

# codingOn x posco

K-Digital Training 신재생에너지 활용 IoT 과정

# Python 조건문과 반복문

# 오늘 수업은?

- 지금까지는 개발에 필요한 기초 개념에 대해서 공부하였습니다.
- 이번시간은 조건문과 반복문을 학습하면서 본격적인 프로그래밍에 대해서 공부하도록 하겠습니다.
- 조건문과 반복문만 안다면 프로그래밍을 할 수 있다고 말할 정도로 매우 중요하고 그만큼 비중도 높습니다. 한 언어만 배워도 다른언어에 똑같이 적용할 수 있습니다.

# 학습목표

- if, else, elif, 중첩 조건문에 대해서 이해한다.
- while문에 대해서 이해한다.
- for문과 이중for문에 대해서 이해한다.

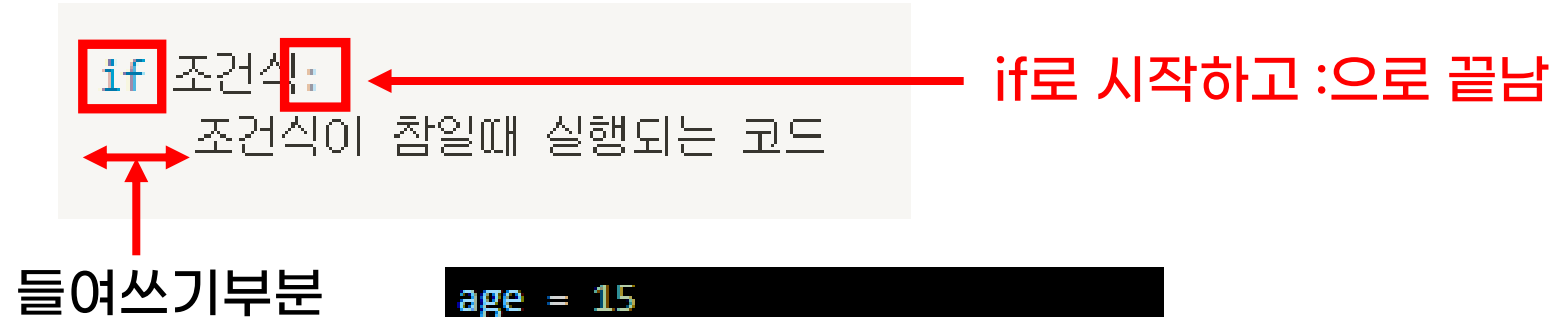
# 1. 조건문

# 우유를 왜 6개 사왔을까?



# 조건문

- ‘만약에’ 라는 if의 의미처럼 조건을 걸 때 사용하여 프로그램의 흐름을 제어
- 컴퓨터에게 만약 OO 라면, OO 해줘 라는 명령을 내리는 것!
- if키워드를 사용하기 때문에 if문이라고 함



```
age = 15

# 조건문
if age >= 20:
    print("미성년자가 아닙니다")

print(f'나이는 {age}세입니다.')
```

age >= 20은 조건문의 조건식  
조건식이 참(True)일때 조건문의 코드가 실행

# 조건문에서 거짓

- 숫자형 : 0은 거짓, 그외의 숫자는 참
- 문자열 : 빈문자열
- 리스트, 딕셔너리, 튜플, 셋(Set): 값이 비어있는 경우 [], {}, (), set()
- None
- False



# 실습. if

- 비밀번호 abc123을 입력했을때 '비밀번호가 맞습니다' 출력시켜주기

```
비밀번호를 입력하세요: abc123  
비밀번호가 맞습니다
```

- 짝수 숫자를 입력했을때 '짝수입니다' 출력시켜주기

```
숫자를 입력하세요: 12  
짝수입니다.
```

# else문

- 지금까지는 조건식이 참일때만 코드가 실행되게 해왔음
- 그렇다면 참이 아닌 거짓일때 실행되게 하려면?
- 만약에(if) => 참이 실행되는부분
- 그렇지 않다면(else)를 사용하면됨 => 거짓이 실행되는 부분

```
if 조건식:  
    조건식이 참일때 실행되는 코드  
else:  
    조건식이 거짓일때 실행되는 코드
```

