Structured Types 2

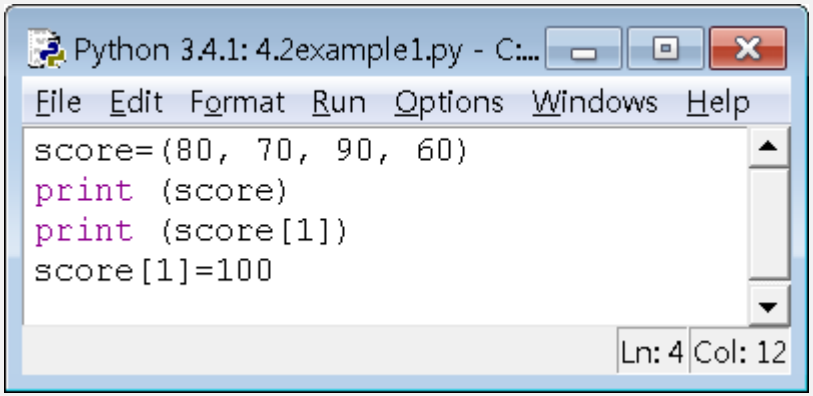
국민대 소프트웨어학부

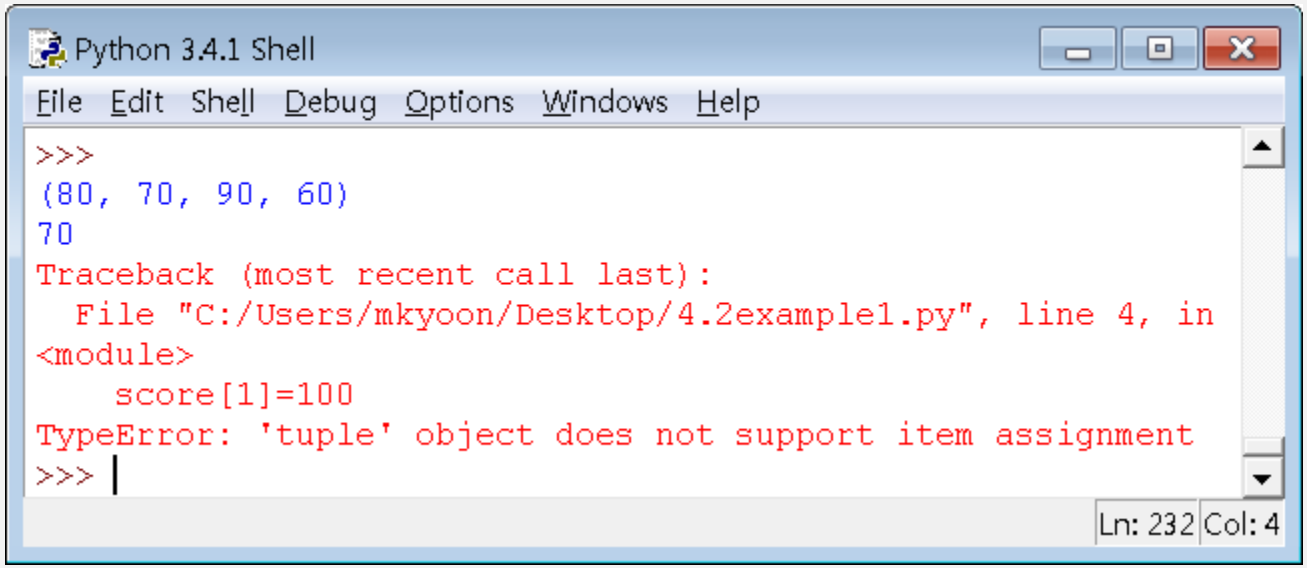
수업목표

- Tuples
- Dictionaries
- Sets

Tuples

- 변경 불가능한 리스트
  - 추가, 수정, 삭제 불가





Tuples

- 튜플 대입 연산
  - 튜플에서 여러 개의 변수로 한번에 값을 대입하는 기능

```
student = (20153360, "김영재", 23)

id, name, age = student

print(id)
print(name)
print(age)
```

Dictionaries

- 사전
  - 키(key)와 값(value)으로 구성된 집합 (비순차 데이터 타입) → 사전
  - 유일한 키로 검색
    - 비순차형이며 Index 접근 불가 (friends[2])
  - 프로그래밍 언어별로 사전, 맵(map), 테이블(table) 등 다양한 이름 존재

Dictionaries

- 사전 입력 방법
  - 사전이름={키1:값1, 키2:값2}
  - 사전이름[키]=값
  - 사전이름.update({키3:값3, 키4:값4})

