

# LAB 04

2020-01 소프트웨어의 이해 01분반 / 조교 이경민

# LAB 04

---

- If
- If-elif-else
- 논리연산
- 예제
- 실습 과제

# if 문

- if 문 구조

- 조건이 True 면, 문장1을 실행

- 들여쓰기 매우 매우 중요



If 조건식 :  
문장1

```
balance = 10000
print("You have " + str(balance) + " won.")
amount = int(input("Write amount to deposit : "))

if amount <= balance :
    balance = balance - amount
print("Current Balance :", balance)
```

```
===== RES
===
You have 10000 won.
Write amount to deposit : 5000
Current Balance : 5000
>>>
===== RES
===
You have 10000 won.
Write amount to deposit : 20000
Current Balance : 10000
```

# 논리 연산

---

- 두 개 이상의 비교연산이 필요할 때
  - 각 조건에 **괄호**를 해주는 것이 좋음

```
if (amount <= balance) and (amount > 0) :  
    balance = balance - amount  
print("Current Balance :", balance)
```

| 연산   | 연산자     | 결과                    |
|------|---------|-----------------------|
| 논리합  | A or B  | A나 B 조건 중 하나만 만족해도 실행 |
| 논리곱  | A and B | A와 B 두 조건을 모두 만족해야 실행 |
| 논리부정 | not A   | A 조건이 아니면 실행          |

# True / False

---

- True / False의 또 다른 형태
- ex) if 1 :  
      if True :

표 6.4 여러 자료형의 참과 거짓

| 자료형 | 참                        | 거짓 |
|-----|--------------------------|----|
| 숫자  | 0이 아닌 숫자, 예: 3           | 0  |
| 문자열 | "" 이 아닌 문자열, 예: "abc"    | "" |
| 리스트 | [ ]이 아닌 리스트, 예: [1,2,3]  | [] |
| 튜플  | ( )이 아닌 튜플, 예: (1,2,3)   | () |
| 사전  | { }이 아닌 사전, 예: {"a":"b"} | {} |

# if-else 문

- if-else 문 구조
  - 조건이 True면 : 문장1을 실행
  - 조건이 False면 (if 조건식의 모든 반대의 경우) : 문장2를 실행

If 조건식 :  
문장1  
else :  
문장2

```
balance = 10000
print("You have "+str(balance)+" won.")
amount = int(input("Write amount to deposit : "))

if amount <= balance :
    balance = balance - amount
    print("Current Balance :", balance)
else :
    print("Lack of balance")
```

```
===== REST
==
You have 10000 won.
Write amount to deposit : 20000
Lack of balance
>>>
===== REST
==
You have 10000 won.
Write amount to deposit : 4000
Current Balance : 6000
```

# if-elif 문

---

- if-elif 문 구조

- 조건식1이 True면 : 문장1을 실행
- 조건식2가 True면 : 문장2를 실행

```
if 조건식1 :  
    문장1  
elif 조건식2 :  
    문장2
```

- if-elif-else 문을 사용할 때, elif 끝에 꼭 else를 사용해야 하나요?  
사용할 수 있는 개수에 제한이 있나요?

- 제한 없음.  
if 문을 처음에 사용하는 것을 제외하고는 프로그래머가 원하는 대로 elif, else를 사용할 수 있습니다.
- 단, else는 무조건 마지막에 딱 한 번만 사용할 수 있습니다. (모든 조건의 반대를 의미하기 때문)

## 예제 1) 학점 계산 예제

- 점수에 따라 학점을 계산하는 프로그램

```
score = int(input("점수를 입력하세요: "))
if score < 0 or score > 100 :
    print("잘못된 점수입니다")
elif score >= 90 :
    print("A")
elif score >= 80 :
    print("B")
elif score >= 70 :
    print("C")
elif score >= 60 :
    print("D")
else :
    print("F")
```

```
=====
==
점수를 입력하세요: 86
B
>>>
=====
==
점수를 입력하세요: 110
잘못된 점수입니다
>>>
=====
==
점수를 입력하세요: -5
잘못된 점수입니다
>>>
=====
==
점수를 입력하세요: 44
F
```



## 예제 2) 가장 큰 숫자 구하기

- 3개의 숫자를 입력 받아 가장 큰 수를 출력하는 프로그램

```
print("세 수를 입력하세요")  
a = float(input())  
b = float(input())  
c = float(input())
```

```
if a > b :  
    if a > c :  
        max = a  
    else :  
        max = c
```

```
else :  
    if b > c :  
        max = b  
    else :  
        max = c
```

```
print("가장 큰 수 :", max)
```

