평가실습 #3 (5주차 실습)

5장(조건문), 6장(반복문) 실습

2019. 10. 01.

오늘의 평가실습 (5장, 6장 실습)

- 다음을 Python으로 코딩하고, 소스코드와 실행결과의 capture화면을 제출하라.
  - 5장
    1. 127페이지 SELF STUDY 5-1
    2. 132페이지 SELF STUDY 5-2
  - 6장
    3. 149페이지 SELF STUDY 6-1
    4. 155페이지 SELF STUDY 6-2
    5. 156페이지 SELF STUDY 6-3
    6. 159페이지 SELF STUDY 6-4
    7. 163페이지 SELF STUDY 6-5
    8. 164페이지 SELF STUDY 6-6
    9. 167페이지 SELF STUDY 6-7
- 제출 : 10. 07.(월) 자정까지 (Late Penalty 있음, 1주 단위로 20%)
- 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

127쪽 Self Study 5-1

- 문제 : Code05-08.py를 다음 조건처럼 좀 더 세분화해 보자
  - 95점 이상 : A+ ,
  - 90점 이상 : A0,
  - 85점 이상 : B+,
  - 80점 이상 : B0,
  - 75점 이상 : C+,
  - 70점 이상 : C0,
  - 65점 이상 : D+,
  - 60점 이상 : D0,
  - 60점 미만 : F

Code05-08.py

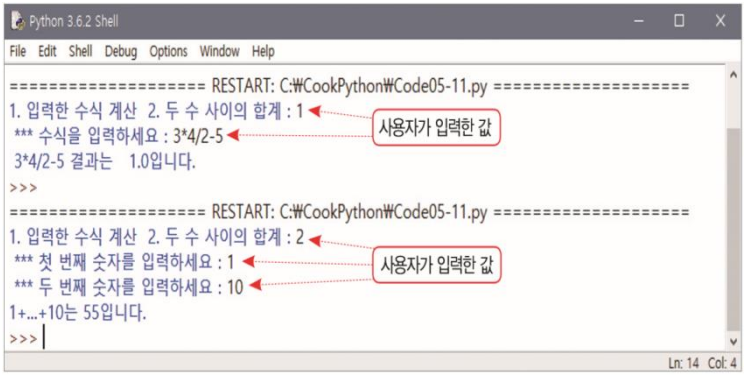
```
score = int(input("점수를 입력하세요 : "))

if score >= 90 :
        print("A")
else :
        if score >= 80 :
                print("B")
        else :
                if score >= 70 :
                        print("C")
                else :
                        if score >= 60 :
                                print("D")
                        else :
                                print("F")

print("학점입니다. ^^")
```

132쪽 Self Study 5-2

- 문제 : Code05-11.py의 두번째 기능처럼 두 숫자를 입력받고 두 숫자 사이의 합계를 구하는 프로그램을 만들어보자.
- 단, 1씩 증가하지 않고 증가하는 숫자도 입력받는다.
- 예를 들어 1, 100, 3을 입력하면 1 + 4 + ... + 100의 합계를 구한다.



Code05-11.py

```
## 변수 선언 부분 ##
select, answer, numStr, num1, num2 = 0, 0, "", 0, 0

## 메인 코드 부분 ##
select = int(input("1. 입력한 수식 계산  2. 두 수 사이의 합계 : "))

if  select == 1 :
    numStr = input(" *** 수식을 입력하세요 : ")
    answer = eval(numStr)
    print(" %s 결과는 %5.1f입니다. " %(numStr, answer))
elif select == 2 :
    num1 = int(input(" *** 첫 번째 숫자를 입력하세요 : "))
    num2 = int(input(" *** 두 번째 숫자를 입력하세요 : "))
    for i in range(num1, num2+1) :
        answer = answer + i
    print("%d+...+%d는 %d입니다. " %(num1, num2, answer))
else :
    print("1 또는 2만 입력해야 합니다.")
```

