

Lecture 03

Python Conditionals and Loops for Data Processing

조건문 (Conditional Statement)

1. if 조건문
2. 조건이 참일 경우에만 수행할 작업을 지정하기
3. 조건이 참일 경우와 거짓일 경우 모두 수행할 작업을 지정하기
4. 조건문과 함께 자주 사용되는 논리 연산자

반복문 (Iteration Statement)

1. for 반복문
2. while 반복문

분기문 (Branch Statement)

1. break
2. continue

Exercise (1)
Exercise (2)

기호	의 미
==	같다
!=	같지 않다
>	크다
>=	크거나 같다
<	작다
<=	작거나 같다

조건문 (Conditional Statement)

1. **if 조건문**
2. 조건이 참일 경우에만 수행할 작업을 지정하기
3. 조건이 참일 경우와 거짓일 경우 모두 수행할 작업을 지정하기
4. 조건문과 함께 자주 사용되는 논리 연산자

반복문 (Iteration Statement)

1. for 반복문
2. while 반복문

분기문 (Branch Statement)

1. break
2. continue

Exercise (1)
Exercise (2)

[Syntax]

if 조건 :

실행 문장 1

실행 문장 2

실행 문장 3

[주의사항 2가지]

1. if 조건 뒤에 **:** (콜론기호) 반드시 쓸 것
2. 들여쓰기(indentation) 주의할 것

조건문 (Conditional Statement)

1. if 조건문
2. 조건이 참일 경우에만 수행할 작업을 지정하기
3. 조건이 참일 경우와 거짓일 경우 모두 수행할 작업을 지정하기
4. 조건문과 함께 자주 사용되는 논리 연산자

반복문 (Iteration Statement)

1. for 반복문
2. while 반복문

분기문 (Branch Statement)

1. break
2. continue

Exercise (1)

Exercise (2)

• if 조건문 활용

1) 예제 1

```
1 name = input('조회할 사람의 이름을 입력하세요: ')
2
3 if name=='홍길동' :
4     print("홍길동 입니다")
5     print("남자입니다")
6     print("활빈당 총수입니다")
```

조회할 사람의 이름을 입력하세요:

홍길동 입니다

남자입니다

활빈당 총수입니다

조건문 (Conditional Statement)

1. if 조건문
2. 조건이 참일 경우에만 수행할 작업을 지정하기
3. 조건이 참일 경우와 거짓일 경우 모두 수행할 작업을 지정하기
4. 조건문과 함께 자주 사용되는 논리 연산자

반복문 (Iteration Statement)

1. for 반복문
2. while 반복문

분기문 (Branch Statement)

1. break
2. continue

Exercise (1)

Exercise (2)

2) 예제 2 : indentation 잘못해서 나오는 에러

```
1 name = input('조회할 사람의 이름을 입력하세요: ')
2
3 if name=='홍길동' :
4     print("홍길동 입니다")
5     print("남자입니다")
6     print("활빈당 총수입니다")
```

File "<ipython-input-2-5e487133f52d>", line 6

```
    print("활빈당 총수입니다")
    ^
```

IndentationError: unexpected indent

조건문 (Conditional Statement)

1. if 조건문
2. 조건이 참일 경우에만
수행할 작업을
지정하기
3. 조건이 참일 경우와
거짓일 경우 모두
수행할 작업을
지정하기
4. 조건문과 함께
자주 사용되는
논리 연산자

[Syntax]

if 조건 :

실행 문장 1

else :

실행 문장 2

반복문 (Iteration Statement)

1. for 반복문
2. while 반복문

분기문 (Branch Statement)

1. break
2. continue

Exercise (1)

Exercise (2)

조건문 (Conditional Statement)

1. if 조건문
2. 조건이 참일 경우에만
수행할 작업을
지정하기
3. 조건이 참일 경우와
거짓일 경우 모두
수행할 작업을
지정하기
4. 조건문과 함께
자주 사용되는
논리 연산자

반복문 (Iteration Statement)

1. for 반복문
2. while 반복문

분기문 (Branch Statement)

1. break
2. continue

Exercise (1)
Exercise (2)

3) 예제 3

```
1 name = input('조회할 사람의 이름을 입력하세요: ')
2
3 if name=='홍길동' :
4     print("홍길동 입니다")
5     print("남자입니다")
6     print("활빈당 총수입니다")
7 else :
8     print("홍길동이 아니고 %s 입니다" %name)
9     print("남자인가요?")
10    print("여자인가요?")
```

조회할 사람의 이름을 입력하세요:

홍길동이 아니고 성춘향 입니다

남자인가요?