Dictionaries

- 사전 입력 및 조회

Python 3.4.1: 4.2example2.py - C:/Users/mkyoon/Desktop/4.2example2.py

File Edit Format Run Options Windows Help

```
dic ={2015123: "Alice", 2015200: "Bob", "KMU": "JeongReung"}
dic[2015300]="Carole"
dic.update({2015400:"Dave", 2015500:"Eve"})
print(dic)
print (dic["KMU"])
print (dic[2015200])
print (dic["Alice"])
```

Ln: 4 Col: 10

Python 3.4.1 Shell

File Edit Shell Debug Options Windows Help

```
{2015200: 'Bob', 2015123: 'Alice', 2015300: 'Carole', 2015400: 'Dave', 201
5500: 'Eve', 'KMU': 'JeongReung'}
JeongReung
Bob
Traceback (most recent call last):
  File "C:/Users/mkyoon/Desktop/4.2example2.py", line 7, in <module>
    print (dic["Alice"])
KeyError: 'Alice'
```

Ln: 288 Col: 0

Dictionaries

- 사전 전체 검색 방법

Python 3.4.1: 4.2example3.py - C:/Users/mkyoon/Desktop/4.2example3.py

File Edit Format Run Options Windows Help

```
dic ={2015123: "Alice", 2015200: "Bob", "KMU": "JeongReung"}
for key in dic:
    print (key, "'s value is ", dic[key])
print()
for key, value in dic.items():
    print (key, "'s value is ", value)
```

Ln: 1 Col: 7

Python 3.4.1 Shell

File Edit Shell Debug Options Windows Help

```
2015200 's value is  Bob
2015123 's value is  Alice
KMU 's value is  JeongReung

2015200 's value is  Bob
2015123 's value is  Alice
KMU 's value is  JeongReung
>>>
```

Ln: 340 Col: 4

Dictionaries

- 사전 전체 검색 방법
  - 없는 키 접근 → 에러 발생

```
dic = {2015123: "Alice", 2015200: "Bob", "KMU": "JeongReung"}
for key in dic:
    print(key, "'s value is ", dic[key])
print()
for key, value in dic.items():
    print (key, "'s value is ", value)

print(dic[2017100])
```

```
2015123 's value is  Alice
2015200 's value is  Bob
KMU 's value is  JeongReung
Traceback (most recent call last):
  File "C:/Users/mkyoon/PycharmProjects/hello_world/hello.py", line 8, in <module>
    print(dic[2017100])
KeyError: 2017100

Process finished with exit code 1
```

Dictionaries

- 키 존재 확인
  - <키> in <사전> : <키> 가 <사전>에 있으면 True, 없으면 False
  - <사전>.get(<키>) : <키> 가 <사전>에 있으면 <키>에 해당하는 <값>을 리턴, 없으면 None 을 리턴

```
dic = {2015123: "Alice", 2015200: "Bob", "KMU": "JeongReung"}
for key in dic:
    print(key, "'s value is ", dic[key])
print()
for key, value in dic.items():
    print (key, "'s value is ", value)

if 2017100 in dic :
    print(dic[2017100])
else:
    print("key error")
```

Dictionaries

- 키 존재 확인
  - <키> in <사전> : <키> 가 <사전>에 있으면 True, 없으면 False
  - <사전>.get(<키>) : <키> 가 <사전>에 있으면 <키>에 해당하는 <값>을 리턴, 없으면 None 을 리턴

```
dic = {2015123: "Alice", 2015200: "Bob", "KMU": "JeongReung"}
for key in dic:
    print(key, "'s value is ", dic[key])
print()
for key, value in dic.items():
    print (key, "'s value is ", value)

id = dic.get(2017100)
print("ID:", id)
```

```
2015123 's value is  Alice
2015200 's value is  Bob
KMU 's value is  JeongReung

2015123 's value is  Alice
2015200 's value is  Bob
KMU 's value is  JeongReung
ID: None
```

