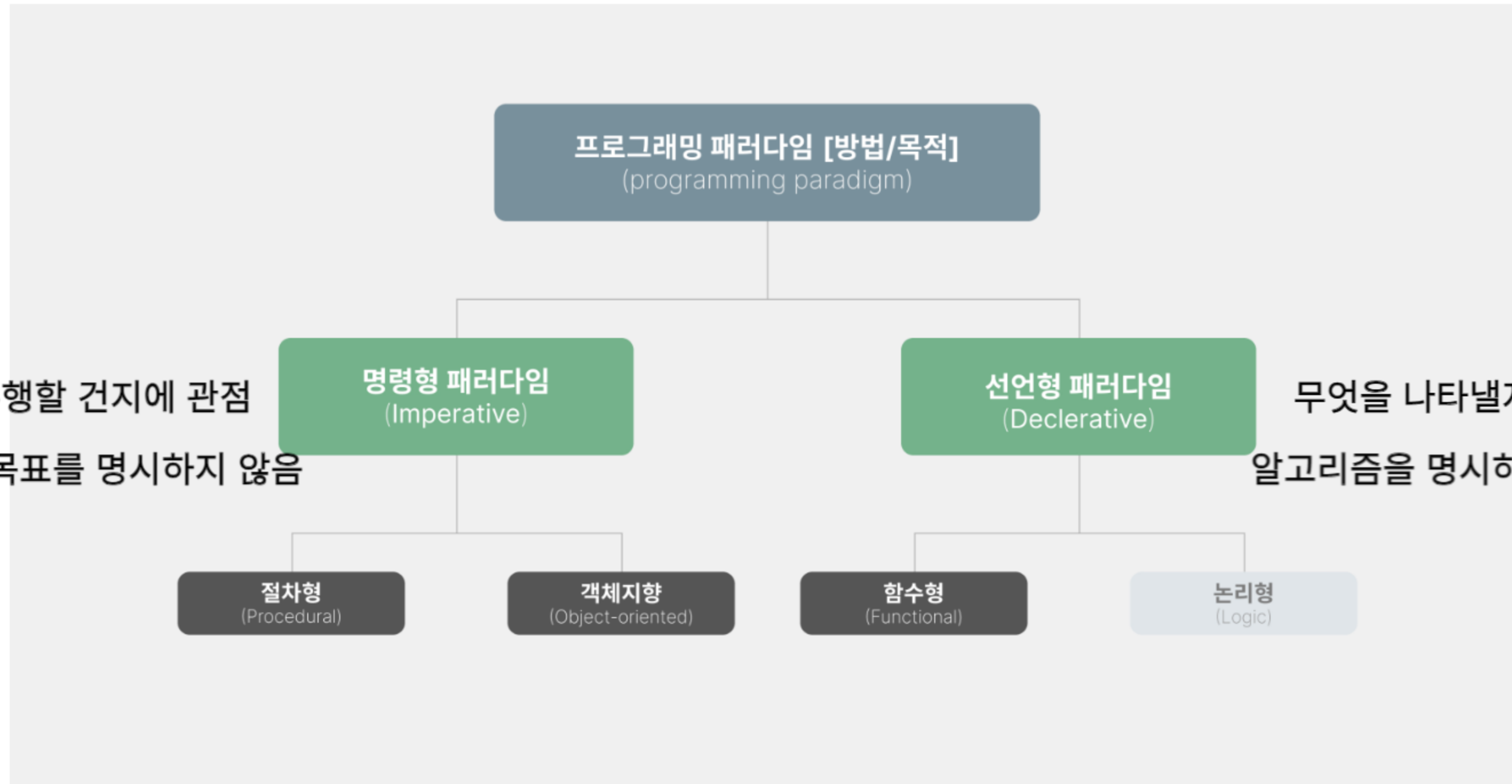
종류



01 프로그래밍 패러다임

종류

어떤 일을 어떻게 수행할 건지에 관점
알고리즘을 명시하고 목표를 명시하지 않음

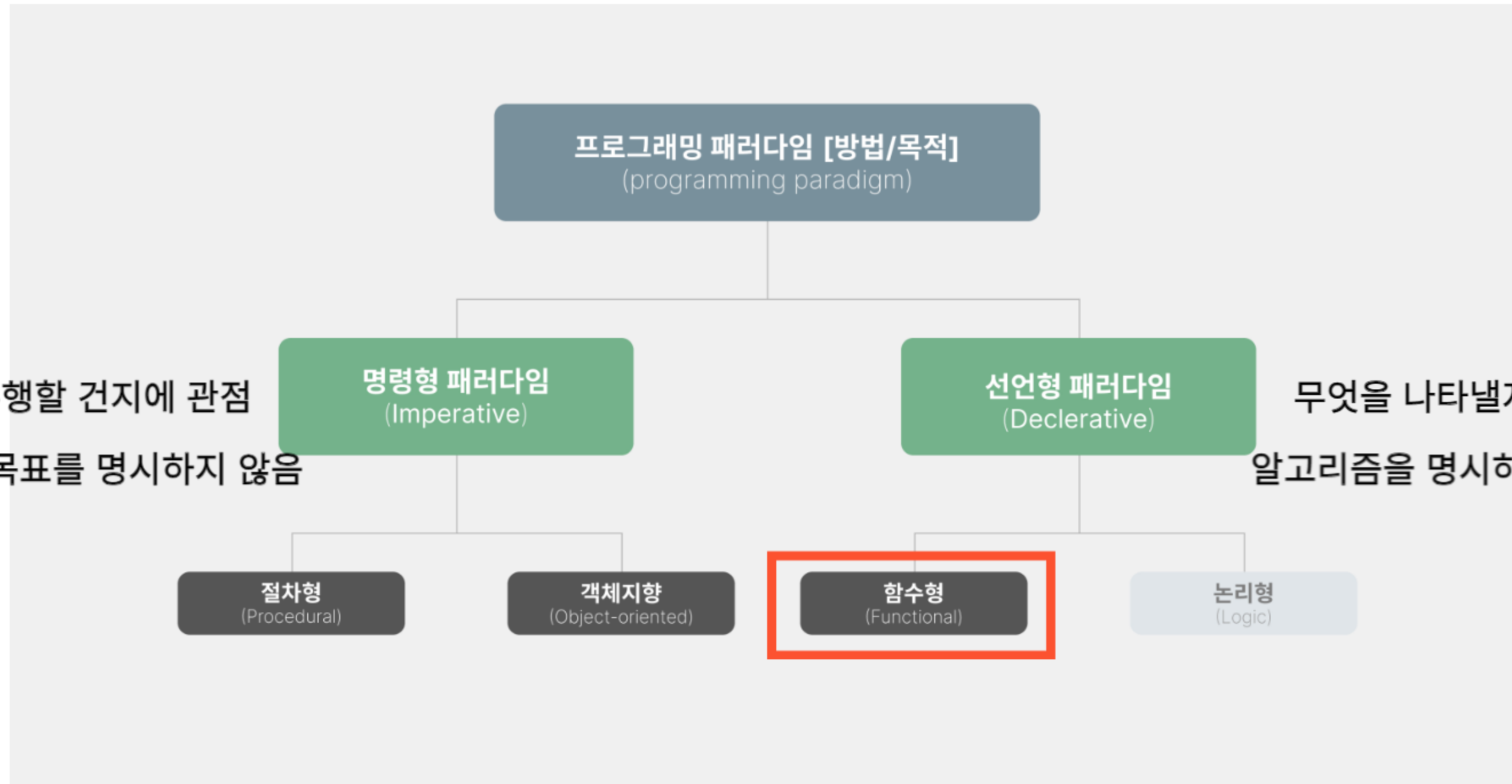


무엇을 나타낼지에 대해 관점
알고리즘을 명시하지 않고 목표만 명시함

01 프로그래밍 패러다임

종류

어떤 일을 어떻게 수행할 건지에 관점
알고리즘을 명시하고 목표를 명시하지 않음



무엇을 나타낼지에 대해 관점
알고리즘을 명시하지 않고 목표만 명시함

02 함수형 프로그래밍

함수형 프로그래밍 (functional programming)

상태 및 데이터 변경에는 사용하지 않으며 작은 문제를 해결하기 위한
'함수' 단위로 개발을 하는 기법

Java 1.8 버전 부터 함수형 프로그래밍을 한다.

함수형 프로그래밍을 위해 람다식, Stream API, Functional Interface의 주요한 기능들을 사용한다.

02 함수형 프로그래밍

대표적인 언어



클로저(Clojure)

LISP라는 독특한 걸모습의 프로그래밍 언어 패밀리 중 하나.

동적 타입 언어라서 구체적인 타입을 명시하지 않으며
순수 함수형 언어를 추구하지는 않아서 배우는 부담이 적은
함수형 프로그래밍 언어이다.

02 함수형 프로그래밍

대표적인 언어



스칼라(Scala)

명령형 프로그래밍에서 적극 활용되는 객체 지향 프로그래밍 방식과 함수형 프로그래밍 방식을 둘 다 활용한 프로그래밍 언어

02 함수형 프로그래밍

대표적인 언어



하스켈(Haskell)

정통 순수 함수형 프로그래밍 언어

특징으로는 패턴 맞춤, 커링, 조건 제시법, 가드, 연산자 정의 등이 있다.

단일체, 타입 클래스 등은 하스켈만의 독창적인 개념이며 이러한 특징들은 절차적인 프로그래밍 언어에서 매우 힘들었던 함수 정의를 쉽게 만들 수 있다.