# else문

- 주의) else는 단독으로 사용할 수 없고 무조건 if문과 함께 사용해야함
- 주의) else에는 조건식이 올 수 없음
- 주의) else는 항상 마지막에 와야함

```
age = 15

# 조건 문
if age >= 20:
    print("미성년자가 아닙니다")
else:
    print('미성년자입니다.')

print(f'나이는 {age}세입니다.')
```

if문 조건식이 참이 아니면 else문이 실행

# 실습. if ~ else

- 비밀번호 abc123을 입력했을때 '비밀번호가 맞습니다' 출력시켜주기
- 비밀번호가 틀리면 '비밀번호가 틀렸습니다' 출력시켜주기

```
비밀번호를 입력하세요 : abc1234  
비밀번호가 틀렸습니다
```

- 짝수 숫자를 입력했을때 '짝수입니다' 출력시켜주기
- 짝수가 아닌 수를 입력했을때 '홀수입니다' 출력시켜주기

```
숫자를 입력하세요 : 15  
홀수입니다.
```

- if~else문 같은경우 두가지의 경우에만 사용할 수 있음
- 하지만 개발을 진행하다보면 두가지 이상의 조건을 써야할때가 더 많음
- 이때 사용해야하는 것이 elif문
- elif문은 필요한 만큼 계속 작성해도 됨
- if와 elif의 조건식중 참인부분의 코드가 실행
- 만약에(if) => 참이면 실행되는부분
- 그렇지 않고 만약에(elif) => 참이면 실행되는 부분
- 그렇지 않다면(else) => 위 조건 모두 거짓일때 실행되는 부분
- else는 선택사항으로 안써도 무관

```
if 조건식1:
    조건식1이 참일때 실행되는 코드
elif 조건식2:
    조건식2가 참일때 실행되는 코드
elif 조건식3:
    조건식3이 참일때 실행되는 코드
elif 조건식4:
    조건식4가 참일때 실행되는 코드
.
.
.
else:
    위 모든 조건이 거짓일때 실행되는 코드
```

주의) elif는 단독으로 사용할 수 없고 무조건 if문과 함께 사용해야함

주의) elif문은 조건식이 있어야함

주의) else는 항상 마지막에 와야함

```
age = int(input("나이를 입력하세요: "))

if age < 20:
    print("10대입니다.")
elif age < 30:
    print("20대입니다.")
elif age < 40:
    print("30대입니다.")
elif age < 50:
    print("40대입니다.")
else:
    print("50대 이상입니다.")
```

나이를 입력하세요: 35  
30대입니다

# 실습. elif

점수를 입력 받아 점수에 따라 A~F 학점으로 나누기

- A - 90점 이상
- B - 80점 이상 90점 미만
- C - 70점 이상 80점 미만
- D - 60점 이상 70점 미만
- F - 60점 미만

출력 예시

```
점수를 입력하세요: 77  
학점: C
```

# 중첩 조건문

- 하나의 조건문 안에 또 다른 조건문을 포함한 형태
- 더 복잡한 조건을 처리하기 위해서 사용

```
if 조건식1:
    if 조건식 1-1:
        조건식1이 참이면서 조건식 1-1이 참일때 실행되는 코드
    elif 조건식 1-2:
        조건식1이 참이면서 조건식 1-2가 참일때 실행되는 코드
elif 조건식2:
    if 조건식 2-1:
        조건식2가 참이면서 조건식 2-1이 참일때 실행되는 코드
    else:
        조건식2가 참이면서 조건식 2-1이 거짓일때 실행되는 코드
else:
    if 조건식 3-1:
        위 모든 조건이 거짓이면서 조건식 3-1이 참일때 실행되는 코드
```



# 중첩 조건문

- 이전 실습 elif의 코드의 단점 => 음수도 입력이 가능

```
score = int(input("점수를 입력하세요: "))

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



중첩문을 활용하여  
음수입력을 방지