Dictionaries

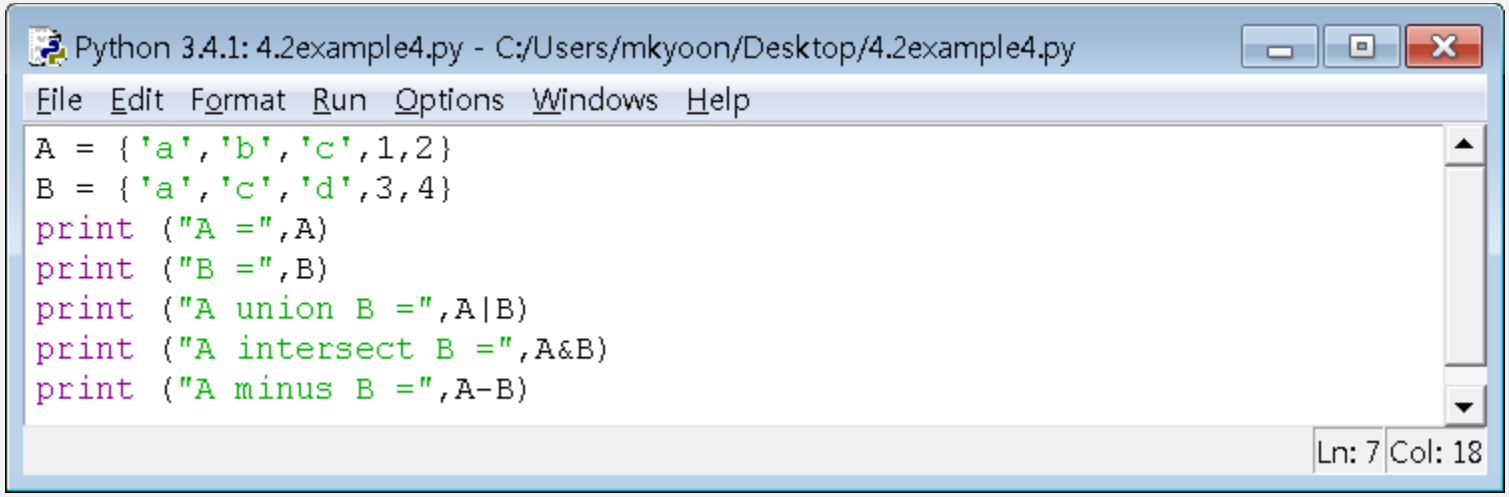
- 사전 삭제
  - del <사전>[<키>] : <키>를 <사전>에서 찾은 다음 삭제 한다.
  - <사전>.pop(<키>) : <키>를 <사전>에서 찾은 항목을 반환한 다음 삭제 한다.

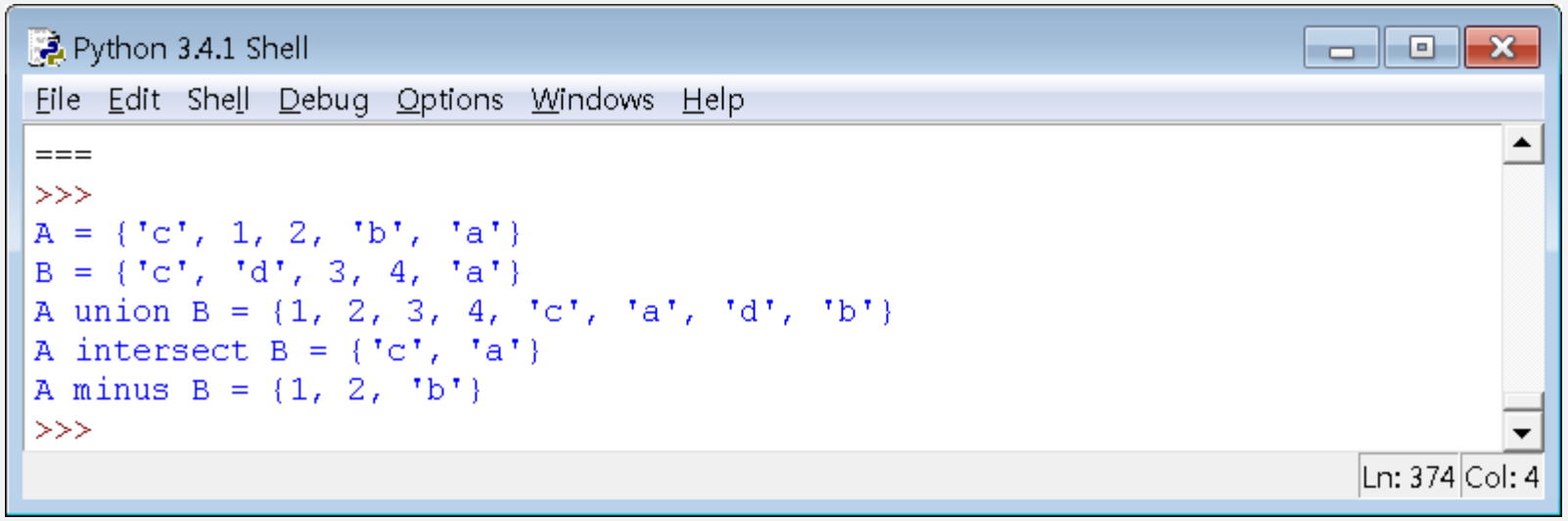
Sets

- 집합
  - 유일한 값들로 구성된 비순차적 자료구조
  - 수학에서의 집합 구현
  - 집합이름 = {값1, 값2,…}
  - 집합연산 지원
    - 합집합 연산: '|'
    - 교집합 연산: '&'
    - 차집합 연산: '-'
    - 대칭 차집합 연산: '^'