세 수를 입력하세요

24

96

100

가장 큰 수 : 100.0

## 예 제 3) BMI 계 산 예 제

---

- 체질량지수(BMI) = 몸무게 / (키 \* 키)
- 단위
  - 몸무게 : kg
  - 키 : m (1m = 100cm)

| 체질량지수       | 비만 정도 |
|-------------|-------|
| 30 이상       | 고도 비만 |
| 25 ~ 30     | 비만    |
| 23 ~ 24.9   | 과체중   |
| 18.5 ~ 22.9 | 정상    |
| 18.5 미만     | 저체중   |

### 예제 3) BMI 계산 예제

```
print("키와 몸무게를 입력하면 체질량지수를 계산합니다.")
```

```
height = float(input("키(cm) : "))
```

```
weight = float(input("몸무게(kg) : "))
```

```
BMI = weight / ((height/100) * (height/100))
```

$(height/100)**2$

```
print("BMI : %4.1f" % BMI)
```

```
if BMI < 18.5 :
```

```
    print("저체중")
```

```
elif BMI < 23 :
```

```
    print("정상")
```

```
elif BMI < 25 :
```

```
    print("과체중")
```

```
elif BMI < 30 :
```

```
    print("비만")
```

```
else :
```

```
    print("고도비만")
```

===== RESTART: C:/SW\_19

===

키와 몸무게를 입력하면 체질량지수를 계산합니다.

키(cm) : 185

몸무게(kg) : 77

BMI : 22.5

정상

## LAB04 실습과제 1 (bank.py)

---

- 계좌 소유주 이름과 원하는 업무 (입금/인출/잔액확인/종료)를 입력 받아 계산하는 프로그램
- 조건
  - 단, 4를 입력할 때까지 계속해서 프로그램이 실행되어야 한다. (다음페이지 구조 참고)
  - 초기 잔액 : 100,000원으로 지정
  - 출금 시, 0 이하인 값을 인출하려 할 경우 "정확한 금액을 입력해주세요." 라는 메시지 출력
  - 현재 잔액보다 큰 값을 인출하려 할 경우 "잔액 부족. 거래 거절되었습니다." 라는 메시지 출력
- 다음 페이지 실행화면 참고

# LAB04 실습과제 1 (bank.py)

- 다음 구조를 사용할 것 (다음 시간에 자세히 배울 예정)

넣으세요!

...생략...

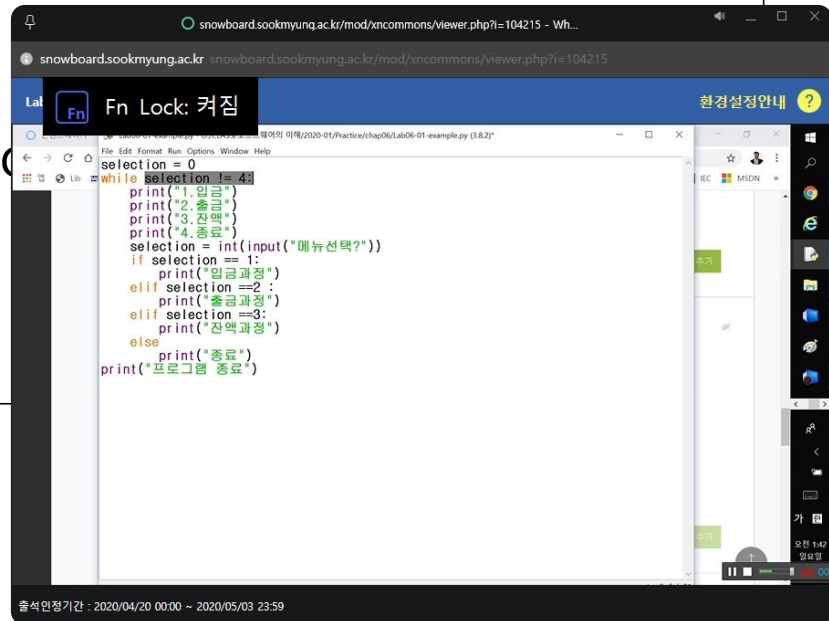
```
selection = 0
while selection != 4 :
```

...생략...

selection

...생략...