```
score = int(input("점수를 입력하세요: "))

if score >= 0 and score <= 100:
    if score >= 90:
        print("학점: A")
    elif score >= 80:
        print("학점: B")
    elif score >= 70:
        print("학점: C")
    elif score >= 60:
        print("학점: D")
    else:
        print("학점: F")
else:
    print("0에서 100 사이의 값을 입력하세요.")
```

# 실습. 중첩 조건문

- 교통비를 낼 때 나이와 결제 방법에 따른 금액을 출력하세요  
(예. 결제방법이 카드인 경우는 카드만 입력하는 것으로 간주함)

나이	8세 미만	14세 미만	20세 미만	75세 미만	75세 이상
카드	무료	450원	720원	1200원	무료
현금			1000원	1300원	

```
나이를 숫자로 입력해주세요 : 14
결제방법을 입력해주세요 (현금 또는 카드): 카드
14세의 카드 요금은 720원입니다.
```

# 우유를 왜 6개 사왔을까?



## 남편생각

```
milK = 0
avocado = True

if avocado:
|   milK += 6
else:
|   milK += 1

print(f"남편생각: 우유는 {milK}개 사오면됨")
```

# 삼항연산자

- 조건문 같은 경우 한 줄로 줄여서 쓰는 방법도 존재
- 삼항연산자라고 부름
- 간단한 조건을 처리할 때 주로 사용함

참일 때 값 `if` 조건 `else` 거짓일 때 값

```
age = int(input("나이를 입력하세요: "))  
message = "성인입니다." if age > 19 else "미성년자입니다."  
print(message)
```

# pass

- 조건을 만들어서 코드를 생성해야하지만 잠시 아무일도 하지 않고 넘어가야할 경우 사용

```
age = int(input('나이를 숫자로 입력해주세요: '))

if age < 20:
|   pass # 나중에 구현할 코드 자리
elif age < 30:
|   print("20대입니다.")
elif age < 40:
|   pass # 나중에 구현할 코드 자리
elif age < 50:
|   print("40대입니다.")
else:
|   print("50대 이상입니다.")
```

# 조건문에서 in 연산자 활용

- 리스트, 튜플, 딕셔너리 등을 사용하여 특정값이 존재하는지 쉽게 확인가능

```
fruit = input("과일을 한글로 입력하세요. :")

if fruit in ["사과", "바나나", "체리"]:
    print("이 과일은 포함되어 있습니다.")
else:
    print("존재하지 않는 과일입니다.")
```

```
과일을 한글로 입력하세요. :사과
이 과일은 포함되어 있습니다.
```

# 실습. in 연산자 활용

- 과일의 칼로리를 담은 딕셔너리 생성
- 예) apple: 95, banana: 105, cherry: 50
- 사용자로 부터 과일을 입력받아 해당 과일의 칼로리 출력
- 존재하지 않는 과일이라면 포함되지 않았다는 메시지 출력

- 출력 예시

```
과일을 영문으로 입력하세요. :apple  
apple의 칼로리는 95Kcal입니다.
```

## 2. 반복문



- 아래와 같이  $x * 1$ 부터  $x * 100$ 까지 출력 해보겠습니다

```
x = 2
x * 1 = 2
x * 2 = 4
x * 3 = 6
x * 4 = 8
x * 5 = 10
x * 6 = 12
x * 7 = 14
.
..
...
x * 98 = 196
x * 99 = 198
x * 100 = 200
```

