

6주차 실습

7장(리스트, 튜플, 딕셔너리) 실습

2019. 10. 8.

오늘의 평가실습 (7장 실습)

- 다음을 Python으로 코딩하고, 소스코드와 실행결과의 capture화면을 제출하라.
 - 7장
 - 183페이지 SELF STUDY 7-1
 - 185페이지 SELF STUDY 7-2
 - 193페이지 SELF STUDY 7-3
 - 198페이지 SELF STUDY 7-4
 - 205페이지 SELF STUDY 7-5
- 제출 : 10. 14.(월) 자정까지 (**지각 제출 없음**)
- 제출물 : 아래아한글 파일 (또는 pdf 파일) 보고서 1개로 제출
 - 소스코드는 실행해 볼 수 있도록 표안에 붙여넣기로 제출
 - 실행결과는 소스코드와 실행결과를 볼 수 있게 화면을 capture하여 제출

183페이지 SELF STUDY 7-1

- 문제 :
 - 값을 4개가 아닌 10개를 입력 받아 합계를 출력하도록 Code07-03.py를 수정해 보자.
 - 또 합계를 구하는 마지막 for 문 대신 while문을 사용해 보자

Code07-03.py

```
aa = []
for i in range(0, 4) :
    aa.append(0)
hap = 0

for i in range(0, 4) :
    aa[i] = int(input(str(i+1) + "번째 숫자 : " ))

hap = aa[0] + aa[1] + aa[2] + aa[3]

print("합계 ==> %d" % hap)
```

185페이지 SELF STUDY 7-2

- 문제 :

- Code07-04.py를 리스트 aa에 3의 배수를 200개 입력하도록 수정해 보자.
- 그리고 리스트 bb에는 리스트 aa의 역순으로 입력해보자.
- 최종적으로 bb[0] 과 bb[199]의 값을 출력하면 다음과 같다.

Code07-04.py

```
aa = []
bb = []
value = 0

for i in range(0, 100) :
    aa.append(value)
    value += 2

for i in range(0, 100) :
    bb.append(aa[99 - i])

print("bb[0]에는 %d이, bb[99]에는 %d이 입력
됩니다." % (bb[0], bb[99]))
```

```
===== RESTART: C:\Users\T2\AppData\Local\Pro
bb[0]에는 5970이, bb[199]에는 0이 입력됩니다.
>>> |
```

193페이지 SELF STUDY 7-3

- 문제 :
 - 4행 5열의 2차원 리스트를 만들고, 0부터 3의 배수를 입력하고 출력하도록 Code07-06.py를 수정해보자.
 - 출력결과는 다음과 같다.

```
===== RESTART: C:\#Us
  0    3    6    9   12
 15   18   21   24   27
 30   33   36   39   42
 45   48   51   54   57
>>> |
```

Code07-06.py

```
list1 = []
list2 = []
value=1
for i in range(0, 3) :
    for k in range(0, 4) :
        list1.append(value)
        value += 1
    list2.append(list1)
    list1 = []

for i in range(0, 3) :
    for k in range(0, 4) :
        print("%3d" % list2[i][k], end = " ")
    print("")
```

198페이지 SELF STUDY 7-4

- 문제 :

- 다음과 같이 2차원 튜플을 생성한 후 모든 값을 출력해보자.

```
tt = ((1,2,3),  
      (4,5,6),  
      (7,8,9))
```

```
===== RE  
1  2  3  
4  5  6  
7  8  9  
>>> |
```

- 힌트 :

- for 루프를 이용할 것.
- 출력에서 다음의 차이를 이용할 것.
- `print(tt[i][j], end=' ')`
- `print(tt[i][j], end='\n')`

205페이지 SELF STUDY 7-5

- 문제 :

- Code07-10.py를 수정해 동물의 어미와 새끼의 이름을 딴 셔너리로 묶는 프로그램을 작성해 보자.

- 힌트)

- 동물의 어미와 새끼의 이름은 인터넷에서 검색한다.

```
===== RESTART: C:\Users#T2\AppData\Local\Programs\Python\Python37#1.py =====
[<메리다>, '개', '곰', '고등어', '명태', '말', '호랑이'] 중 새끼이름을 알고 싶은 동물
? 메리다의 새끼는 <늑소니>입니다.
[<메리다>, '개', '곰', '고등어', '명태', '말', '호랑이'] 중 새끼이름을 알고 싶은 동물
? 호랑이의 새끼는 <개호주>입니다.
[<메리다>, '개', '곰', '고등어', '명태', '말', '호랑이'] 중 새끼이름을 알고 싶은 동물
? 명태의 새끼는 <노가리>입니다.
[<메리다>, '개', '곰', '고등어', '명태', '말', '호랑이'] 중 새끼이름을 알고 싶은 동물
? 곰의 새끼는 <소>입니다.
[<메리다>, '개', '곰', '고등어', '명태', '말', '호랑이'] 중 새끼이름을 알고 싶은 동물
? 개는 동물이 없습니다. 확인해 보세요.
[<메리다>, '개', '곰', '고등어', '명태', '말', '호랑이'] 중 새끼이름을 알고 싶은 동물
? 메리다의 새끼는 <늑소니>입니다.
```

Code07-10.py

```
## 변수 선언 부분 ##
```

```
foods = {"떡볶이": "오뎅",
          "짜장면": "단무지",
          "라면": "김치",
          "피자": "피클",
          "맥주": "땅콩",
          "치킨": "치킨무",
          "삼겹살": "상추" };
```

```
## 메인 코드 부분 ##
```

```
while (True) :
    myfood = input(str(list(foods.keys())) + " 중 좋아하는 음식은?")
    if myfood in foods :
        print("<%s> 궁합 음식은 <%s>입니다." % (myfood,
        foods.get(myfood)))
    elif myfood == "끝" :
        break
    else :
        print("그런 음식이 없습니다. 확인해 보세요.")
```