

Chapter 1. 연산자

- [학습 목표]
- 산술, 논리, 문자열 연산자의 개념을 이해하고 직접 사용할 수 있다.
- [명령어 정의]
- - +, -, *, / : 덧셈, 뺄셈, 곱셈, 나눗셈
- - //, %, **: 몫, 나머지, 제곱
- - and, or, not : 논리 연산자
- - 문자열 * 숫자, 문자열 + 문자열 : 문자열 반복과 연결

[예제 코드]

```
a = 10  
b = 3  
print(a + b)  # 13  
print(a // b) # 3  
print(a ** b) # 1000
```

```
x = True  
y = False  
print(x and y) # False  
print(not x)   # False
```

```
text = "Hi"  
print(text * 3) # HiHiHi  
print(text + "!") # Hi!
```

Chapter 2. 문자열

- [학습 목표]
- 문자열의 선언, 조작, 출력 방법을 익히고 다양한 함수를 활용할 수 있다.
- [명령어 정의]
 - 문자열 선언: `""`, `"`
 - `\n` : 줄바꿈
 - `split()` : 문자열 나누기
 - `strip()` : 앞뒤 공백 제거
 - `upper()`, `lower()` : 대소문자 변환
 - `format()` : 문자열 안에 값 넣기

[예제 코드]

```
name = " Alice "  
print(name.strip())      # 'Alice'  
print(name.upper())      # ' ALICE '  
print("안녕, {}!".format("철수")) # 안녕, 철수!
```

```
sentence = "사과,바나나,포도"  
fruits = sentence.split(",")  
print(fruits) # ['사과', '바나나', '포도']
```

```
print("줄바꿈  
예시입니다")
```

Chapter 3. 리스트

- [학습 목표]
- 리스트의 생성, 인덱싱, 슬라이싱, 정렬 등의 기초와 2차원 리스트를 이해한다.
- [명령어 정의]
- - 리스트 생성: []
- - 인덱싱: list[0], 슬라이싱: list[1:3]
- - append(): 요소 추가
- - sort(): 정렬
- - 2차원 리스트: [[1,2],[3,4]]

[예제 코드]

```
nums = [5, 2, 9, 1]
nums.append(7)
print(nums)      # [5, 2, 9, 1, 7]
print(nums[0])   # 5
print(nums[1:3]) # [2, 9]
```

```
nums.sort()
print(nums)      # [1, 2, 5, 7, 9]
```

```
matrix = [[1,2], [3,4]]
print(matrix[1][0]) # 3
```

Chapter 4. 조건문 (if)

- [학습 목표]
- 조건에 따라 실행 흐름을 바꾸는 if문 구조를 익히고 다양한 조건을 다룰 수 있다.
- [명령어 정의]
 - – if, elif, else: 조건 분기
 - – 비교 연산자: >, <, ==, !=
 - – 논리 연산자: and, or, not
 - – 삼항 연산자: x if 조건 else y

[예제 코드]

```
age = 13
if age >= 18:
    print("성인입니다.")
else:
    print("청소년입니다.")
```

```
score = 75
if score >= 90:
    print("A")
elif score >= 80:
    print("B")
else:
    print("C")
```

```
a = 3
b = 5
max_num = a if a > b else b
print(max_num) # 5
```


Chapter 5. 반복문 (for, while)

- [학습 목표]
- 같은 동작을 반복 수행하는 구조를 이해하고 반복 제어문을 사용할 수 있다.
- [명령어 정의]
- – *for i in range(n)*: 반복
- – *while* 조건: 반복
- – *break*: 반복 중단
- – *continue*: 다음 반복으로 건너뛸

[예제 코드]

```
for i in range(5):  
    print(i) # 0부터 4까지 출력
```

```
n = 1  
while n <= 3:  
    print(n)  
    n += 1 # 1, 2, 3 출력
```

```
for i in range(1, 10):  
    if i % 2 == 0:  
        continue  
    print(i) # 홀수만 출력
```

Chapter 6. 함수

- [학습 목표]
- 함수를 정의하고 호출하는 방법을 익히며 코드를 구조적으로 작성할 수 있다.
- [명령어 정의]
 - - def 함수이름(매개변수):
 - - return: 결과값 반환
 - - 함수 호출: 함수이름(인자)

[예제 코드]

```
def say_hello(name):  
    print("안녕,", name)  
say_hello("철수")
```

```
def add(x, y):  
    return x + y  
result = add(3, 5)  
print(result) # 8
```

Chapter 7. 클래스

- [학습 목표]
- 객체지향 개념을 익히고 클래스를 사용해 속성과 기능을 정의할 수 있다.
- [명령어 정의]
 - - class: 클래스 정의
 - - __init__: 생성자
 - - self: 객체 자기 자신 참조
 - - 메서드: 클래스 내부 함수

[예제 코드]

```
class Student:
    def __init__(self, name, age):
        self.name = name
        self.age = age
    def introduce(self):
        print("저는", self.name, "이고", self.age, "살입니다.")

s1 = Student("지민", 12)
s1.introduce() # 저는 지민이고 12살입니다.
```