

데이터 처리를 위한 Python 프로그래밍 입문

8-2강. 사용자 정의 함수

ERICA 2018-2

강의 내용

- ▶ 사용자 정의 함수
- ▶ 사용자 정의 함수 응용

사용자 정의 함수

- ▶ 사용자 정의 함수
 - ▶ 내장 함수가 아닌 사용자 요구사항을 충족하는 함수 정의
 - ▶ 한 프로그램 내에서 특정한 작업이 여러 번 나오는 경우가 존재
 - ▶ 같은 내용의 코드를 다시 작성해야 하는 번거러움 해소

사용자 정의 함수 응용(1/10)

- ▶ 사용자 함수 : 점수를 입력받아 학점을 계산
 - ▶ 상황 : 국어, 수학, 영어 점수를 각각 정수형으로 입력받고
각각의 점수를 학점으로 계산하여
결과 화면과 같이 출력하는 프로그램
 - ▶ 결과 화면

```

국어 점수를 입력하세요 : 91
수학 점수를 입력하세요 : 89
영어 점수를 입력하세요 : 79
국어 학점은 A 입니다.
수학 학점은 B 입니다.
영어 학점은 C 입니다.
>>>
\\

```

사용자 정의 함수 응용(2/10)

- ▶ 사용자 함수 : 점수를 입력받아 학점을 계산(계속)
 - ▶ 해결 코드(함수 사용 없음)

1

```
kor_score = int(input('국어 점수를 입력하세요 : '))  
math_score = int(input('수학 점수를 입력하세요 : '))  
eng_score = int(input('영어 점수를 입력하세요 : '))
```

2

```
#국어 점수 학점 계산  
if kor_score >= 90:  
    kor_grade = 'A'  
elif kor_score >= 80:  
    kor_grade = 'B'  
elif kor_score >= 70:  
    kor_grade = 'C'  
elif kor_score >= 60:  
    kor_grade = 'D'  
else:  
    kor_grade = 'F'
```

3

```
#수학 점수 학점 계산  
if math_score >= 90:  
    math_grade = 'A'  
elif math_score >= 80:  
    math_grade = 'B'  
elif math_score >= 70:  
    math_grade = 'C'  
elif math_score >= 60:  
    math_grade = 'D'  
else:  
    math_grade = 'F'
```

4

```
#영어 점수 학점 계산
if eng_score >=90:
    eng_grade='A'
elif eng_score >= 80:
    eng_grade='B'
elif eng_score >= 70:
    eng_grade='C'
elif eng_score >= 60:
    eng_grade='D'
else:
    eng_grade='F'
```

5

```
print('국어 학점은', kor_grade, '입니다.')
print('수학 학점은', math_grade, '입니다.')
print('영어 학점은', eng_grade, '입니다.')
```

사용자 정의 함수 응용(3/10)

▶ 사용자 함수 : 점수를 입력받아 학점을 계산(계속)

▶ 해결 코드(함수 사용)

```
def calgrade(score):  
    if score >= 90:  
        grade = 'A'  
    elif score >= 80:  
        grade = 'B'  
    elif score >= 70:  
        grade = 'C'  
    elif score >= 60:  
        grade = 'D'  
    else:  
        grade = 'F'  
    return grade
```

```
kor_score = int(input('국어 점수를 입력하세요 :'))  
math_score = int(input('수학 점수를 입력하세요 :'))  
eng_score = int(input('영어 점수를 입력하세요 :'))
```

```
kor_grade = calgrade(kor_score)  
math_grade = calgrade(math_score)  
eng_grade = calgrade(eng_score)
```

```
print('국어 학점은', kor_grade, '입니다.')  
print('수학 학점은', math_grade, '입니다.')  
print('영어 학점은', eng_grade, '입니다.')
```

사용자 정의 함수 응용(4/10)

- ▶ 사용자 함수 정의 : 합격 여부

- ▶ 상황 : int형 점수를 list로 입력받음

점수의 평균을 계산하고,

평균이 60점 이상이면 합격, 미만은 실패로 판단

결과 화면과 같이 출력하는 프로그램

* 두 개 함수 정의 : `def calavg(numlist), def passFail(numlist)`

- ▶ 결과 화면

```
>>> passFail([60,70,90,60])  
'Pass'  
>>> passFail([60,70,90,60,0])  
'Fail'  
>>>
```


사용자 정의 함수 응용(5/10)

▶ 상용자 함수 정의 : 합격 여부(계속)

▶ 해결 코드

```
def calavg(numlist):  
    sum=0  
    avg=0  
    for number in numlist:  
        sum += number  
    avg = sum/len(numlist)  
    return avg  
  
def passFail(numlist):  
    if calavg(numlist) >= 60:  
        message = "Pass"  
    else:  
        message = "Fail"  
    return message
```

사용자 정의 함수 응용(6/10)