```
x = 2
print(f"x * 1 = {x*1}")
print(f"x * 2 = {x*2}")
print(f"x * 3 = {x*3}")
print(f"x * 4 = {x*4}")
print(f"x * 5 = {x*5}")
print(f"x * 6 = {x*6}")
print(f"x * 7 = {x*7}")
...
...
...
print(f"x * 98 = {x*98}")
print(f"x * 99 = {x*99}")
print(f"x * 100 = {x*100}")
```

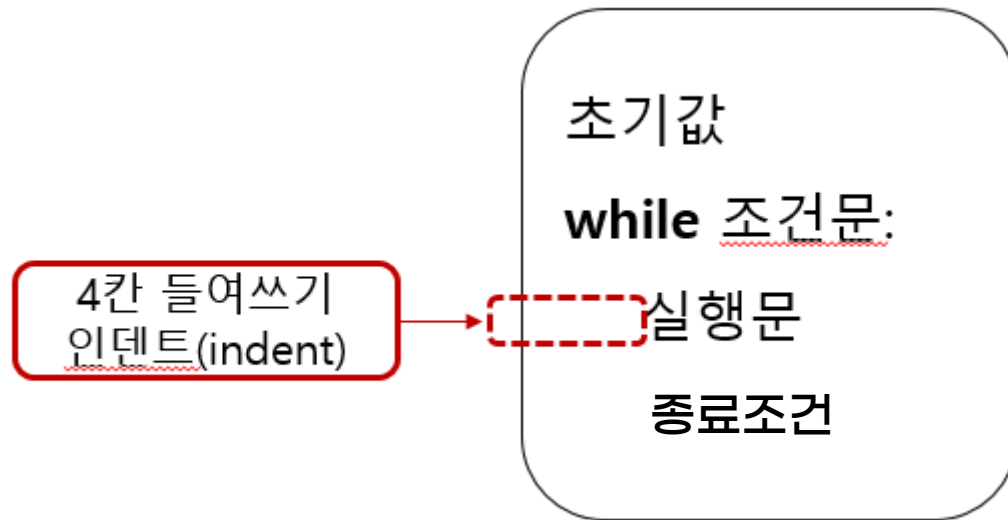
# 반복문

- 컴퓨터의 계산 속도는 사람보다 빠르므로 반복시키는 일을 하기 좋음
- 반복문을 사용하면 코드를 줄이고 똑같은 일을 반복하지 않아도 됨
  - ex) 1부터 100까지 출력, `print(1), print(2), , , , ,`
- 간결하고 효율적인 프로그램을 만드는데 필요함
- 파이썬에는 for문과 while문이 존재
  - while - 특정 조건이 만족되는 동안 반복
  - for문 - 지정한 횟수만큼 반복

# 3. while문

# while문

- while : ~하는 동안, ~하는 동안에
- 조건식이 참이면 반복, 문장이 끝나는 곳에서 조건식으로 돌아와 반복함
- 조건식이 False가 되면 반복문을 종료



```
i = 0
while i < 5:
    print("안녕하세요", i)
    i += 1
print('종료')
# 안녕하세요 0
# 안녕하세요 1
# 안녕하세요 2
# 안녕하세요 3
# 안녕하세요 4
# 종료
```

# while문

- 예제코드

```
# 합계구하기
num = 1
total = 0
while num <= 10:
    total += num
    num += 1
print(f"1부터 10까지의 합은 {total}입니다.")
# 1부터 10까지의 합은 55입니다.

# 입력값 받기
user_input = ""
while user_input != "종료":
    user_input = input("종료하려면 '종료'를 입력하세요: ")
    print(f"입력한 값: {user_input}")
print("프로그램이 종료되었습니다.")
```

# while문 - 무한 루프

- while문에서 무한 루프 (무한 반복)를 사용하는 경우가 있는데 이때 조건식을 항상 참이 되게 하면 됨
- 단, while문을 사용하여 무한 루프를 만들때는 반드시 언제 False가 되는지 확인. 그렇지 않으면 무한 루프에 빠져 프로그램이 종료되지 않게 됨

```
i = 0
while i < 5:
    print("안녕하세요", i)
    # i += 1 종료조건이 누락되어 무한반복됨
print('종료')
```

주의) 위 코드 입력해서 실행하지 마세요

# break문

- 프로그램 종료 없이 반복문 탈출하는 법
- break: 깨다, 부수다 -> 반복문 흐름을 깨다, 부수다

```
while True:
    lunch = input("오늘 점심 후보는? ")
    if lunch == '그만':
        break
    print(lunch + "!")
print("점심 후보 완료")
# 오늘 점심 후보는? 돈까스
# 돈까스!
# 오늘 점심 후보는? 제육
# 제육!
# 오늘 점심 후보는? 짜장면
# 짜장면!
# 오늘 점심 후보는? 그만
# 점심 후보 완료
```