149페이지 SELF STUDY 6-1

- 문제 : Code06-03.py를 0과 100 사이에 있는 7의 배수 합계를 구하도록 수정해보자

Code06-03.py

```
i, hap = 0, 0

for i in range(501, 1001, 2) :
    hap = hap + i

print("500과 1000 사이에 있는 홀수의 합계 : %d" % hap)
```

155페이지 SELF STUDY 6-2

- 문제 : Code06-07.py를 각 단의 제목이 출력되도록 수정해 보자.

```
====== RESTAR
##  2 단 ##
2 X 1 =  2
2 X 2 =  4
2 X 3 =  6
2 X 4 =  8
2 X 5 = 10
2 X 6 = 12
2 X 7 = 14
2 X 8 = 16
2 X 9 = 18

##  3 단 ##
3 X 1 =  3
3 X 2 =  6
3 X 3 =  9
3 X 4 = 12
3 X 5 = 15
3 X 6 = 18
3 X 7 = 21
3 X 8 = 24
3 X 9 = 27
```

Code06-07.py

```
i, k = 0, 0

for i in range(2, 10, 1) :
    for k in range(1, 10, 1) :
        print("%d X %d = %2d" %(i, k, i * k))
    print("")
```

156페이지 SELF STUDY 6-3

- 문제 : Code06-08.py를 거꾸로 출력되도록 수정해보자

```
>>>
====== RESTART: C:₩Users₩T2₩AppData₩Local₩Programs₩Python₩Python37₩1.py ======
 #  2단   # #  3단   # #  4단   # #  5단   # #  6단   # #  7단   # #  8단   # #  9단   #
 2X  1=  2 3X  1=  3 4X  1=  4 5X  1=  5 6X  1=  6 7X  1=  7 8X  1=  8 9X  1=  9
 2X  2=  4 3X  2=  6 4X  2=  8 5X  2= 10 6X  2= 12 7X  2= 14 8X  2= 16 9X  2= 18
 2X  3=  6 3X  3=  9 4X  3= 12 5X  3= 15 6X  3= 18 7X  3= 21 8X  3= 24 9X  3= 27
 2X  4=  8 3X  4= 12 4X  4= 16 5X  4= 20 6X  4= 24 7X  4= 28 8X  4= 32 9X  4= 36
 2X  5= 10 3X  5= 15 4X  5= 20 5X  5= 25 6X  5= 30 7X  5= 35 8X  5= 40 9X  5= 45
 2X  6= 12 3X  6= 18 4X  6= 24 5X  6= 30 6X  6= 36 7X  6= 42 8X  6= 48 9X  6= 54
 2X  7= 14 3X  7= 21 4X  7= 28 5X  7= 35 6X  7= 42 7X  7= 49 8X  7= 56 9X  7= 63
 2X  8= 16 3X  8= 24 4X  8= 32 5X  8= 40 6X  8= 48 7X  8= 56 8X  8= 64 9X  8= 72
 2X  9= 18 3X  9= 27 4X  9= 36 5X  9= 45 6X  9= 54 7X  9= 63 8X  9= 72 9X  9= 81
>>>
```

```
 #  9단   # #  8단   # #  7단   # #  6단   # #  5단   # #  4단   # #  3단   # #  2단   #
 9X  9= 81 8X  9= 72 7X  9= 63 6X  9= 54 5X  9= 45 4X  9= 36 3X  9= 27 2X  9= 18
 9X  8= 72 8X  8= 64 7X  8= 56 6X  8= 48 5X  8= 40 4X  8= 32 3X  8= 24 2X  8= 16
 9X  7= 63 8X  7= 56 7X  7= 49 6X  7= 42 5X  7= 35 4X  7= 28 3X  7= 21 2X  7= 14
 9X  6= 54 8X  6= 48 7X  6= 42 6X  6= 36 5X  6= 30 4X  6= 24 3X  6= 18 2X  6= 12
 9X  5= 45 8X  5= 40 7X  5= 35 6X  5= 30 5X  5= 25 4X  5= 20 3X  5= 15 2X  5= 10
 9X  4= 36 8X  4= 32 7X  4= 28 6X  4= 24 5X  4= 20 4X  4= 16 3X  4= 12 2X  4=  8
 9X  3= 27 8X  3= 24 7X  3= 21 6X  3= 18 5X  3= 15 4X  3= 12 3X  3=  9 2X  3=  6
 9X  2= 18 8X  2= 16 7X  2= 14 6X  2= 12 5X  2= 10 4X  2=  8 3X  2=  6 2X  2=  4
 9X  1=  9 8X  1=  8 7X  1=  7 6X  1=  6 5X  1=  5 4X  1=  4 3X  1=  3 2X  1=  2
>>>
```