▶ 상용자 함수 정의 : 최고 점수 찾기

- ▶ 상황 : 여러 개의 점수를 list로 입력받아, 가장 높은 점수를 구하는 함수로 정의하고, 호출하여 결과 화면과 같이 출력하는 프로그램

▶ 결과 화면

```
>>> max_score([81,91,93,88,89,94,81,90,79,76,83,89,88,96,77,89,76,79])
96
>>> math_score=[81,91,93,88,89,94,81,90,79,76,83,89,88,96,77,89,76,79]
>>> max_score(math_score)
96
>>>
```

▶ 해결 코드

```
def max_score(score_list):
    max_num = 0
    for score in score_list:
        if score > max_num:
            max_num = score
    return max_num
```

사용자 정의 함수 응용(7/10)

▶ 상용자 함수 정의 : 계산기

▶ 상황 : 연산(덧셈, 뺄셈, 곱셈, 나눗셈)을 선택

두 개 정수를 입력받음

연산결과를 결과 화면과 같이 출력하는 프로그램

* 종료를 선택할 때 까지 계속 연산선택

▶ 결과 화면

```
#### 계산기 ####
1.덧셈 2.뺄셈 3.곱셈 4.나눗셈 5.종료 : 1
정수 2개를 입력하세요.
정수1 : 4
정수2 : 6
계산 결과 : 10
1.덧셈 2.뺄셈 3.곱셈 4.나눗셈 5.종료 : 2
정수 2개를 입력하세요.
정수1 : 4
정수2 : 6
계산 결과 : 2
```

```
계산기
1.덧셈 2.뺄셈 3.곱셈 4.나눗셈 5.종료 : 3
정수 2개를 입력하세요.
정수1 : 4
정수2 : 6
계산 결과 : 24
1.덧셈 2.뺄셈 3.곱셈 4.나눗셈 5.종료 : 4
정수 2개를 입력하세요.
정수1 : 4
정수2 : 6
계산 결과 : 1.5
1.덧셈 2.뺄셈 3.곱셈 4.나눗셈 5.종료 : 5
>>>
```

사용자 정의 함수 응용(8/10)

▶ 상용자 함수 정의 : 계산기(계속)

▶ 해결 코드

```
def add(n1, n2):  
    result = n1 + n2  
    return result  
def sub(n1, n2):  
    if n1 >= n2:  
        result = n1 - n2  
    else:  
        result = n2 - n1  
    return result  
def mul(n1, n2):  
    result = n1 * n2  
    return result  
def div(n1, n2):  
    if n1 >= n2:  
        result = n1 / n2  
    else:  
        result = n2 / n1  
    return result
```

```
print('#### 계산기 ####')  
num_result = 0  
while(True):  
    num = int(input('1.덧셈    2.뺄셈    3.곱셈    4.나눗셈    5.종료 : '))  
    if num == 5:  
        break  
    else:  
        print('정수 2개를 입력하세요.')        n1 = int(input('정수1 : '))  
        n2 = int(input('정수2 : '))  
        if num == 1:  
            num_result = add(n1, n2)  
        elif num == 2:  
            num_result = sub(n1, n2)  
        elif num == 3:  
            num_result = mul(n1, n2)  
        else:  
            num_result = div(n1, n2)  
        print('계산 결과 : ', num_result)
```

사용자 정의 함수 응용(9/10)

- ▶ 상용자 함수 정의 : TV 채널, 음량 조절
 - ▶ 상황 : TV 기능(채널, 음량)을 선택 후, 변경할 만큼의 수를 입력 받음
연산결과를 결과 화면과 같이 출력하는 프로그램
 - * 종료를 선택할 때 까지 계속 연산선택
 - * 초기 채널은 5, 초기 볼륨은 4

▶ 결과 화면

```

1.채널 2.음량 3.종료 : 1
바꾸고 싶은 만큼 입력 : 3
현재 채널 : 8
1.채널 2.음량 3.종료 : 1
바꾸고 싶은 만큼 입력 : 6
현재 채널 : 14
1.채널 2.음량 3.종료 : 2
바꾸고 싶은 만큼 입력 : 9
현재 음량 : 13
1.채널 2.음량 3.종료 : 2
바꾸고 싶은 만큼 입력 : -2
현재 음량 : 11
1.채널 2.음량 3.종료 : 3
>>>
  
```

사용자 정의 함수 응용(10/10)

▶ 상용자 함수 정의 : TV 채널, 음량 조절(계속)

▶ 해결 코드

```
def channel(now_cha, num):  
    now_cha = now_cha+num  
    return now_cha  
def volumn(now_vol, num):  
    now_vol = now_vol+num  
    return now_vol  
  
now_cha = 5  
now_vol = 4  
while(True):  
    what = int(input('1.채널   2.음량   3.종료   : '))  
    if what == 3:  
        break  
    else:  
        if what ==1:  
            num = int(input('바꾸고 싶은 만큼 입력 : '))  
            now_cha = channel(now_cha, num)  
            print('현재 채널 : ', now_cha)  
        elif what ==2:  
            num = int(input('바꾸고 싶은 만큼 입력 : '))  
            now_vol = volumn(now_vol, num)  
            print('현재 음량 : ', now_vol)  
        else:  
            print('잘못 입력하였습니다. 다시 입력해주세요.')
```

Thank you