```
count = 0
while True:
    print(count, end=" ")
    count += 1
    if count == 3:
        break

# 0 1 2
```

# continue문

- continue : 계속하다 -> 다음 조건으로 계속 가다
- continue문을 만나면 무조건 블록의 남은 부분을 건너뛰고 반복문의 처음으로 돌아감 -> 처음부터 반복문을 수행한다고 생각하면 됨

```
count = 0

while count < 10:
    count += 1
    if count % 2 != 0:
        continue # 아래 print문 실행안됨
    print(count, end=" ")

# 2 4 6 8 10
```



# 실습. while문 사용하기

- 사용자에게 숫자를 입력받아, 그 숫자가 양수일 경우 1부터 해당 숫자까지의 합을 계산하고 출력하는 프로그램을 작성하세요.
- 다음 조건을 만족해야 함
  1. 양수만 입력 가능. 음수나 숫자가 아닌 값이 입력되면 “양수만 입력 하세요.”라는 메시지를 출력하고 다시 입력을 받습니다.
  2. 0이 입력되면 그 값은 건너뛰고, 다시 입력을 받습니다.
  3. "종료"라고 입력하면 프로그램이 종료됩니다.
  4. 양수가 입력된 경우, 1부터 해당 숫자까지의 합을 계산하여 "1부터 n까지의 합은 X입니다." 형식으로 출력합니다.

# 실습. while문 사용하기

- 출력예시

```
양수를 입력하세요 ('종료' 입력 시 프로그램 종료): 10
1부터 10까지의 합은 55입니다.
양수를 입력하세요 ('종료' 입력 시 프로그램 종료): 30
1부터 30까지의 합은 465입니다.
양수를 입력하세요 ('종료' 입력 시 프로그램 종료): 0
양수를 입력하세요 ('종료' 입력 시 프로그램 종료): 안녕
양수만 입력하세요.
양수를 입력하세요 ('종료' 입력 시 프로그램 종료): -10
양수만 입력하세요.
양수를 입력하세요 ('종료' 입력 시 프로그램 종료): 5
1부터 5까지의 합은 15입니다.
양수를 입력하세요 ('종료' 입력 시 프로그램 종료): 종료
프로그램을 종료합니다.
```

## 4. for문

# for문

- for문은 주어진 시퀀스의 요소를 순회하며 반복실행
- 시퀀스 : 문자열, 리스트, 튜플, range() 등

```
for 변수 in 시퀀스:  
    # 반복 실행할 코드 블록
```

# range 함수

- 연속된 정수를 생성하는 데 사용
- range((start), end, (step))
- start: 시작할 값이며 값 포함됨. 기본값은 0
- end: 끝값이며 값은 포함되지 않음
- step: 간격을 지정하며 기본값은 1

```
for i in range(5):  
|     print(i, end=" ")  
#0부터 5미만까지  
#0 1 2 3 4  
  
for i in range(2, 6):  
|     print(i, end=" ")  
#2부터 6미만까지  
#2 3 4 5  
  
for i in range(1, 10, 2):  
|     print(i, end=" ")  
#1부터 10미만까지 2씩 건너뛰기  
#1 3 5 7 9
```

# 리스트와 for문

- 예제코드

```
# 리스트 반복
fruits = ["사과", "바나나", "체리"]
for fruit in fruits:
    print(fruit, end=" ") # 사과 바나나 체리

# 합계구하기
numbers = [1, 2, 3, 4, 5]
total = 0
for num in numbers:
    total += num
print(f"합계는 {total}입니다.") # 합계는 15입니다.

# 조건문 사용하기
numbers = [1, 3, 5, 7, 9, 11]
for num in numbers:
    if num > 5:
        print(num, end=" ") # 7 9 11
```

# 리스트와 for문

- 리스트 내포 : 리스트 안에 for문을 사용

[표현식 for 요소 in 시퀀스 (if 조건)]

- 표현식 : 각 요소에 대해 계산되어 리스트에 추가될 값
- 요소: 시퀀스(리스트, 튜플, 문자열 등)에서 가져온 각 요소를 나타내는 변수
- if 조건(선택적) : 필터링을 위해 사용되며, 조건이 참(True)인 요소만 포함
- 예시코드

