

데이터 처리를 위한 Python 프로그래밍 입문

7강. 조건문반복문

ERICA 2018-2

강의 내용

- ▶ 조건문과 반복문 혼용
- ▶ 조건문과 반복문 혼용 활용

조건문반복문 혼용

- ▶ 반복문과 조건문 혼용
 - ▶ 조건에 해당하는 경우에만 반복문을 수행
 - ▶ 반복문 내에서 조건의 결과에 따라 실행문을 수행

표기	내 용
<pre>for variable in range(num) : if조건 : True_statements</pre>	range 범위내의 variable 각각에 대해서 조건문의 조건을 확인한 후, 조건이 참일 경우 True_statements를 수행

조건문반복문 혼용 활용(1/12)

- ▶ 반복문과 조건문 : 5의 배수
 - ▶ 상황 : 40미만의 숫자 중 5의 배수만을 출력하는 프로그램(나머지)
 - * for문과 if문을 사용

- ▶ 결과 화면

```
>>> for number in range(40):  
        if number %5 == 0:  
            print(number)
```

```
0  
5  
10  
15  
20  
25  
30  
35  
>>>
```

조건문반복문 혼용 활용(2/12)

- ▶ 반복문과 조건문 : 범위내 정수 중 짝수 출력
 - ▶ 상황 : 10부터 19까지 정수가 있을때 , 짝수만 결과 화면과 같이 출력하는 프로그램
 - * 10부터 19까지 범위를 지정 : range(10:20)
 - * 범위내에서 짝수 : 2로 나누었을 때 나머지가 0인 수

- ▶ 결과 화면

```
10 is a even number .  
12 is a even number .  
14 is a even number .  
16 is a even number .  
18 is a even number .  
>>>
```

조건문반복문 혼용 활용(3/12)

- ▶ 반복문과 조건문 : 범위내 정수 중 짝수 출력(계속)

- ▶ 해결 코드

```
>>> for num in range(10,20):  
        if num %2 == 0:  
            print(num, ' is a even number.')
```

조건문반복문 혼용 활용(4/12)

- ▶ 반복문과 조건문 : 데이터 타입 판별

- ▶ 상황 : string형과 int형으로 구성된

`mixlist=['apple',3,'banana','포도','메론',6,9]`

각 원소의 형이 string형인지 int형인지 판별하여
결과 화면과 같이 출력하는 프로그램

조건문반복문 혼용 활용(5/12)

결과 화면

```
apple type is string.  
3 type is integer  
banana type is string.  
포도 type is string.  
melon type is string.  
6 type is integer  
9 type is integer
```


조건문반복문 혼용 활용(6/12)

▶ 반복문과 조건문 : 데이터 타입 판별(계속)

▶ 해결 코드

```
mixlist = ['apple',3,'banana','포도','melon',6,9]

for element in mixlist:
    if type(element) == str:
        print(element,' type is string.\n')
    else:
        print(element,' type is integer\n')
```

조건문반복문 혼용 활용(7/12)

▶ 반복문과 조건문 : 음식 궁합 프로그램

▶ 상황 : foods라는 딕셔너리에 키와 값을 넣는다.

딕셔너리의 키에 음식을 입력하면 키 값의 대응하는 값 출력

“끝”을 입력하면 프로그램 종료

결과 화면과 같이 출력하는 프로그램

▶ 결과 화면

```
['떡볶이', '짜장면', '라면', '피자', '맥주', '삼겹살'] 중 좋아하는 것은 ? 라면
<라면> 궁합 음식은 <김치> 입니다.
['떡볶이', '짜장면', '라면', '피자', '맥주', '삼겹살'] 중 좋아하는 것은 ? 끝
```

조건문반복문 혼용 활용(8/12)

▶ 반복문과 조건문 : 음식 궁합 프로그램(계속)

▶ 해결 코드

```
foods = { "떡볶이" : "오뎅",  
          "짜장면" : "단무지",  
          "라면" : "김치",  
          "피자" : "피클",  
          "맥주" : "치킨",  
          "삼겹살" : "소주" }  
  
while (True) :  
    myfood = input( str(list(foods.keys())) + " 중 좋아하는 것은 ? " )  
    if myfood in foods :  
        print(" <%s> 궁합 음식은 <%s> 입니다." % (myfood, foods.get(myfood)))  
    elif myfood == "끝" :  
        break  
    else :  
        print("그런 음식이 없네요. 확인해 보세요. ")
```

