

# 연산자, 표현식, 흐름 제어

## CSED101 프로그래밍과 문제해결

Student ID: 20240505

Name: 공현성

POVIS ID: hyunseong

Phone Number: 010-9649-3020

# 3월 수업 일정

03/08 (토)

03/15 (토)

03/22 (토) (TODAY)

03/29 (토)

# Content

1. 연산자와 표현식
2. 조건문
3. 반복문

03/08	OT, 파이썬 설치, 개념 복습
03/15	입출력, 변수, 타입
03/22	연산자, 표현식, 흐름제어 (today)
03/29	수업
04/05	중간고사 대비
04/12	수업
04/19	수업
04/26	수업
05/03	수업
05/10	수업
05/17	수업
05/24	기말고사 대비

# 연산자

연산자(operator)는 연산을 수행할 수 있도록 파이썬에서 정의한 기호, 키워드이다.

우리가 앞선 코드들에서 봤을 +, - 등이 바로 그 예시이다.

연산자는 다음과 같은 종류가 있다.

- ① 산술(arithmetic): 수의 연산을 다루는 연산자
- ② 관계(relational): 양 변의 관계를 다루는 연산자
- ③ 논리(logical): 논리식을 다루는 연산자
- ④ 대입(assignment): 대입 연산을 수행하는 연산자
- ⑤ 단축 대입: 산술 연산 + 대입 연산을 동시에 수행하는 연산자

# 연산자 - 중첩된 연산

연산자는 **여러번 중첩**하여 사용할 수 있다. 이때는 **연산의 순서(우선순위)**가 중요해진다.

산술 연산자: 기본적인 연산 순서를 따른다.

관계 연산자:  $a < b < c$ 와 같이 중복하여 쓰는 경우, 2개씩 쪼개어져 and 연산으로 바뀐다( $a < b$  and  $b < c$ ).

논리 연산자: 왼쪽부터 하나씩 연산이 수행된다. 가령 and, or가 여러개 중첩되어 있는 경우,

① A and B: A가 False면 B를 보지 않고 False 반환.

② A or B: A가 True면 B를 보지 않고 True 반환.

대입 연산자:  $a = b = 2$ 와 같이 중첩하여 쓰는 경우, 가장 오른쪽부터 순서대로 연산이 수행된다.

단축 대입 연산자: 중첩할 수 없다.

# 표현식과 구문

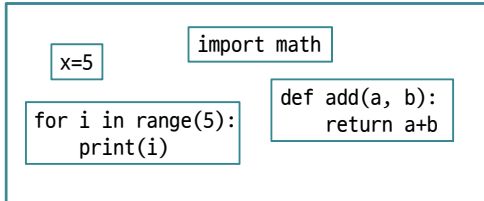
표현식(expression)은 값을 산출하는 코드 조각이다.

구문(statement)은 실행가능한 프로그램의 단위이다.

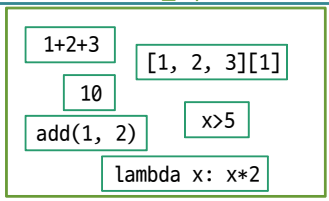
표현식은 항상 값을 산출해야 하지만 구문 같은 경우에는 값을 산출하지 않을 수도 있다.

구문에는 연산, 조건문, 반복문, 함수, 모듈 import 등이 포함된다.

구문



표현식



# 제어문

제어문(control flow statement)은 프로그램의 실행 흐름을 관리하는 구문이다. 크게 2종류가 있다.

## ① 조건문

- (a) if
- (b) if-else

## ② 반복문

- (a) for
- (b) while

# 제어문 - 조건제어

특정한 조건을 만족했을 때, 내부 코드를 실행하는 구문을 **조건 제어문(conditional flow)**이라고 한다.

- **if**: 주어진 조건문이 참이면 내부 코드를 실행. 거짓이면 무시.
- **if-else**: 주어진 조건문이 참이면 내부 코드를 실행. 거짓이면 다른 코드를 실행.

**elif** 문을 이용해 조건의 분기를 더 세부적으로 할 수 있다.

※ 추가: 제어문을 활용하는 특별한 예시가 있다.

모든 파이썬의 파일(모듈)은 `__name__`이라고 하는 이미 정의된 변수를 가지고 있다.

모듈이 import되면 `__name__`은 **모듈명**으로 설정되는 반면,  
모듈이 곧바로 실행되면 `__name__`이 `"__main__"`로 설정된다.

이를 이용하여 파일이 곧바로 실행되는 것을 감지하여 별도의 코드를 실행하게 할 수 있다.

# 제어문 - 반복제어

반복문은 특정 조건이 만족하는 상황에서 주어진 코드를 반복적으로 실행하는 구문이다.

- **while**: 주어진 조건문이 참이면 거짓이 될 때까지 내부 코드를 반복하여 실행.
- **for**: 리스트 같은 연속된 자료의 모임인 데이터를 **시퀀스형 데이터(sequence data)**라고 한다. for문은 시퀀스형 데이터를 순회하며 각 요소에 대해 작업을 수행.

**range(n)**, **range(a,b,k)** 함수를 사용하면, for문을 더 유연하게 사용할 수 있다.



# 반복제어의 조작

반복문 도중에 조건문을 이용해 반복문을 강제로 종료시킬 수 있다.

- **continue**: 반복의 현재 단계를 건너뛴다.
- **break**: 반복을 아예 중단시킨다.

위의 두 키워드는 조건문을 무시하기에, 보통 위의 두 키워드는 반복문에 새로운 조건을 추가하는 목적으로 조건문 안에 넣어 사용한다.

Lecture3 끝  
Q/A?