```
# 리스트 내포
squares = [x ** 2 for x in range(1, 11)]
print(squares) # [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

# 조건문과 함께 사용
even_squares = [x ** 2 for x in range(1, 11) if x % 2 == 0]
print(even_squares) # [4, 16, 36, 64, 100]
```

# 딕셔너리와 for문

- 예제코드

```
my_dict = {"name": "홍길동", 1: "a", "city": "서울", "hobby": ["캠핑", "등산"]}

# for문을 사용해 키만 순회
for key in my_dict:
    print(key, end=" ") # name 1 city hobby

# for문을 사용해 값만 순회
for value in my_dict.values():
    print(value, end=" ") # 홍길동 a 서울 ['캠핑', '등산']

# for문을 사용해 키와 값 함께 순회
for key in my_dict.keys():
    print(f"{key}: {my_dict[key]}", end=" ")
for key, value in my_dict.items():
    print(f"{key}: {value}", end=" ")
# name: 홍길동 1: a city: 서울 hobby: ['캠핑', '등산']
```



# 실습. 구구단 만들기

- 구구단 만들기 - 사용자가 입력한 숫자의 구구단 출력
- 출력결과

```
몇단을 출력할까요?: 7
```

```
7 x 1 = 7
```

```
7 x 2 = 14
```

```
7 x 3 = 21
```

```
7 x 4 = 28
```

```
7 x 5 = 35
```

```
7 x 6 = 42
```

```
7 x 7 = 49
```

```
7 x 8 = 56
```

```
7 x 9 = 63
```

# 실습. 정수 합 계산

- 1부터 사용자가 입력한 수까지 홀수의 합 구하기
- 출력결과

```
어디까지 계산할까요?: 5  
1부터 5까지의 홀수의 합: 9
```

# 실습. 평균값 구하기

- 학생1의 국어, 영어, 수학 점수는 각각 83, 92, 88 점이다.
- 학생2의 국어, 영어, 수학 점수는 각각 90, 79, 86 점이다.
- 학생3의 국어, 영어, 수학 점수는 각각 88, 86, 94 점이다.
- 위 학생 3명의 국,영,수 점수에 대한 평균값을 계산하여 출력
- 모든 학생은 하나의 딕셔너리에 선언되어야 함

# 이중 for문

- 반복문 안에 또 다른 반복문
- 중첩 반복문 (외부/내부 반복문)
  - 내부 반복문은 외부 반복문이 한번 반복할 때마다 새롭게 실행

```
for i in range(3): # 외부 반복문
    for j in range(2): # 내부 반복문
        print(f'i: {i}, j:{j}')
    print() # 내부 반복문이 종료될 때마다 실행
```

```
i: 0, j:0
i: 0, j:1

i: 1, j:0
i: 1, j:1

i: 2, j:0
i: 2, j:1
```

# 2차원리스트와 이중 for문

- 예제코드

```
matrix = [  
    [1, 2, 3],  
    [4, 5, 6],  
    [7, 8, 9]  
]  
# 요소의 합계 계산  
total_sum = 0  
for row in matrix:  
    for element in row:  
        total_sum += element  
print(f"전체 합: {total_sum}")  
# 전체 합: 45
```

```
matrix = [  
    [3, 5, 8],  
    [10, 15, 20],  
    [25, 30, 35]  
]  
print("짝수만 출력:", end=" ")  
for row in matrix:  
    for element in row:  
        if element % 2 == 0:  
            print(element, end=" ")  
# 짝수만 출력: 8 10 20 30
```

# 실습. 이중for문 구구단 만들기

- 2단부터 9단까지 구구단 출력
- 출력결과

```
[6 단]
6 x 1 = 6
6 x 2 = 12
6 x 3 = 18
6 x 4 = 24
6 x 5 = 30
6 x 6 = 36
6 x 7 = 42
6 x 8 = 48
6 x 9 = 54
```