여자인가요?

조건문 (Conditional Statement)

1. if 조건문
2. 조건이 참일 경우에만
수행할 작업을
지정하기
3. 조건이 참일 경우와
거짓일 경우 모두
수행할 작업을
지정하기
4. 조건문과 함께
자주 사용되는
논리 연산자

반복문 (Iteration Statement)

1. for 반복문
2. while 반복문

분기문 (Branch Statement)

1. break
2. continue

Exercise (1)
Exercise (2)

[Syntax] 여러 개의 조건에 따라 다른 작업을 해야 하는 경우

if 조건1 :

실행 문장 1

elif 조건2 :

실행 문장 2

elif 조건3 :

실행 문장 3

else :

실행 문장 4

조건문 (Conditional Statement)

조건문 (Conditional Statement)

1. if 조건문
2. 조건이 참일 경우에만
수행할 작업을
지정하기
3. 조건이 참일 경우와
거짓일 경우 모두
수행할 작업을
지정하기
4. 조건문과 함께
자주 사용되는
논리 연산자

연산자	의 미
A and B (또는 &)	A와 B 두 조건 모두 만족할 경우 실행하기
A or B (또는)	A 나 B 둘 중 하나만 만족할 경우 실행하기
not A (또는 !=)	A가 아닐 때 실행하기

반복문 (Iteration Statement)

1. for 반복문
2. while 반복문

참	거짓
1 (0이 아닌 모든 숫자)	0
True	False

분기문 (Branch Statement)

1. break
2. continue

Exercise (1)
Exercise (2)

조건문 (Conditional Statement)

1. if 조건문
2. 조건이 참일 경우에만
수행할 작업을
지정하기
3. 조건이 참일 경우와
거짓일 경우 모두
수행할 작업을
지정하기
4. 조건문과 함께
자주 사용되는
논리 연산자

반복문 (Iteration Statement)

1. for 반복문
2. while 반복문

분기문 (Branch Statement)

1. break
2. continue

Exercise (1)
Exercise (2)

4) 예제 4 : and 연산자 사용

```
1 jumsu = int(input("점수를 입력하세요: "))
2
3 if jumsu >= 91 and jumsu <= 100 :
4     print("A 등급입니다!")
5 elif jumsu >= 81 and jumsu <= 90 :
6     print("B 등급입니다!")
7 elif jumsu >= 71 and jumsu <= 80 :
8     print("C 등급입니다!")
9 else :
10    print("D 등급입니다!")
```

점수를 입력하세요:
B 등급입니다!

조건문 (Conditional Statement)

1. if 조건문
2. 조건이 참일 경우에만
수행할 작업을
지정하기
3. 조건이 참일 경우와
거짓일 경우 모두
수행할 작업을
지정하기
4. 조건문과 함께
자주 사용되는
논리 연산자

반복문 (Iteration Statement)

1. for 반복문
2. while 반복문

분기문 (Branch Statement)

1. break
2. continue

- Exercise (1)
Exercise (2)

5) 예제 5 : & 연산자 사용

```
1 jumsu = int(input("점수를 입력하세요: "))
2
3 # &를 사용할 경우에는 관계연산자 부분에 반드시 괄호 연산자를 사용해야 함
4 if (jumsu >= 91) & (jumsu <= 100) :
5     print("A 등급입니다!")
6 elif (jumsu >= 81) & (jumsu <= 90) :
7     print("B 등급입니다!")
8 elif (jumsu >= 71) & (jumsu <= 80) :
9     print("C 등급입니다!")
10 else :
11     print("D 등급입니다!")
```

점수를 입력하세요: 88

B 등급입니다!

조건문 (Conditional Statement)

1. if 조건문
2. 조건이 참일 경우에만
수행할 작업을
지정하기
3. 조건이 참일 경우와
거짓일 경우 모두
수행할 작업을
지정하기
4. 조건문과 함께
자주 사용되는
논리 연산자

반복문 (Iteration Statement)

1. for 반복문
2. while 반복문

분기문 (Branch Statement)

1. break
2. continue

Exercise (1)

Exercise (2)

6) 예제 6 : or 연산자 사용

```
1 answer = input("Y 또는 y 를 입력하세요: ")
2
3 if answer == 'Y' or answer == 'y' :
4     print("입력하신 문자는 %s 입니다" %answer)
5 else :
6     print("Y 또는 y 를 입력하세요")
7
```

