Section 1 시퀀스 자료형이란?

- **C, Java**
  - 같은 형의 데이터들을 배열array 자료구조로 표현

- **파이썬**
  - 배열array 기능 + 확장된 기능 = 시퀀스sequence 자료형 제공

Section 1 시퀀스 자료형이란?

- **파이썬에서 제공하는 시퀀스 자료형**
  - 문자열 : name='cskim' , home="seoul"
  - 리스트 : fname=['kim', 'lee', 'park', 'jung', 1, 2, 3, 4]
  - 튜플 : student=('computer', 3, 201933345, 'hongkildong')

- **문자열, 리스트, 튜플**
  - 표현 방법은 다름
  - 순서가 정해져 있는 자료형이라는 공통적인 특성

Section 2 시퀀스 자료형의 연산

시퀀스 자료형의 연산

1. 인덱싱 indexing

- 순차적인 자료구조에 인덱스(첨자) 값을 가지고 접근할 수 있는 기능

형식 `[i]  # i번째 값을 의미`

시퀀스 자료형의 연산

2. 슬라이싱 Slicing

- 시퀀스 자료형에서 일부분을 잘라내어 동일한 자료형으로 반환하는 기능

형식 [start:stop:step] #start부터 시작해서 stop 이전까지 step 간격으로 추출

3. 연결 Concatenation

- "+"연산자를 사용. 두 개의 자료를 연결. 새로운 시퀀스 자료형 생성

형식 자료형+자료형

4. 반복 Repetition

- "*"연산자를 사용하여 시퀀스 자료형을 원하는 만큼 반복

형식 자료형*반복횟수

시퀀스 자료형의 연산

5. 멤버 유무 검사

- 시퀀스 자료형에 특정 자료가 있는지를 알려주는 기능
- “in”연산자는 for 반복문에 효율적으로 사용

★형식 자료 in 자료형

6. 길이 정보

- 내장함수인 len() 함수를 이용하여 시퀀스 자료형의 길이 추출

★형식 len(자료형)

Section 3 문자열

- 시퀀스 자료형인 문자열은 파이썬에서 가장 많이 사용되는 자료형

1. 문자열 생성

```
>>> s1='I like Python'          # 작은 따옴표로 생성
>>>
>>> s2="I love Python"          # 큰 따옴표로 생성
>>>
>>> s3='''I really love Python'''          # 3개의 작은 따옴표로 생성
>>>
>>> s4='''I like Python
I love Python
I really love Python'''          # 3개의 작은 따옴표를 사용하여 여러줄의 문자열 생성
>>>
>>> print(s1,s2,s3)
I like Python I love Python I really love Python
>>> print(s4)
I like Python
I love Python
I really love Python
>>>
>>> s5=''                       # 작은 따옴표로 빈 문자열 생성
>>> s6=str()                    # str() 함수를 사용하여 빈 문자열 생성
>>> len(s5)                     # 빈 문자열의 길이
0
```

1. 문자열 생성

- **str() 함수**
  - **다른 자료형을 문자열 변환하여 새로운 문자열 생성**

```
>>> s1=str(1234)              # 정수로 문자열 생성
>>> s1
'1234'
>>> type(s1)                  # s1의 형?
<class 'str'>
>>>
>>> s2=str(123.456)           # 실수로 문자열 생성
>>>
>>> s3=str([1,2,3])           # 리스트로 문자열 생성. 리스트 자체를 문자열로 인식
>>> s3[0]                     # 문자열의 첫 번째 문자는?
'['
>>>
>>> s4=str((1,2,3))           # 튜플로 문자열 생성. 튜플 자체를 문자열로 인식
>>> s4[2]                     # 문자열의 세 번째 문자는?
','
```

2. 문자열 메소드

- 내장함수
  - 파이썬 언어 자체에서 제공되는 함수
- 메소드
  - 해당되는 객체에서 제공되는 기능

2. 문자열 메소드

문자열 메소드				
capitalize()	format_map()	isnumeric()	maketrans()	split()
casefold()	index()	isprintable()	partition()	splitlines()
center()	isalnum()	isspace()	replace()	startswith()
count()	isalpha()	istitle()	rfind()	strip()
encode()	isascii()	isupper()	rindex()	swapcase()
endswith()	isdecimal()	join()	rjust()	title()
expandtabs()	isdigit()	ljust()	rpartition()	translate()
find()	isidentifier()	lower()	rsplit()	upper()
format()	islower()	lstrip()	rstrip()	zfill()