```
[7 단]
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
```

# 과제. 자판기 프로그램

아래와 같은 값을 가지는 자판기가 있을 때  
조건에 맞는 프로그램 작성

```
vending_machine = ['게토레이', '게토레이', '레쓰비', '레쓰비', '생수', '생수', '생수', '이프로']
```

```
# 조건1. 사용자는 소비자, 주인 두 가지 종류로 입력받기(번호 또는 값 입력), 그 외의 값은 잘못된 값으로 출력
# 조건2. 소비자일 때 마시고 싶은 음료를 입력받기
# 조건2-1. 값이 있으면 vending_machine에서 제거, 없으면 없음 출력
# 조건3. 주인일 때 추가, 삭제 두 가지 종류 입력 받기 (조건1과 같게)
# 조건3-1. 추가일 때 추가할 음료수 입력받고 vending_machine에 추가 후, 같은 값끼리 연결되게 출력
# 조건3-2. 삭제일 때 삭제할 음료수 입력받고 값이 있으면 제거, 없으면 없음 출력
```

# 과제. 자판기 프로그램

## 출력 예시)

남은 음료수: ['게토레이', '게토레이', '레쓰비', '레쓰비', '생수', '생수', '생수', '이프로']

사용자 종류를 입력하세요:

1. 소비자

2. 주인

1

마시고 싶은 음료? 게토레이

게토레이 드릴게요

남은 음료수: ['게토레이', '레쓰비', '레쓰비', '생수', '생수', '생수', '이프로']

남은 음료수: ['게토레이', '게토레이', '레쓰비', '레쓰비', '생수', '생수', '생수', '이프로']

사용자 종류를 입력하세요:

1. 소비자

2. 주인

2

할 일 선택:

1. 추가

2. 삭제

추가

남은 음료수: ['게토레이', '게토레이', '레쓰비', '레쓰비', '생수', '생수', '생수', '이프로']

추가할 음료수? 레쓰비

추가 완료

남은 음료수: ['게토레이', '게토레이', '레쓰비', '레쓰비', '레쓰비', '생수', '생수', '생수', '이프로']



# 과제. 자판기 프로그램

## 출력 예시)

```
남은 음료수: ['게토레이', '게토레이', '레쓰비', '레쓰비', '생수', '생수', '생수', '이프로']
사용자 종류를 입력하세요:
1. 소비자
2. 주인
주인
할 일 선택:
1. 추가
2. 삭제
2
남은 음료수: ['게토레이', '게토레이', '레쓰비', '레쓰비', '생수', '생수', '생수', '이프로']
삭제할 음료수? 포카리
포카리는 지금 없네요
```

```
남은 음료수: ['게토레이', '게토레이', '레쓰비', '레쓰비', '생수', '생수', '생수', '이프로']
사용자 종류를 입력하세요:
1. 소비자
2. 주인
2
할 일 선택:
1. 추가
2. 삭제
삭제
남은 음료수: ['게토레이', '게토레이', '레쓰비', '레쓰비', '생수', '생수', '생수', '이프로']
삭제할 음료수? 게토레이
삭제 완료
남은 음료수: ['게토레이', '레쓰비', '레쓰비', '생수', '생수', '생수', '이프로']
```

# 학습정리

- if 조건문을 사용하여 코드를 제어할 수 있다.
- if~elif~else를 사용하여 상황에 맞게 코드를 제어할 수 있다.
- 반복문에는 while문과 for문이 존재하며 특정 조건이 만족되게 반복하려면 while문을 사용하고 횟수를 지정하여 반복하게 하려면 for문을 사용한다.
- while문을 사용할 때는 무한 루프에 빠지지 않도록 주의해야한다.

# 다음 수업은?

- 이번수업은 앞으로 파이썬에 사용할 다양한 개념에 대해서 알아보았습니다.
- 다음시간에는 개발의 핵심 함수에 대해서 공부하겠습니다.

복.습.철.저

**수고하셨습니다**