Lec_03

Variables 2

Loop Syntax 1

Variables 2

- List
  - 대괄호([]) 사용
  - 다양한 자료형을 항목으로 가질 수 있음
  - 각 항목이 다른 자료형이라도 가능

```
lst = ['Song', 'Nam', 'Jung', 'Kim', 'Park']
lst_mix = [92, '파이썬', True, lst]
lst_num = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

print(type(lst_mix))
```

Variables 2

- List

```
lst_merge = lst + lst_mix    # 결합
print(lst_merge)

print(lst_num[2:8:2])        # 슬라이싱 [시작:끝:간격]

del lst_num[5]               # 항목 삭제
print(lst_num)

print(70 in lst_num)         # 항목 존재 여부
print(110 in lst_num)
```

Variables 2

- List
  - method : List 데이터형에서 사용할 수 있는 함수

```
lst_num.append(110)                 # 마지막에 항목 추가
lst_num.insert(6, '칠십오')          # 특정 위치에 항목 추가
lst_num.remove(80)                  # 입력값과 일치하는 항목 삭제
lst_num.pop()                       # 마지막 항목을 반환하고 삭제
lst_num.index(30)                   # 입력된 인자와 일치하는 항목의 위치 반환
lst_num.count(50)                   # 입력된 인자와 일치하는 항목의 개수 반환
lst_num.sort()                      # 리스트 항목을 오름차순으로 정렬
lst_num.reverse()                   # 리스트 항목을 내림차순으로 정렬
```

Variables 2

- Tuple
  - 소괄호(()) 사용 혹은 사용하지 않고 항목 입력
  - 다양한 자료형을 항목으로 가질 수 있음
  - 각 항목이 다른 자료형이라도 가능
  - **단, 한 번 입력하면 항목을 변경할 수 없음!!**

```
tup1 = 10, 20, '삼십', False, lst
tup2 = (lst, '김', '이', '박')
print(tup1,  " >>> type : ", type(tup1))
print("{} >>> type : {}".format(tup2, type(tup2)))
```

Variables 2

- Tuple

```
tup3 = (100, )                  # 항목이 하나인 튜플을 생성할 때는 항목 뒤 ,
tup4 = 200,

tup1[1] = 30                    # 에러 발생. 생성된 튜플은 항목 변경 불가!!
del tup1[2]                     # 에러 발생. 생성된 튜플은 항목 삭제 불가!!

tup2.index('이')                # 입력된 인자와 일치하는 항목의 위치 반환
tup2.count('김')                # 입력된 인자와 일치하는 항목의 개수 반환
```

Variables 2

- Set
  - 중괄호({}) 사용
  - 다양한 자료형을 항목으로 가질 수 있음
  - 각 항목이 다른 자료형이라도 가능
  - **항목의 순서가 없고, 중복해서 쓸 수 없음**

```
set1 = {1, 2, '삼', 4, '오', 1}
print(set1)
print(type(set1))
```

Variables 2

- Set

```
set2 = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
set3 = {3, 5, 7, 9, 11, 13, 15}

set2.intersection(set3)                    # 교집합
set2.union(set3)                           # 합집합
set2.difference(set3)                      # 차집합
```

Variables 2

- List - Tuple - Set 간 타입 변환

```
a = [1, 2, 3, 4, 5, 5, 3, 2]

print(a, type(a), sep=' >> type : ')

b = tuple(a)
print(b, type(b), sep=' >> type : ')

c = set(b)
print(c, type(c), sep=' >> type : ')

d = list(c)
print(d, type(d), sep=' >> type : ')
```

Loop Syntax 1

- While 문
  - **프로그램을 지속적으로 On 시켜놓기 위해 필수적인 문법**
  - 특정 조건이 만족되는 동안 코드 블록을 반복 실행

```
# while <조건문>:
#     <코드블록>
i = 0

while i < 10:
    i = i + 1
    print('{}번째 입니다.'.format(i))
```

Loop Syntax 1

- 무한반복문, continue, break

```
i = 0

while True:                 # True 혹은 1을 조건으로 하면 무한반복
    i += 1
    if i % 3 = 0:
        continue            # continue를 만나면 아래 코드는 실행 하지 않고 다시 반복
    print(i)
    if i % 10 = 0:
        break               # break를 만나면 반복문을 종료함

print('반복문 종료!')
```

Loop Syntax 1

- 연습문제

```
# 1 ~ 100 까지의 숫자를 더한 값을 구하시오.
i = 0
sum = 0

while i < 100
...

# 1 ~ 200 까지의 짝수를 더한 값을 구하시오.
```

[MiniProject] 메뉴판 업그레이드 1

- While문 활용 무한반복
- 메뉴 선택 화면에서 0을 입력하면 프로그램 종료
- 메뉴를 여러 가지 선택할 수 있도록
- 메뉴는 리스트에 하나씩 추가하여 저장
- 마지막 화면에서는 리스트에 저장된 메뉴가 전부 출력
- 단, for문을 배우지 않았으므로 리스트 형태 그대로 출력함