2. 문자열 메소드

- 문자열의 대소문자 변환과 연관된 메소드
  - upper(), lower()
- 문자열에서 정렬 메소드
  - center(), ljust(), rjust()
- 특정 문자가 있는 위치를 찾아주는 메소드와 문자열의 시작과 끝을 확인할 수 있는 메소드
- 문자열을 결합하고 분리하기 위한 메소드
  - join(), split(), rsplit(), splitlines()

2. 문자열 메소드

- 불필요한 스페이스를 제거
  - strip(), rstrip(), lstrip()
- 문자열을 분할해서 튜플로 반환
  - partition(), rpartition()
- 문자열을 대체
  - replace()
- 문자열에 0을 채움
  - zfill()
- 문자열 객체에서는 문자열을 검사하여 참 또는 거짓을 반환하는 다양한 메소드를 제공
- 문자열 객체에서는 문자열을 검사하여 참 또는 거짓을 반환하는 다양한 메소드를 제공

실습 1

사용자로부터 입력받은 문장을 다음과 같이 출력하는 프로그램을 작성하시오.

실행결과

```
========== RESTART: D:/program/chap05/예제5-3-2-1.py ==========
영어 문장을 입력 : I love python
입력된 문장의 길이는 : 13
각 단어의 첫 문자를 대문자로 변환 : I Love Python
모든 글자를 대문자로 변환 : I LOVE PYTHON
문자열에 a가 몇번 나타났는가 : 0
입력된 문자열이 모두 문자로만 구성되었는가? : False
입력된 문자열이 모두 숫자로만 구성되었는가? : False
입력된 문자열이 모두 대문자 구성되었는가? : False
```

실습 2

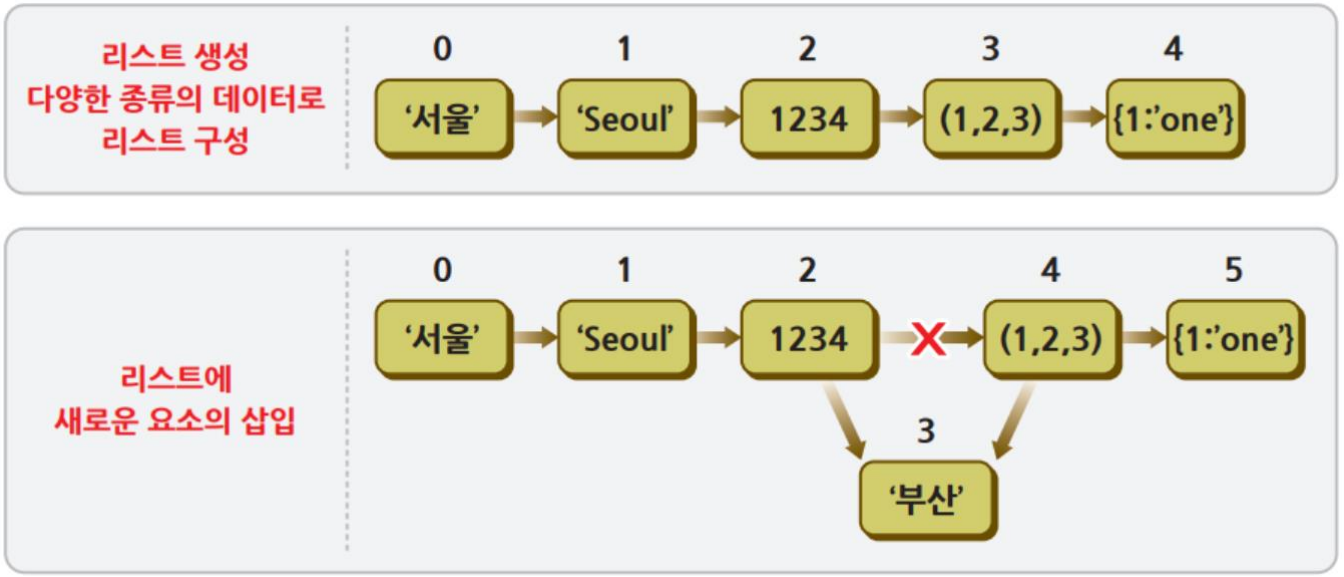
사용자로부터 입력받은 문장에서 스페이스가 몇 개 인지를 출력하고, 스페이스를 없앤 문장을 출력하는 프로그램을 작성하시오.