Y 또는 y 를 입력하세요:
입력하신 문자는 y 입니다

조건문 (Conditional Statement)

1. if 조건문
2. 조건이 참일 경우에만
수행할 작업을
지정하기
3. 조건이 참일 경우와
거짓일 경우 모두
수행할 작업을
지정하기
4. 조건문과 함께
자주 사용되는
논리 연산자

반복문 (Iteration Statement)

1. for 반복문
2. while 반복문

분기문 (Branch Statement)

1. break
2. continue

Exercise (1)

Exercise (2)

7) 예제 7 : != 연산자 사용

```
1 answer = input("Y 또는 y 를 입력하세요: ")
2
3 if answer != 'Y' and answer != 'y' :
4     print("입력하신 문자는 %s 입니다" %answer)
5 else :
6     print("Y 또는 y 를 입력하셨군요~")
7
```

Y 또는 y 를 입력하세요:
입력하신 문자는 k 입니다

조건문 (Conditional Statement)

1. if 조건문
2. 조건이 참일 경우에만 수행할 작업을 지정하기
3. 조건이 참일 경우와 거짓일 경우 모두 수행할 작업을 지정하기
4. 조건문과 함께 자주 사용되는 논리 연산자

반복문 (Iteration Statement)

1. **for** 반복문
2. **while** 반복문

분기문 (Branch Statement)

1. **break**
2. **continue**

Exercise (1)
Exercise (2)

[Syntax]

for 변수 **in** 반복횟수 :

실행할 문장 1

실행할 문장 2

실행할 문장 3

조건문 (Conditional Statement)

1. if 조건문
2. 조건이 참일 경우에만 수행할 작업을 지정하기
3. 조건이 참일 경우와 거짓일 경우 모두 수행할 작업을 지정하기
4. 조건문과 함께 자주 사용되는 논리 연산자

반복문 (Iteration Statement)

1. for 반복문
2. while 반복문

분기문 (Branch Statement)

1. break
2. continue

Exercise (1)

Exercise (2)

• for 반복문 활용

1) 예제 1 : tuple 형식 활용

```
1 a = (10,20,30)
2 print("a : ", a)
3 print("a 의 데이터 형식은 ", type(a))
4
5 for i in a :
6     print( i )
```

```
a : (10, 20, 30)
a 의 데이터 형식은 <class 'tuple'>
10
20
30
```

```
1 for i in (10, 20, 30) :
2     print( i )
```

```
10
20
30
```

조건문

(Conditional Statement)

1. if 조건문
2. 조건이 참일 경우에만 수행할 작업을 지정하기
3. 조건이 참일 경우와 거짓일 경우 모두 수행할 작업을 지정하기
4. 조건문과 함께 자주 사용되는 논리 연산자

반복문

(Iteration Statement)

1. for 반복문
2. while 반복문

분기문

(Branch Statement)

1. break
2. continue

Exercise (1)

Exercise (2)

2) 예제 2 : list 형식 활용

```
1 data = ["하나", "둘", "셋"]
2 for i in data :
3     print( i )
4
5 print("data : ", data)
6 print("data 의 데이터 형식은 ", type(data))
```

하나

둘

셋

data : ['하나', '둘', '셋']

data 의 데이터 형식은 <class 'list'>

```
1 data = ["김구", "유관순", "이순신"]
2 for i in data :
3     print("%s 님은 대한민국의 위대한 위인이고 자랑스롭니다~!" % i)
```

김구 님은 대한민국의 위대한 위인이고 자랑스롭니다~!

유관순 님은 대한민국의 위대한 위인이고 자랑스롭니다~!

이순신 님은 대한민국의 위대한 위인이고 자랑스롭니다~!

조건문 (Conditional Statement)

1. if 조건문
2. 조건이 참일 경우에만
수행할 작업을
지정하기
3. 조건이 참일 경우와
거짓일 경우 모두
수행할 작업을
지정하기
4. 조건문과 함께
자주 사용되는
논리 연산자

반복문 (Iteration Statement)

1. for 반복문
2. while 반복문

분기문 (Branch Statement)

1. break
2. continue

Exercise (1)
Exercise (2)

3) 예제 3 : range 활용

```
1 for i in range(1,5) :  
2     print("밥 %s 그릇 주세요~" %i)
```

밥 1 그릇 주세요~
밥 2 그릇 주세요~
밥 3 그릇 주세요~
밥 4 그릇 주세요~

조건문 (Conditional Statement)

1. if 조건문
2. 조건이 참일 경우에만 수행할 작업을 지정하기
3. 조건이 참일 경우와 거짓일 경우 모두 수행할 작업을 지정하기
4. 조건문과 함께 자주 사용되는 논리 연산자

반복문 (Iteration Statement)

1. for 반복문
2. **while** 반복문

분기문 (Branch Statement)

1. break
2. continue

Exercise (1)
Exercise (2)

[Syntax]

while 조건 :

실행 문장 1

실행 문장 2

반복 조건을 증감하는 문장

....

