

# Structured Types & Functions

정지현

# Index

- dictionary
- set
- function

# dictionary

- key : value 형식의 데이터 집합
- 선언
  - `dic = {}`
  - `dic = dict()`

# 활용

- 원소 추가
  - 사전 이름[key] = value
  - 사전 이름.update(사전)
- 모든 key
  - 사전 이름.keys()
- 모든 value
  - 사전 이름.values()

```
student = {}  
  
student[20171701] = "정지현"  
student.update({20180000: "홍길동"})  
print(student)  
print(student.keys())  
print(student.values())
```

# set

- `s = set()`
- 원소 추가
  - 집합 이름.add(원소)
- 연산
  - 합집합('|'), 교집합('&'), 차집합('-'), 대칭 차집합('^')

# functions

- 수학에서의 함수랑 크게 다르지 않다.
- 값을 넣으면(필수는 아니지만) 그 값에 대한 값을 출력해줌.
- 함수 선언은 def 키워드 사용.
- 선언 형식은 def 함수이름(파라미터):
- 함수 선언 아래에 들여쓰기된 블록이 함수의 몸체

# 함수를 쓰는 이유

- 반복 호출
- 코드의 가독성 (중복된 코드 제거)

# 인자와 매개변수

- 인자
  - argument라고도 함
  - 함수 호출할 때 넣는 값
- 매개변수
  - parameter라고도 함
  - 함수에 존재하는 외부에서 받아오는 값을 위한 변수



# 실습

- 함수를 이용해서 dictionary에 값을 넣어보자.
- dictionary는 맨 위에 선언
- 함수 파라미터는 k와 v
- k는 사전의 key, v는 사전의 value
- 테스트
  - key는 "id"와 "name"
  - value는 본인의 학번과 이름
  - 함수를 사용해서 추가한 후 출력해보세요.