실행결과

```
문장을 입력하세요 : I like python I love Python
문장에서 스페이스 개수 : 5
스페이스가 삭제된 문장 : IlikepythonIlovePython
```

Section 4 리스트

- 리스트는 생성된 후에 변경이 가능한 자료형
- 동일한 형의 데이터는 물론 파이썬에서 제공하는 모든 자료형의 데이터를 요소로 가질 수 있다



1. 리스트 생성

- 리스트 생성
  - 대괄호를 사용하여 직접 생성
  - list() 함수를 사용하여 다른 자료형의 데이터를 리스트로 생성.
- 리스트 중간에 데이터를 삽입하고 삭제할 수 있습니다.

```
>>> list1=[1,2,3,4,5]                    # 리스트 생성
>>> list2=[1,'one',(1,2,3),"Seoul"]      # 다양한 자료를 가진 리스트 생성
>>>
>>> lang='Python'                        # 문자열 생성
>>> list3=list(lang)                     # 문자열을 리스트로 생성
>>> list3
['P', 'y', 't', 'h', 'o', 'n']
>>> type(list3)                          # list3의 형?
<class 'list'>
>>>
>>> tup1=(1,2,3,4,5)                     # 튜플 생성
>>> list4=list(tup1)                     # 튜플을 리스트로 생성
>>> list4
[1, 2, 3, 4, 5]
>>> type(list4)                          # list4의 형?
<class 'list'>
```

2. 리스트 메소드

- 리스트 메소드
  - 리스트 객체에서 제공되는 유용한 기능

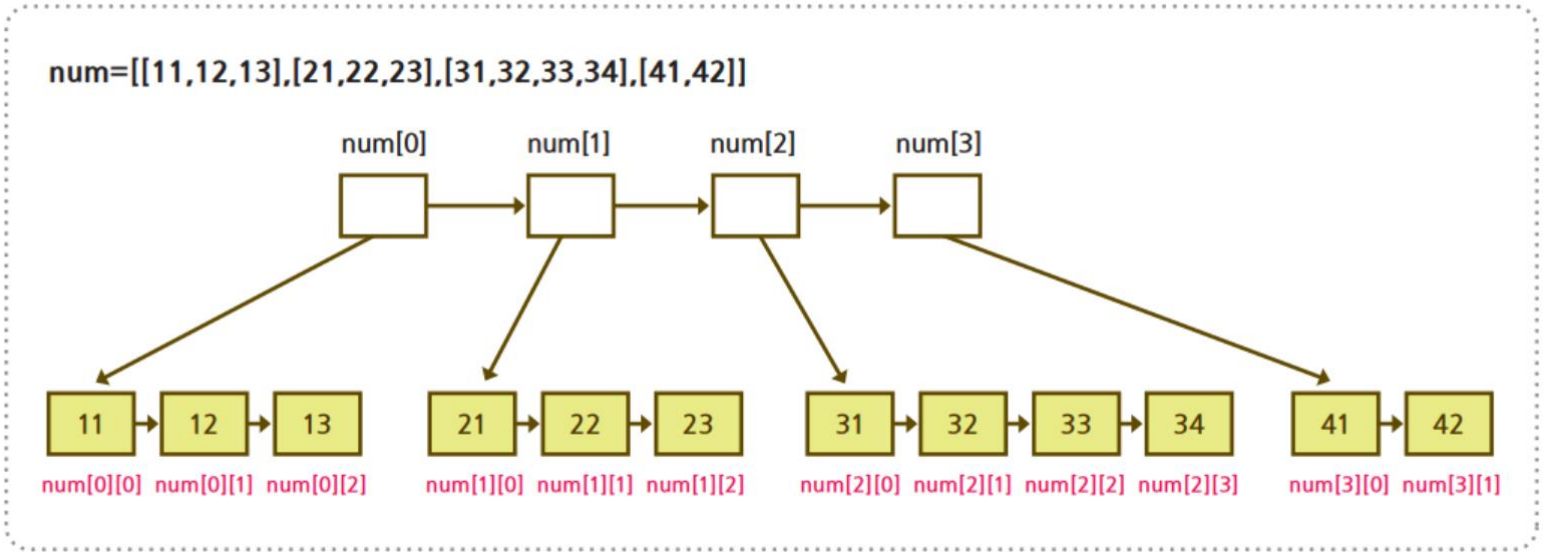
리스트 메소드			
append()	count()	insert()	reverse()
clear()	extend()	pop()	sort()
copy()	index()	remove()	