조건문 (Conditional Statement)

1. if 조건문
2. 조건이 참일 경우에만 수행할 작업을 지정하기
3. 조건이 참일 경우와 거짓일 경우 모두 수행할 작업을 지정하기
4. 조건문과 함께 자주 사용되는 논리 연산자

반복문 (Iteration Statement)

1. for 반복문
2. while 반복문

분기문 (Branch Statement)

1. break
2. continue

Exercise (1)
Exercise (2)

• while 반복문 활용

1) 예제 1

```
1 no1 = 1
2 while no1 < 9 :
3     print(no1)
4     no1 += 2
```

1
3
5
7

2) 예제 2

```
1 no1 = 0
2 while no1 < 9 :
3     no1 += 2
4     print(no1)
```

2
4
6
8
10

조건문 (Conditional Statement)

1. if 조건문
2. 조건이 참일 경우에만 수행할 작업을 지정하기
3. 조건이 참일 경우와 거짓일 경우 모두 수행할 작업을 지정하기
4. 조건문과 함께 자주 사용되는 논리 연산자

반복문 (Iteration Statement)

1. for 반복문
2. while 반복문

분기문 (Branch Statement)

1. break
2. continue

Exercise (1)
Exercise (2)

- 반복문, 조건문, break, continue 혼합 활용

1) 예제 1 : break 활용 - 반복문 탈출

break

중간에 멈추기

```
1 snack = ["새우깡", "감자깡", "고구마깡", "맛동산", "짱구"]
2 for i in snack :
3     if i == "고구마깡" :
4         break
5     print(i, " 완전 맛있어요~!")
```

새우깡 완전 맛있어요~!

감자깡 완전 맛있어요~!

조건문

(Conditional Statement)

1. if 조건문
2. 조건이 참일 경우에만 수행할 작업을 지정하기
3. 조건이 참일 경우와 거짓일 경우 모두 수행할 작업을 지정하기
4. 조건문과 함께 자주 사용되는 논리 연산자

반복문

(Iteration Statement)

1. for 반복문
2. while 반복문

분기문

(Branch Statement)

1. break
2. continue

Exercise (1)

Exercise (2)

2) 예제 2 : continue 활용 - 반복문 루틴으로 가서 실행을 계속 이어감

```
1 snack = ["새우깡", "감자깡", "고구마깡", "맛동산", "짱구"]
2 for i in snack :
3     if i == "고구마깡" :
4         continue
5     print(i, " 완전 맛있어요~!")
```

새우깡 완전 맛있어요~!

감자깡 완전 맛있어요~!

맛동산 완전 맛있어요~!

짱구 완전 맛있어요~!

```
1 snack = ["새우깡", "고구마깡", "맛동산", "짱구"]
2 for i in snack :
3     if i == "고구마깡" :
4         continue
5     print(i)
```

새우깡

맛동산

짱구

continue

건너뛰어
계속 실행하기

조건문

(Conditional Statement)

1. if 조건문
2. 조건이 참일 경우에만 수행할 작업을 지정하기
3. 조건이 참일 경우와 거짓일 경우 모두 수행할 작업을 지정하기
4. 조건문과 함께 자주 사용되는 논리 연산자

반복문

(Iteration Statement)

1. for 반복문
2. while 반복문

분기문

(Branch Statement)

1. break
2. continue

Exercise (1)

Exercise (2)

아래와 같은 결과가 나오도록 코딩하시오. (프로그램 순서 : 다음 슬라이드)

- | | |
|-----------------|------------------|
| 1. 짜장면 - 5,000원 | 2. 짬뽕 - 6,000원 |
| 3. 군만두 - 8,000원 | 4. 탕수육 - 10,000원 |

1. 위 메뉴 중 주문할 메뉴의 번호를 쓰세요:
2. 위 메뉴의 주문 수량을 쓰세요:

주문하신 메뉴는 짬뽕이고 주문 수량은 3 그릇이며 주문금액은 18000 입니다

3. 추가 주문을 하시겠습니까? (Y / N) :