조건문반복문 혼용 활용(9/12)

- ▶ 반복문과 조건문 : 계산기 프로그램
 - ▶ 상황 : 덧셈, 뺄셈, 곱셈, 나눗셈, 나머지까지
계산해주는 계산기
결과 화면과 같이 출력하는 프로그램
 - ▶ 결과 화면

```
계산할 첫 번째 수 입력 : 22
계산할 두 번째 수 입력 : 33
계산할 연산자를 입력 : *
22 * 33 = 726 입니다.
계산할 첫 번째 수 입력 : 10
계산할 두 번째 수 입력 : 4
계산할 연산자를 입력 : %
10 % 4 = 2 입니다.
계산할 첫 번째 수 입력 :
```

조건문반복문 혼용 활용(10/12)

- ▶ 반복문과 조건문 : 계산기 프로그램(계속)
- ▶ 해결 코드

```
while (True) :  
    a = int(input("계산할 첫 번째 수 입력 : "))  
    b = int(input("계산할 두 번째 수 입력 : "))  
    ch = input("계산할 연산자를 입력 : ")  
  
    if (ch == "+") :  
        print("%d + %d = %d 입니다." % (a, b, a+b))  
    elif (ch == "-") :  
        print("%d - %d = %d 입니다." % (a, b, a-b))  
    elif (ch == "*") :  
        print("%d * %d = %d 입니다." % (a, b, a*b))  
    elif (ch == "/") :  
        print("%d / %d = %5.2f 입니다." % (a, b, a/b))  
    elif (ch == "%") :  
        print("%d %% %d = %d 입니다." % (a, b, a%b))  
    elif (ch == "//") :  
        print("%d // %d = %d 입니다." % (a, b, a//b))  
    elif (ch == "**") :  
        print("%d ** %d = %d 입니다." % (a, b, a**b))  
    else :  
        print("연산자를 잘못 입력했습니다.")
```

조건문반복문 혼용 활용(11/12)

▶ 반복문과 조건문 : 한쪽 끝이 막힌 주차장 프로그램

- ▶ 상황 : 들어가는 순서는 $A \rightarrow B \rightarrow C$
나오는 순서는 $C \rightarrow B \rightarrow A$

결과 화면과 같이 출력하는 프로그램

▶ 결과 화면

```
<1> 자동차 넣기 <2> 자동차 빼기 <3> 끝 : 1
A 자동차 들어감. 주차장 상태==>['A']
```

```
<1> 자동차 넣기 <2> 자동차 빼기 <3> 끝 : 1
B 자동차 들어감. 주차장 상태==>['A', 'B']
```

```
<1> 자동차 넣기 <2> 자동차 빼기 <3> 끝 : 2
B 자동차 나감. 주차장 상태==>['A']
```

```
<1> 자동차 넣기 <2> 자동차 빼기 <3> 끝 : 3
현재 주차장에 1 대가 있음
```

조건문반복문 혼용 활용(12/12)

▶ 반복문과 조건문 : 한쪽 끝이 막힌 주차장 프로그램(계속)

▶ 해결 코드

```

parking = []
carName = "A"

while (True):
    select = int(input("<1> 자동차 넣기 <2> 자동차 빼기 <3> 끝 : "))

    if (select == 1):
        if (len(parking) >= 5):
            print("주차장이 꽉 차서 못들어감\n")
        else:
            parking.append(carName)
            print("%s 자동차 들어감. 주차장 상태==>%s \n" %(carName, parking))
            carName = chr(ord(carName) + 1)
    elif (select == 2):
        if (len(parking) <= 0):
            print("자동차 없음\n")
        else:
            outCar = parking.pop()
            print("%s 자동차 나감. 주차장 상태==>%s \n" %(outCar, parking))
            carName = chr(ord(carName) - 1)
    elif (select == 3):
        break
    else:
        print("잘 못 입력 \n")

print("현재 주차장에 %d 대가 있음" %len(parking))

```

Thank you