2. 리스트 메소드

- 데이터를 추가, 확장
  - append(), insert(), extend()
- 요소를 삭제, 개수를 반환
  - pop(), remove(), clear(), count()
- 리스트 복사
  - copy()
- 특정 요소의 위치를 반환, 정렬
  - index(), sort(), reversed()

3. 다중 리스트

- 리스트의 요소로 또 다른 리스트를 가질 수 있다



3. 다중 리스트

- 다중 리스트를 구성하고, 리스트 요소의 합을 구하는 예

```
>>> num=[[11,12,13],[21,22,23],[31,32,33,34],[41,42]]        # 다중 리스트 생성
>>> for j in range(len(num)) :            # 리스트 길이 만큼 반복
        sum=0
        for i in range(len(num[j])) :    # 리스트의 각 요소의 길이 만큼 반복
                sum = sum + num[j][i]    # 각 요소의 합

        print(j+1,'번째 줄의 합 : ',sum)  # 합을 출력


1 번째 줄의 합 :  36
2 번째 줄의 합 :  66
3 번째 줄의 합 :  130
4 번째 줄의 합 :  83
>>>
>>> comp=[1,2,['a','b','c',['A','B']]]        # 다중 리스트 생성
>>> comp[2]                                   # 세 번째 요소
['a', 'b', 'c', ['A', 'B']]
>>> comp[2][3]                                # 세 번째 요소의 네 번째 요소
['A', 'B']
>>> comp[2][3][1]                             # 세 번째 요소의 네 번째 요소의 두 번째 요소
'B'
>>> comp[2][3][2]                             # 세 번째 요소의 네 번째 요소의 세 번째 요소. 오류 발생
Traceback (most recent call last):
  File "<pyshell#1098>", line 1, in <module>
    comp[2][3][2]                            # 세 번째 요소의 네 번째 요소의 세 번째 요소. 오류 발생
IndexError: list index out of range
```

실습 1

다음과 같은 2차원 다중리스트가 있을 때, 각 줄의 합을 출력하고, 모든 요소 중에서 가장 작은 값을 구하는 프로그램을 작성하시오. 최소값을 구하는 min() 함수와 합을 구하는 sum() 함수를 사용하시오.

```
[[11,33,22,7],[77,2,90],[3,66,44,9,8]]
```

실행결과

```
1 번째 줄의 합은 : 73
2 번째 줄의 합은 : 169
3 번째 줄의 합은 : 130
리스트에서 가장 작은 값은 : 2
```

코딩 힌트

sum(), min() 함수를 이용하면 중첩된 반복문을 사용할 필요가 없게 됩니다.

실습 2

1부터 사용자가 입력한 숫자 사이의 소수prime number의 리스트와 개수가 몇 개인지를 출력하는 프로그램을 작성하시오.

소수 : 소수는 1이 아닌 자연수 중에서 1과 자기 자신 이외의 약수를 갖지 않는 수

실행결과 (31을 입력한 경우)

```
숫자를 입력 : 31
1부터 31 까지의 소수의 리스트 : [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31]
1부터 31 까지의 소수의 개수 : 11
```

실습 3

1부터 100사이의 정수 난수random number 10개를 발생시켜 리스트에 저장하고, 리스트에서 가장 큰 값과 가장 작은 값을 출력하고, 리스트 값의 합계와 오름차순으로 정렬된 리스트를 출력하는 프로그램을 작성하시오.

파이썬에서 난수를 발생시키기 위해서는 랜덤 모듈을 사용해야하며, 랜덤 모듈을 사용하기 위해서 프로그램의 처음에 "import random"을 포함시켜야 합니다. 또한 1부터 100사이의 정수 난수를 구하는 메소드로는 random.randint(1,100)를 사용합니다.