- | | |
|-----------------|------------------|
| 1. 짜장면 - 5,000원 | 2. 짬뽕 - 6,000원 |
| 3. 군만두 - 8,000원 | 4. 탕수육 - 10,000원 |

1. 위 메뉴 중 주문할 메뉴의 번호를 쓰세요:
2. 위 메뉴의 주문 수량을 쓰세요:

주문하신 메뉴는 탕수육이고 주문 수량은 2 그릇이며 주문금액은 20000 입니다

3. 추가 주문을 하시겠습니까? (Y / N) :

조건문 (Conditional Statement)

1. if 조건문
2. 조건이 참일 경우에만 수행할 작업을 지정하기
3. 조건이 참일 경우와 거짓일 경우 모두 수행할 작업을 지정하기
4. 조건문과 함께 자주 사용되는 논리 연산자

반복문 (Iteration Statement)

1. for 반복문
2. while 반복문

분기문 (Branch Statement)

1. break
2. continue

Exercise (1) Exercise (2)

프로그램 순서

- ① 메뉴 이름과 가격을 출력한다.
- ② 주문할 **메뉴의 번호**를 입력 받는다.
- ③ 이어서 **주문할 수량**을 입력 받는다.
- ④ 주문 내역을 요약해서 출력한다.
- ⑤ **“추가 주문을 하시겠습니까? (Y / N) : ”** 문장이 출력되면 사용자는 다음과 같이 입력할 수 있다.
 - “Y” 또는 “y” 를 입력
 - ① 에서 보여준 메뉴 이름과 가격을 다시 출력하여 계속 실행한다.
 - “Y” 나 “y” 이 아닌 다른 문자가 입력
 - 종료

조건문

(Conditional Statement)

1. if 조건문
2. 조건이 참일 경우에만 수행할 작업을 지정하기
3. 조건이 참일 경우와 거짓일 경우 모두 수행할 작업을 지정하기
4. 조건문과 함께 자주 사용되는 논리 연산자

반복문

(Iteration Statement)

1. for 반복문
2. while 반복문

분기문

(Branch Statement)

1. break
2. continue

Exercise (1)

Exercise (2)

Exercise1에서 추가 주문을 받는 횟수가 3회가 넘을 경우,
아래와 같이 안내 메시지를 보이면서 프로그램이 종료되도록 코드를 수정하시오.

1. 짜장면 - 5,000원 2. 짬뽕 - 6,000원
3. 군만두 - 8,000원 4. 탕수육 - 10,000원

1. 위 메뉴 중 주문할 메뉴의 번호를 쓰세요 : 4
2. 위 메뉴의 주문 수량을 쓰세요 : 2

주문하신 메뉴는 탕수육이고 주문 수량은 2 그릇이며 주문금액은 20000 원 입니다

3. 추가 주문을 하시겠습니까? (Y / N) : y
1 번째 추가 주문입니다

1. 짜장면 - 5,000원 2. 짬뽕 - 6,000원
3. 군만두 - 8,000원 4. 탕수육 - 10,000원

1. 위 메뉴 중 주문할 메뉴의 번호를 쓰세요 : 1
2. 위 메뉴의 주문 수량을 쓰세요 : 2

주문하신 메뉴는 짜장면이고 주문 수량은 2 그릇이며 주문금액은 10000 원 입니다

3. 추가 주문을 하시겠습니까? (Y / N) : y
2 번째 추가 주문입니다

1. 짜장면 - 5,000원 2. 짬뽕 - 6,000원
3. 군만두 - 8,000원 4. 탕수육 - 10,000원

1. 위 메뉴 중 주문할 메뉴의 번호를 쓰세요 : 3
2. 위 메뉴의 주문 수량을 쓰세요 : 2

주문하신 메뉴는 군만두이고 주문 수량은 2 그릇이며 주문금액은 16000 원 입니다

3. 추가 주문을 하시겠습니까? (Y / N) : y
3 번째 추가 주문입니다

1. 짜장면 - 5,000원 2. 짬뽕 - 6,000원
3. 군만두 - 8,000원 4. 탕수육 - 10,000원

1. 위 메뉴 중 주문할 메뉴의 번호를 쓰세요 : 2
2. 위 메뉴의 주문 수량을 쓰세요 : 1

주문하신 메뉴는 짬뽕이고 주문 수량은 1 그릇이며 주문금액은 6000 원 입니다

추가주문은 총 3회 까지만 가능합니다 π π