Code06-08.py

```
## 전역 변수 선언 부분 ##
i, k, guguLine = 0, 0, ""

## 메인 코드 부분 ##
for i in range(2, 10) :
    guguLine = guguLine + (" #  %d단  #" % i)

print(guguLine)

for i in range(1, 10) :
    guguLine = ""
    for k in range(2, 10) :
        guguLine = guguLine + str("%2dX %2d= %2d" % (k, i, k*i))
    print(guguLine)
```

159페이지 SELF STUDY 6-4

- 문제 : Code06-05.py를 while문으로 수정해 보자
  - Code06-09.py를 참고한다

Code06-05.py

```
i, hap = 0, 0
num1, num2, num3 = 0, 0, 0

num1 = int(input("시작값을 입력하세요 : "))
num2 = int(input("끝값을 입력하세요 : "))
num3 = int(input("증가값을 입력하세요 : "))

for i in range(num1, num2+1, num3) :
    hap = hap + i

print("%d에서 %d까지 %d씩 증가시킨 값의 합계 : %d" %
(num1, num2, num3, hap))
```

Code06-09.py

```
i, hap = 0, 0

i = 1
while i < 11 :
    hap = hap + i
    i = i + 1

print("1에서 10까지의 합계 : %d" % hap)
```

163페이지 SELF STUDY 6-5

- 문제 : '$'를 입력하면 while문을 빠져나가도록 Code06-12.py를 수정해보자

Code06-12.py

```
hap = 0
a, b = 0, 0

while True :
    a = int(input("더할 첫 번째 수를 입력하세요 : "))
    if a == 0 :
        break
    b = int(input("더할 두 번째 수를 입력하세요 : "))
    hap = a + b
    print("%d + %d = %d" % (a, b, hap))

print("0을 입력해 반복문을 탈출했습니다")
```

164페이지 SELF STUDY 6-6

- 문제 : Code06-13.py를 while문으로 변경해보자.
- 출력결과는 동일하다.

Code06-13.py

```
hap, i = 0,0

for i in range(1,101) :
    hap += i

    if hap >= 1000 :
        break

print("1~100의 합계를 최초로 1000이 넘게 하는 숫자 : %d" % i)
```

167페이지 SELF STUDY 6-7

- 문제 : Code06-15.py를 모두 for 문으로 변경해보자.
- 출력은 하트 모양을 사용하는데, 하트모양의 유니코드는 16진수 '2665'이다.

Code06-15.py

```
## 전역 변수 선언 부분 ##
i, k = 0, 0

## 메인 코드 부분 ##
i = 0
while i < 9 :
    if i<5 :
        k = 0
        while k < 4-i :
            print('  ', end = '')
            k += 1
        k = 0
        while k < i*2+1 :
            print('₩u2605',  end = '')
            k += 1
    else :
        k = 0
        while k < i-4 :
            print('  ', end = '')
            k += 1
        k = 0
        while k < (9-i)*2-1 :
            print('₩u2605',  end = '')
            k += 1
    print()
    i += 1
```