실행결과

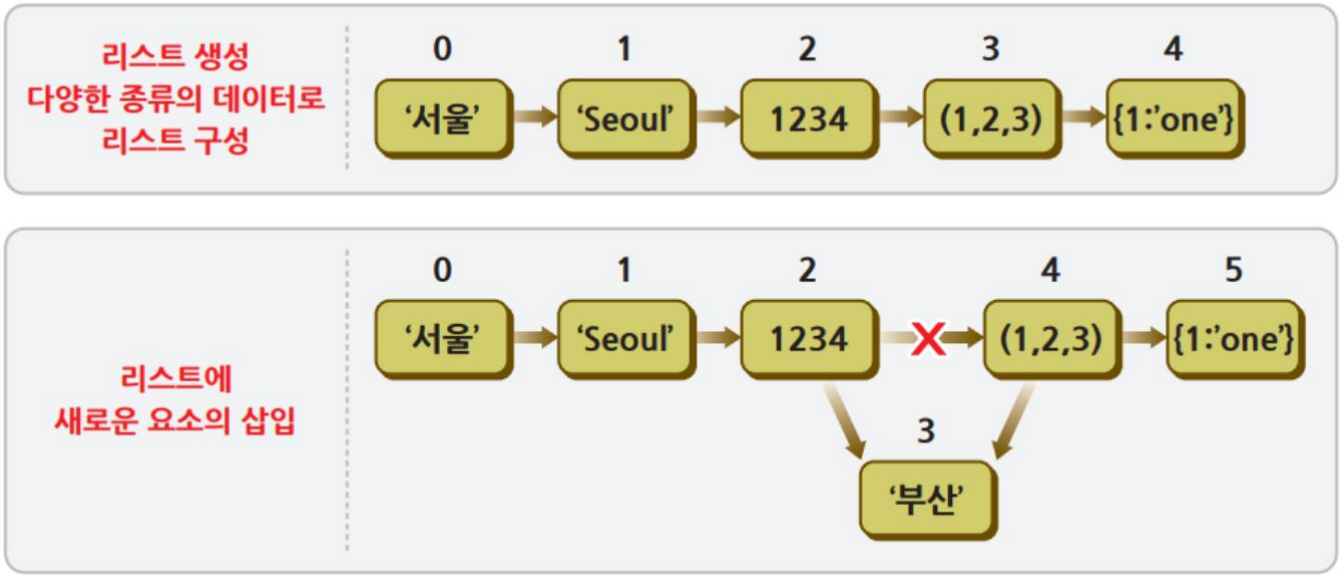
```
생성된 리스트 : [27, 74, 76, 65, 17, 85, 66, 83, 84, 79]
가장 큰 값 : 85
가장 작은 값 : 17
전체 요소의 합 : 656
정렬된 리스트 : [17, 27, 65, 66, 74, 76, 79, 83, 84, 85]
```

코딩 힌트

시퀀스 자료형에서 사용되는 다양한 함수와 리스트 객체에서 제공되는 메소드를 이용합니다.

Section 4 튜플

- 시퀀스 자료형인 튜플은 생성된 후에 변경이 불가능하다는 점을 제외하고는 리스트와 동일



1. 튜플 생성

- 튜플 생성
  - ()를 사용하여 직접 생성
  - list() 함수를 사용하여 다른 자료형의 데이터를 튜플로 생성

2. 튜플 메소드

- 튜플 메소드
  - count(), index()

실습 4

다음과 같은 두 개의 튜플을 생성하고, 튜플로부터 하나의 리스트를 생성하는 프로그램을 작성하시오.

```
fruit=('사과','배','파인애플','포도')
price=(5000,7000,4500,6000)
```

실행결과

```
과일 튜플 : ('사과', '배', '파인애플', '포도')
가격 튜플 : (5000, 7000, 4500, 6000)
튜플로부터 생성된 과일+가격 리스트 : ['사과', 5000, '배', 7000, '파인애플', 4500, '포도', 6000]
```

실습 5

다음과 같은 튜플을 생성하고, 각 요소가 튜플에 몇 번 나타났는지를 출력하는 프로그램을 작성하시오.

```
num=(1,4,6,5,4,3,2,0,1,2,4,6,7,9,4,0)
```

실행결과

```
생성된 튜플 : (1, 4, 6, 5, 4, 3, 2, 0, 1, 2, 4, 6, 7, 9, 4, 0)
1 개수 : 2
4 개수 : 4
6 개수 : 2
5 개수 : 1
3 개수 : 1
2 개수 : 2
0 개수 : 2
7 개수 : 1
9 개수 : 1
```