...생략...



```
selection = 0
while selection != 4:
    print("1.입금")
    print("2.출금")
    print("3.잔액")
    print("4.종료")
    selection = int(input("메뉴선택?"))
    if selection == 1:
        print("입금과정")
    elif selection == 2:
        print("출금과정")
    elif selection == 3:
        print("잔액과정")
    else:
        print("종료")
print("프로그램 종료")
```

## LAB04 실습과제 1 (bank.py)

---

```
=====
                        은행 ATM
=====
계좌 소유자의 이름을 입력해주세요 : 이경민
1. 입금
2. 출금
3. 잔액확인
4. 종료
=====
메뉴선택 : 1
입금할 금액을 입력해주세요 : 40000

40000원이 성공적으로 입금되었습니다.

=====
                        은행 ATM
=====
1. 입금
2. 출금
3. 잔액확인
4. 종료
=====
메뉴선택 : 2
출금할 금액을 입력해주세요 : -500

정확한 금액을 입력해주세요.
```

## LAB04 실습과제 1 (bank.py)

---

```
=====
                        은행 ATM
=====
1. 입금
2. 출금
3. 잔액확인
4. 종료
=====
메뉴선택 : 2
출금할 금액을 입력해주세요 : 0

정확한 금액을 입력해주세요.

=====
                        은행 ATM
=====
1. 입금
2. 출금
3. 잔액확인
4. 종료
=====
메뉴선택 : 2
출금할 금액을 입력해주세요 : 500000

잔액 부족. 거래 거절되었습니다.
```

## LAB04 실습과제 1 (bank.py)

---

```
=====
                        은행 ATM
=====
1. 입금
2. 출금
3. 잔액확인
4. 종료
=====
메뉴선택 : 3

이경민님의 현재 잔액은 140000원입니다.

=====
                        은행 ATM
=====
1. 입금
2. 출금
3. 잔액확인
4. 종료
=====
메뉴선택 : 2
출금할 금액을 입력해주세요 : 40000

40000원이 성공적으로 인출되었습니다.
```



## LAB04 실습과제 1 (bank.py)

---

```
=====
                        은행 ATM
=====
1. 입금
2. 출금
3. 잔액확인
4. 종료
=====
메뉴선택 : 3

이경민님의 현재 잔액은 100000원입니다.

=====
                        은행 ATM
=====
1. 입금
2. 출금
3. 잔액확인
4. 종료
=====
메뉴선택 : 4

이용해주셔서 감사합니다.
```

## LAB04 실습과제 2 (leap\_year.py)

---

- 연도를 입력 받아서 윤년인지 확인하는 프로그램 작성
- 조건
  - 윤년은 4로 나누어 떨어지고 **동시에** 100으로 나누어 떨어지지 않는 해는 윤년으로 한다.
  - **혹은**, 400으로 나누어 떨어지는 해는 윤년으로 둔다.

✓

```
===== RESTART: C:\>
연도를 입력하세요 : 2019
2019년은 윤년이 아닙니다.
>>>
===== RESTART: C:\>
연도를 입력하세요 : 2020
2020년은 윤년입니다.
>>>
===== RESTART: C:\>
연도를 입력하세요 : 2000
2000년은 윤년입니다.
>>>
===== RESTART: C:\>
연도를 입력하세요 : 2100
2100년은 윤년이 아닙니다.
```

# 과제 채점 기준·기한

---

- 과제 제출 기한
  - 실습 다음주 화요일 오후 11시까지
  - 4월 28일 화요일 오후 11시까지 제출
- 제출 장소
  - 스노우보드 해당 주차 과제 제출 페이지에 업로드
- 추가 제출
  - 제출기한 이후 24시간 이내 업로드 : 만점에서 20% 감점
  - 그 이후는 받지 않음
- 표절 X

# 과제 제출 형식 & 질문 메일

---

- **제출물** : 소스파일(.py)과 과제보고서(.docx) 합친 **압축파일 제출**
  - **소스파일 이름** : 매 실습 과제 마다 ppt에 제시 예정
    - ex) test.py
  - **과제 보고서 양식** : 스노우보드에서 다운로드
- **(소스파일 + 과제보고서) 압축 파일 이름 : Lab04\_본인학번\_이름**
  - ex) Lab04\_1911123\_이경민.zip
- **이메일** : newkml22@gmail.com
- **질문 시 주의사항**
  - **과목, 분반, 이름, 학번** 알려주세요.
  - 몇 번 과제에서, 어떤 부분이 막혔는지, 어떤 과정인지 **설명**과 함께 보내주세요.
  - 답장까지 시간이 걸릴 수도 있으니 제출 과제 질문은 미리 해주세요!