02 함수형 프로그래밍

객체지향 프로그래밍과 함수형 프로그래밍 차이

프로그래밍의 덩치가 커지고 복잡해지면 많은 상태들을 관리하는 것에 대해 어려워진다.

객체지향 프로그래밍과 함수형 프로그래밍은 각자의 방식으로 이러한 상태들을 어떤 방식으로 구성하고 정리할 지에 대한 관점이다.

객체 지향 프로그래밍

프로그램을 객체들의 모임으로 구성하는 프로그래밍 패러다임

함수형 프로그래밍

프로그램을 함수들로 구성하는 프로그래밍 패러다임

02 함수형 프로그래밍

객체지향 프로그래밍과 함수형 프로그래밍 차이

프로그래밍의 덩치가 커지고 복잡해지면 많은 상태들을 관리하는 것에 대해 어려워진다.

객체지향 프로그래밍과 함수형 프로그래밍은 각자의 방식으로 이러한 상태들을 어떤 방식으로 구성하고 정리할 지에 대한 관점이다.

객체 지향 프로그래밍

프로그램을 객체들의 모임으로 구성하는 프로그래밍 패러다임

함수형 프로그래밍

프로그램을 함수들로 구성하는 프로그래밍 패러다임

이때, 함수형 프로그래밍은 객체지향 프로그래밍이나
절차적 프로그래밍의 반대 개념으로 생각하면 안됨!
대부분의 시스템은 이 세가지를 모두 사용하는 경향이 있다.

02 함수형 프로그래밍

관련 용어

순수 함수

순수 함수는 동일한 입력에 대해 항상 동일한 출력을 반환하며 외부에 영향을 주지 않는다.

불변성

함수형 프로그래밍에서는 데이터의 상태를 변경하는 대신 변경된 새로운 데이터를 생성한다.

고차함수

함수를 인자로 받거나 함수를 반환할 수 있는 함수이다. 이를 통해 코드를 더 간결하게 작성할 수 있다.

커리

여러개의 인자를 가진 함수를 하나의 인자를 받는 함수로 변환하는 기법이다.

02 함수형 프로그래밍

특징

- 함수형 프로그래밍은 과정보다 결과에 관심이 많다.
- 무엇이 실행 될 지를 강조한다.
- 데이터는 불변하다.
- 함수형 프로그래밍은 문제를 함수로 분해한다.
- 수학적 함수의 개념에 기반한다.
- 조건문이나 반복문을 지원하지 않는다.

02 함수형 프로그래밍

단점

학습곡선

함수형 프로그래밍은 명령형 프로그래밍과 상당히 다른 개념과 접근 방식을 사용한다. 따라서 개발자들이 새로운 개념들을 학습하는데 시간이 걸릴 수 있다.

추상화 수준

함수형 프로그래밍은 높은 수준의 추상화를 제공한다. 이로 인해 코드가 간결하고 모듈화 되긴 하지만 때로는 이해하기 어렵고 복잡할 수 있다.

02 함수형 프로그래밍

단점

성능

함수형 프로그래밍에서는 불변성과 순수함수를 사용해서 상태 변경을 최소화 한다. 그러나 메모리 사용량이 증가하거나 가비지 컬렉션 비용이 높아질 수 있다.

표현력

일부 애플리케이션에서는 명령형 프로그래밍이 더 직관적이고 표현력이 높을 수 있다. 특히 상태 변경이 빈번할 경우 함수형 프로그래밍으로 표현하기 어려운 경우가 있을 수 있다.

03 출처

<https://daeun28.github.io/%EC%BB%B4%ED%93%A8%ED%84%B0%EA%B3%B5%ED%95%99-%EC%8A%A4%ED%84%B0%EB%94%94/post23/>
<https://iosdevlime.tistory.com/entry/CSBasic-%EC%A2%80-%EB%8D%94-%EB%82%98%EC%9D%80-%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%A8%EC%9D%84-%EC%9C%84%ED%95%B4-%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D-%ED%8C%A8%EB%9F%AC%EB%8B%A4%EC%9E%84>
<https://velog.io/@khai96/%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D-%ED%8C%A8%EB%9F%AC%EB%8B%A4%EC%9E%84-Programming-Paradigm>
<https://medium.com/happyprogrammer-in-jeju/%ED%95%A8%EC%88%98%ED%98%95-%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D-%EC%96%B8%EC%96%B4-3%EB%8C%80%EC%9E%A5-d97729e97d4c>
<https://thecho7.tistory.com/entry/%EB%A9%B4%EC%A0%91-%EA%BF%80%ED%8C%81-%ED%95%A8%EC%88%98%ED%98%95-%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8DFunctional-Programming%EC%9D%B4%EB%9E%80>
<https://code-lab1.tistory.com/245>