연습문제

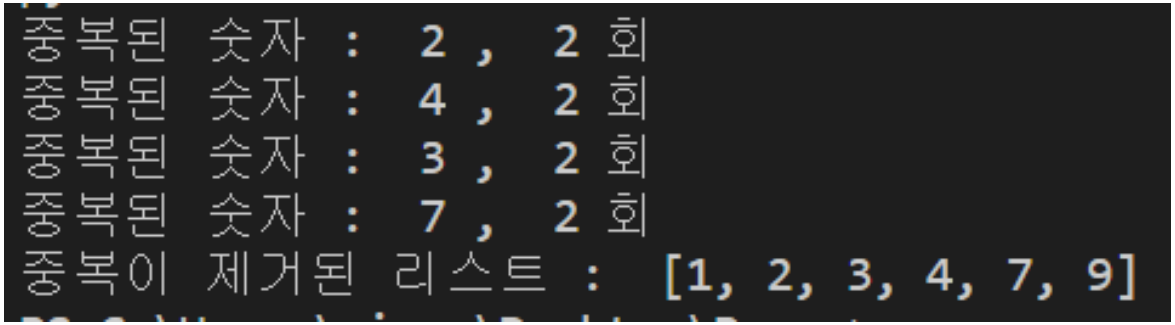
1. 1부터 1000사이의 소수를 구하여 리스트에 저장한 다음 소수와 소수의 개수를 출력하는 프로그램을 작성하시오.

```
1부터 1000사이의 소수의 리스트: [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103,
107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257,
 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431
, 433, 439, 443, 449, 457, 461, 463, 467, 479, 487, 491, 499, 503, 509, 521, 523, 541, 547, 557, 563, 569, 571, 577, 587, 593, 599, 601, 60
7, 613, 617, 619, 631, 641, 643, 647, 653, 659, 661, 673, 677, 683, 691, 701, 709, 719, 727, 733, 739, 743, 751, 757, 761, 769, 773, 787, 7
97, 809, 811, 821, 823, 827, 829, 839, 853, 857, 859, 863, 877, 881, 883, 887, 907, 911, 919, 929, 937, 941, 947, 953, 967, 971, 977, 983,
991, 997]
1부터 1000사이의 소수의 개수: 168
```

연습문제

2. 다음과 같은 튜플에서 중복된 숫자와 중복 횟수를 출력하고, 중복이 제거된 요소를 리스트로 출력하는 프로그램을 작성하시오.

   최초의 튜플 : (1,2,4,4,2,3,7,7,9,3)

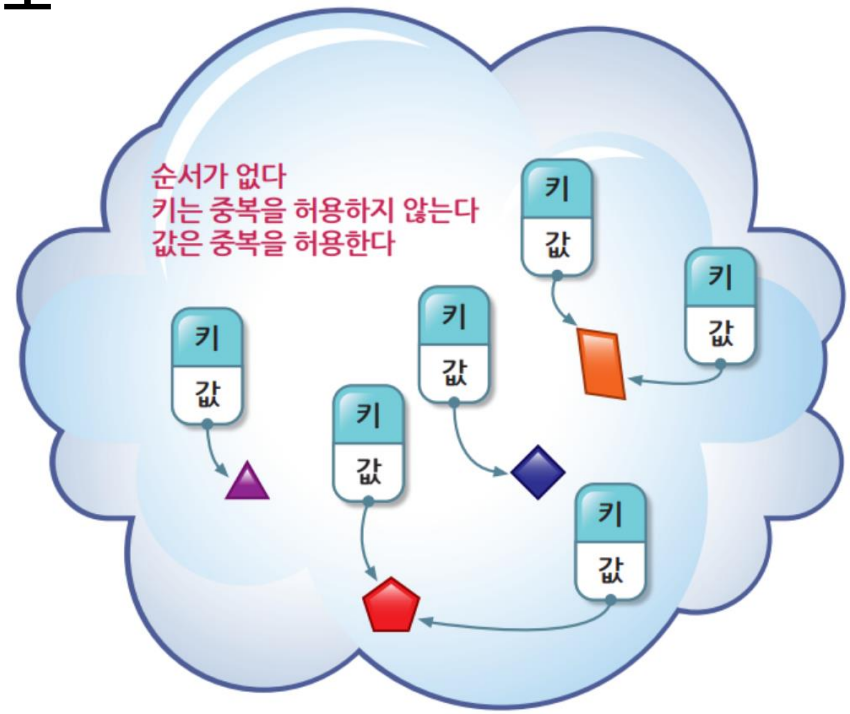


Section 5 집합

- 집합set은 수학에서의 집합 개념과 같으며, 순서가 없으면서 중복을 허용하지 않는 자료구조

  - 주로 멤버 검사와 중복된 항목을 제거할 때 유용

  - 순서가 없기 때문에 인덱스를 통한 접근이 허용되지 않음.



