

Day5 - 함수와 자료구조

학습 목표 - 함수, 리스트, 튜플, 딕셔너리 선언 및 사용

* 함수(Function)

특정 작업을 수행하는 코드 블록.

재사용성, 가독성, 유지보수성을 높이기 위해 사용.

```
def 함수명(매개변수):  
    실행할 코드  
    return 반환값
```

함수는 호출할 때 실행되며, return 문을 통해 결과를 반환할 수 있습니다.

함수는 호출할 때 실행되며, return 문을 통해 결과를 반환할 수 있음.

(반환할 값이 없다면 return문 사용 안함)

매개변수(Parameter): 함수에 입력으로 전달되는 값을 저장하는 변수.

반환값(Return Value): 함수 실행 결과로 돌려주는 값.

기본값 매개변수(Default Parameter): 매개변수에 기본값을 설정할 수 있음.

```
def greet(name="Guest"):  
    print(f"Hello, {name}!")
```

키워드 인수(Keyword Arguments): 함수 호출 시 매개변수 이름을 명시하여 전달

```
def minus(x, y):  
    return x - y  
result = minus(y=3, x=5)
```

*자료구조

데이터를 어떻게 담고, 정리하고, 꺼낼 것인가를 체계적으로 설계하는 방법

데이터를 쉽게 추가하거나 삭제

빠르게 원하는 데이터를 찾을 수 있음

프로그램의 성능과 효율성을 크게 향상시킴

자료구조	설명	특징
리스트(List)	순서가 있는 데이터 집합	변경 가능(mutable), 중복 허용
튜플(Tuple)	순서가 있는 데이터 집합	변경 불가(immutable), 중복 허용
딕셔너리(Dictionary)	키(key)-값(value) 쌍으로 저장	키는 중복 불가, 값은 중복 가능
세트(Set)	중복 없는 데이터 집합	순서 없음, 중복 자동 제거

* 리스트(List)

여러 데이터를 순서대로 저장하는 자료구조.

대괄호 []로 표현하며, 수정 및 추가가 가능.

```
fruits = ["사과", "바나나", "체리"]
print(fruits[0])
```

* 튜플(Tuple)

리스트와 유사하지만, 한 번 생성하면 수정할 수 없는 자료구조.
소괄호 ()로 표현.

```
colors = ("빨강", "초록", "파랑")
print(colors[1])
```

* 딕셔너리(Dictionary)

키(key)와 값(value) 쌍으로 데이터를 저장하는 자료구조.
중괄호 {}로 표현하며, 키를 통해 값을 조회.

```
student = {"이름": "김솔데스크", "나이": 20}
print(student["이름"])
```

예제 1 - 두 수를 입력받아 덧셈하는 함수 만들기

```
def add(a, b):
    return a + b
```

```
num1 = int(input("첫 번째 숫자: "))
num2 = int(input("두 번째 숫자: "))
print("두 수의 합:", add(num1, num2))
```

예제 2 - 과목별 점수 저장 후 출력

```
scores = {"수학": 90, "영어": 85, "과학": 95}
for subject, score in scores.items():
    print(f"{subject}: {score}점")
```

QUIZ - 함수와 자료구조

1. 함수는 왜 사용하는가요?
2. return 키워드의 역할은 무엇인가요?
3. 리스트와 튜플의 차이점을 간단히 설명하세요.

4. 딕셔너리에서 key와 value란 무엇인가요?
5. 다음 중 함수 이름으로 적절한 것은?
(a) 123function (b) add_numbers (c) my function
6. 세 개의 숫자 중 최댓값을 반환하는 함수 작성하기
7. 문자열 리스트를 받아 하나의 문자열로 합치는 함수 작성하기
8. 숫자 리스트를 받아 짝수만 반환하는 함수 작성하기
9. 두 개의 리스트를 합쳐 하나의 리스트로 반환하는 함수 작성하기
10. 단어를 입력받아 글자 수를 반환하는 함수 작성하기