Sets

- 집합


Python 3.4.1: 4.2example4.py - C:/Users/mkyoon/Desktop/4.2example4.py
File Edit Format Run Options Windows Help
A = {'a','b','c',1,2}
B = {'a','c','d',3,4}
print ("A =",A)
print ("B =",B)
print ("A union B =",A|B)
print ("A intersect B =",A&B)
print ("A minus B =",A-B)
Ln: 7 Col: 18


Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
===
>>>
A = {'c', 1, 2, 'b', 'a'}
B = {'c', 'd', 3, 4, 'a'}
A union B = {1, 2, 3, 4, 'c', 'a', 'd', 'b'}
A intersect B = {'c', 'a'}
A minus B = {1, 2, 'b'}
>>>
Ln: 374 Col: 4

Sets

- 집합
- 원소 추가
  - <집합>.add(<원소>) : 집합에 <원소>를 넣는다.
  - <집합>.update([<원소>, <원소>, <원소>, ...]) : 집합에 리스트에 있는 <원소>들을 넣는다.

```
prime_numbers = { 2, 3, 5, 7 }

print("추가전 :", prime_numbers)
# 원소 한개를 추가 할때는 add를 사용 한다.
prime_numbers.add(11)
print("11 추가후 :", prime_numbers)
# 여러개의 원소를 추가 할 때는 update를 사용 한다.
prime_numbers.update([13, 17, 19])
print("[13, 17, 19] 추가후 :", prime_numbers)
```

```
추가전 : {2, 3, 5, 7}
11 추가후 : {2, 3, 5, 7, 11}
[13, 17, 19] 추가후 : {2, 3, 5, 7, 11, 13, 17, 19}
```

Sets

- 집합
- 원소 제거
  - <집합>.discard(<원소>) : <원소>가 집합에 있으면 삭제한다.
  - <집합>.remove(<원소>) : <원소>가 집합에 있으면 삭제하고, 없으면 에러를 발생시킨다.

```
prime_numbers = { 2, 3, 5, 7, 9, 11, 13, 15, 17, 19 }

print("원소 삭제 전 :", prime_numbers)
prime_numbers.discard(9)
print("9 삭제 후 :", prime_numbers)
prime_numbers.remove(15)
print("15 삭제 후 :", prime_numbers)
```

```
원소 삭제 전 : {2, 3, 5, 7, 9, 11, 13, 15, 17, 19}
9 삭제 후 : {2, 3, 5, 7, 11, 13, 15, 17, 19}
15 삭제 후 : {2, 3, 5, 7, 11, 13, 17, 19}
```

Sets

- 실습
  - 자카드 지수 (Jaccard index)
  - 집합 사이의 유사도를 측정하는 방법 중 하나이다. **자카드 계수**(Jaccard coefficient) 또는 **자카드 유사도**(Jaccard similarity)라고도 한다. 자카드 지수는 0과 1 사이의 값을 가지며, 두 집합이 동일하면 1의 값을 가지고, 공통의 원소가 하나도 없으면 0의 값을 가진다. 자카드 지수는 아래의 식으로 정의된다.
  - $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$
  - 아래 4개 집합으로 구성된 리스트에 대해서 모든 집합 쌍의 자카드 지수를 구하는 프로그램을 작성하시오.
    - allSet=[{1,2,3,4,5,6,7,8,9,10}, {1,3,5,7,12,15}, {3,12,15,18,20}, {12,13,14,15,16,17,18,19}]

https://ko.wikipedia.org/wiki/%EC%9E%90%EC%B9%B4%EB%93%9C_%EC%A7%80%EC%88%98

Dictionaries

- 실습
  - 그리스 문자를 “키=영문이름“, “값=한글이름” 으로 사전을 구현하라.

이 름	그리스 문자	
	소문자	대문자
알파 (Alpha)	α	A
베타 (Beta)	β	B
감마 (Gamma)	γ	Γ
델타 (Delta)	δ	Δ
엡실론 (Epdilon)	ε	E
제타 (Zeta)	ζ	Z
에타 (Eta)	η	H
세타 (Theta)	θ	Θ
요타 (Iota)	ι	I
카파 (Kappa)	κ	K
람다 (Lambda)	λ	Λ
뮤 (Mu)	μ	M

이 름	그리스 문자	
	소문자	대문자
뉴 (Nu)	ν	N
크사이 (Xi)	ξ	Ξ
오미크론 (Omicron)	ο	O
파이 (Pi)	π	Π
로 (Rho)	ρ	P
시그마 (Sigma)	σ	Σ
타우 (Tau)	τ	T
입실론(Upsilon)	υ	Y
피 (Phi)	φ	Φ
카이 (Chi)	χ	X
프사이 (Psi)	ψ	Ψ
오메가 (Omega)	ω	Ω