1. 집합의 생성

- 집합 생성
  - 집합기호인 “{}”을 사용하여 생성
  - set() 함수 사용하여 생성(변하지 않는immutable 자료형만 가능)
    - 리스트, 사전 자료형은 불가능

2. 집합 메소드

- 집합은 생성된 후에 변경될 수 있는mutable 자료형

집합 메소드			
add(x)	discard()	issuperset()	union()
clear()	intersection()	pop()	update()
copy()	intersection_update()	remove()	
difference()	isdisjoint()	symmetric_difference()	
difference_update()	issubset()	symmetric_difference_update()	

2. 집합 메소드

- 생성된 집합에 새로운 원소를 추가하고, 기존의 원소를 삭제
- 집합을 복사하기 위해 copy() 메소드를 사용
- 집합 연산과 연관된 다양한 메소드를 제공합니다. 부분집합, 상위집합, 하위집합 등과 연관된 메소드를 제공
- 합집합, 차집합, 교집합 등의 다양한 기능을 제공

실습 1

집합을 이용하여 1부터 100사이의 10개의 랜덤넘버 집합 2개를 생성하고, 두 집합의 합집합, 교집합, 차집합을 구하는 프로그램을 작성하시오.

파이썬에서 난수를 발생시키기 위해서는 랜덤 모듈을 사용해야하며, 랜덤 모듈을 사용하기 위해서 프로그램의 처음에 “import random”을 포함시켜야 합니다. 또한 1부터 100사이의 정수 난수를 구하는 메소드로는 random.randint(1,100)를 사용합니다.

실행결과 (난수에 의한 결과이므로 실행시킬 때마다 결과가 다르게 됩니다.)

```
발생된 10개의 난수 num1 :  {1, 100, 42, 10, 12, 17, 18, 53, 60, 31}
발생된 10개의 난수 num2 :  {32, 14, 15, 83, 61, 21, 56, 60, 29, 31}
num1, num2의 합집합 :  {1, 10, 12, 14, 15, 17, 18, 83, 21, 29, 31, 32, 100, 42, 53, 56, 60, 61}
num1, num2의 교집합 :  {60, 31}
num1, num2의 차집합 :  {1, 100, 42, 10, 12, 17, 18, 53}
```

실습 2

중복을 허용하지 않는 집합을 이용하여 로또넘버(1~45 사이의 정수) 6개를 오름차순으로 정렬하여 출력하는 프로그램을 작성하시오. 난수를 사용합니다.

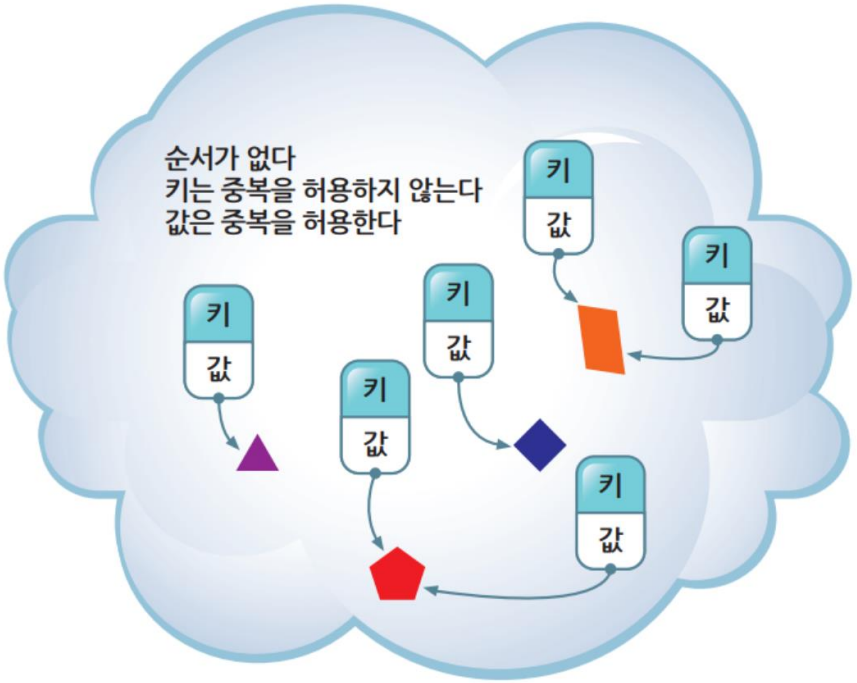
파이썬에서 난수를 발생시키기 위해서는 랜덤 모듈을 사용해야하며, 랜덤 모듈을 사용하기 위해서 프로그램의 처음에 "import random"을 포함시켜야 합니다. 또한 1부터 45사이의 정수 난수를 구하는 메소드로는 random.randint(1,45)를 사용합니다.

실행결과

```
이번주 로또 넘버 :  [7, 12, 27, 32, 34, 40]
중복된 난수의 발생 횟수 :  3
```

Section 6 사전dictionary

- 키key와 값value을 하나의 요소로 묶어 표현한 자료구조로서 사전dictionary을 제공
  - 키는 중복을 허용하지 않고 값은 중복을 허용
  - 사전 역시 집합처럼 순서가 없기 때문에 인덱스를 통한 접근 불가능



[그림 6-2] 사전 자료구조

1. 사전의 생성과 사용

- 사전 생성
  - "{}" 기호를 사용하여 생성. 각 요소에서 키와 값은 ":"으로 구분
  - dict() 함수를 사용하여 사전 자료형을 생성

```
>>> sungjuk={2195111:98, 2195113:88, 2195098:100}        # 3개의 요소를 가진 사전 생성
>>> sungjuk
{2195111: 98, 2195113: 88, 2195098: 100}
>>> type(sungjuk)                                       # 타입은?
<class 'dict'>
>>> len(sungjuk)                                        # 길이는?
3
>>>
>>> num={1:"a",2:"b",1:"c",2:"d"}                       # 중복된 킷값으로 생성
>>> num                                                 # 중복될 수 없으므로 뒤에 입력된 값으로 생성
{1: 'c', 2: 'd'}
>>>
>>> empty=dict()                                        # dict() 함수를 사용하여 빈 사전 생성
>>> empty
{}
```

1. 사전의 생성과 사용

- 두 개의 리스트를 하나의 사전으로 만들기 위해 zip() 함수를 이용
- 사전에 있는 요소를 검색하고, 새로운 요소를 추가, 삭제
- “in” 연산자와 함수를 사용
  - 단 사전에서는 모든 함수들이 킷값에 적용

2. 사전 메소드

- 사전은 생성된 후에 변경될 수 있는mutable 자료형

사전 메소드			
clear()	get()	pop()	update()
copy()	items()	popitem()	values()
fromkeys()	keys()	setdefault()	

2. 사전 메소드

- 생성된 사전에 값을 검색하고, 사전의 내용을 다른 사전으로 복사하고, 사전의 요소들을 삭제
- 사전에 값을 추가하고, 삭제하고, 두 개의 사전을 합친다
  - setdefault(), pop(), popitem(), update()
- for 반복문에서 효율적으로 사용하는 메소드
  - keys(), values(), items()

실습 1

5명 학생의 학번과 전화번호를 사전 자료구조에 저장하고, 학번을 입력받아 해당학생의 전화번호를 출력하는 프로그램을 작성하시오. 학생의 학번이 없는 경우 학생이 없다는 메시지를 출력하시오.

실행결과

```
1번째 학생 학번 입력 : 9095001
1번째 학생 전화번호 : 010-9999-9999
2번째 학생 학번 입력 : 2095031
2번째 학생 전화번호 : 010-8888-8888
3번째 학생 학번 입력 : 2098004
3번째 학생 전화번호 : 010-7777-7777
4번째 학생 학번 입력 : 2099089
4번째 학생 전화번호 : 010-6666-6666
5번째 학생 학번 입력 : 2089001
5번째 학생 전화번호 : 010-5555-5555
학생 전화번호부가 완성되었습니다
검색을 원하는 학생의 학번 입력 : 2095031
입력한 학생의 전화번호 : 010-8888-8888
```

연습문제

1. 1부터 100사이의 정수 난수 10개를 발생시켜 중복을 허용하지 않은 집합에 저장하고 집합에서 가장 큰 수와 작은 수를 구하는 프로그램을 작성하시오. 단 파이썬 max(), min() 함수를 사용하지 말고 작성하시오.

```
생성된 집합 :  {32, 97, 98, 7, 76, 79, 18, 87, 26, 31}
집합에서 가장 큰 수 :  98
집합에서 가장 작은 수 :  7
```

2. 사용자로부터 하나의 자연수를 입력받아 1부터 입력된 자연수까지의 2,3,7 각각의 배수를 집합으로 출력하고 2,3,7 모두의 배수를 집합으로 출력하는 프로그램을 작성하시오.

```
숫자를 입력하시오:100
2의 배수: [2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66,
68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100]
3의 배수: [3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60, 63, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 96, 99]
7의 배수: [7, 14, 21, 28, 35, 42, 49, 56, 63, 70, 77, 84, 91, 98]
2,3,7의 배수: [42, 84]
```