

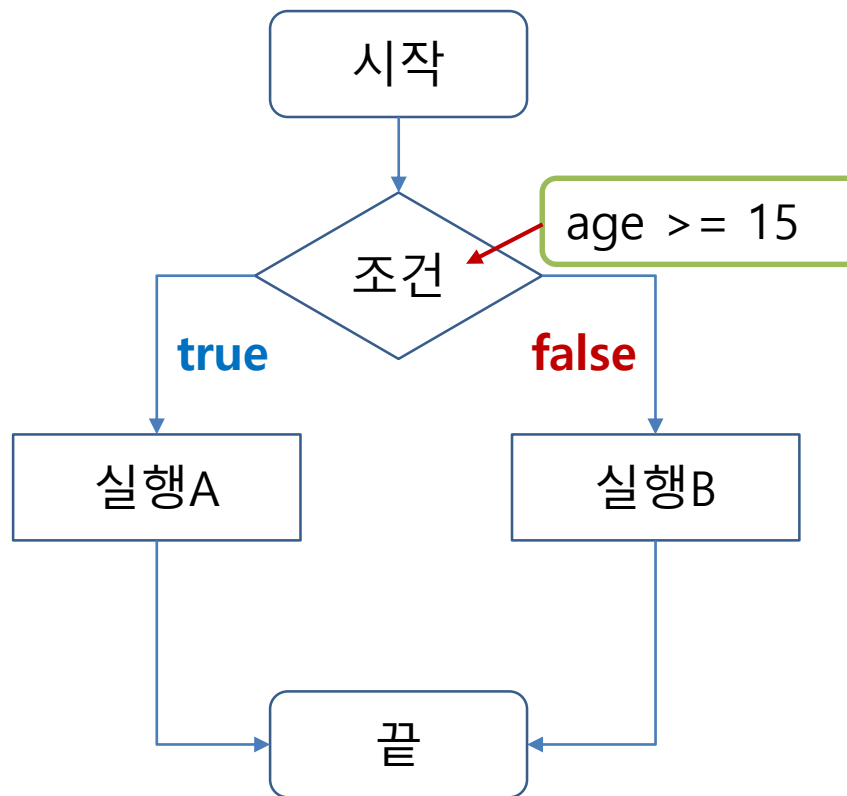
## 2장. 제어문(조건, 반복)



# 조건문(Choice Structure)

## 조건문

- 특정한 조건에 의해서 프로그램 진행이 분기되는 구문



# 조건문 - if

- if 문

**if** 조건식:

수행문

4칸 들여쓰기  
인덴트(indent)

# 조건식이 참이면 수행문 실행

- if ~ else 구문

**if** 조건식:

수행문 1

**else :**

수행문 2

# 조건식이 참이면 수행문1 실행,  
아니면 수행문2 실행

# 조건문 - if

- if 문

```
age = 16
if age >= 15:
    print("관람가")

#print("나이는", age, "세입니다.")
print("나이는 " + str(age) + "세입니다")
```

- if ~ else 구문

```
# 나이가 15세 이상이면 "관람가", 아니면 "관람불가" 출력
if age >= 15:
    print("관람가")
else:
    print("관람불가")
# print("나이는 ", age, "세입니다.")
print(f'나이는 {age}세입니다.')
```

## 조건문 - if

- 좌석의 줄 수를 계산하는 프로그램

입장객 수: 20  
좌석 열 수: 5  
4개의 줄이 필요합니다.

입장객 수: 22  
좌석 열 수: 5  
5개의 줄이 필요합니다.

```
customer = int(input("입장객 수: "))  
column = int(input("좌석 열 수: "))  
row = 0  
  
if customer % column == 0:  
    #row = customer // column  
    row = int(customer / column)  
else:  
    row = int(customer / column) + 1  
  
print(str(row) + "개의 줄이 필요합니다.")
```

# 조건문 - if

- if ~ 내부 if문

수를 입력하세요: 13  
13는(은) 10보다 큰 홀수입니다.

```
# 짝수/홀수 판정 프로그램
num = int(input("수를 입력하세요: "))

if num > 10:
    if num % 2 == 0:
        print(str(num) + "는(은) 10보다 큰 짝수입니다.")
    else:
        print(str(num) + "는(은) 10보다 큰 홀수입니다.")
else:
    if num % 2 == 0:
        print(str(num) + "는(은) 10보다 작은 짝수입니다.")
    else:
        print(str(num) + "는(은) 10보다 작은 홀수입니다.")
```

# 조건문 - if

## ■ 윤년을 판정하는 프로그램

- 4의 배수(4년에 한 번 온다)
  - 100의 배수는 아니다(100년 단위는 윤년이 아님)
  - 400의 배수이다.(400년 단위는 윤년임)
- > 4의 배수이고, 100의 배수는 아니냐(또는), 400의 배수이다.

연도를 입력하세요: 2024  
2024년은 윤년입니다.

연도를 입력하세요: 1900  
1900년은 윤년이 아닙니다.

```
#year = 1900
year = int(input("연도를 입력하세요: "))

if ((year % 4 == 0) and (year % 100 != 0)) or (year % 400 == 0):
    print(f"{year}년은 윤년입니다.")
else:
    print(f"{year}년은 윤년이 아닙니다.")
```

## 다중조건 – if ~elif ~ else

- if ~ elif ~ else구문

if 조건1:

수행문1

elif 조건2:

수행문2

else :

실행문 3

# 조건1이 참이면 수행문1 실행, 조건2가 참이면 수행문2 실행, 조건1,2가 모두 거짓이면 수행문3 실행



## 다중조건 – if ~elif ~ else

### ◆ 놀이 공원 입장료 계산 프로그램

대 상	입장료
취학전 아동	1,000원
초등학생	2,000원
중.고등학생	2,500원
일반인	3,000원

♣ 놀이 공원 입장료 계산 ♣  
나이 입력 : 12  
초등 학생입니다.  
입장료는 2000원 입니다.

## 다중조건 – if ~elif ~ else

### ◆ 놀이 공원 입장료 계산하기

```
print("♣ 놀이 공원 입장료 계산 ♣")

age = int(input("나이 입력 : "))
admission_fee = 0 #입장료

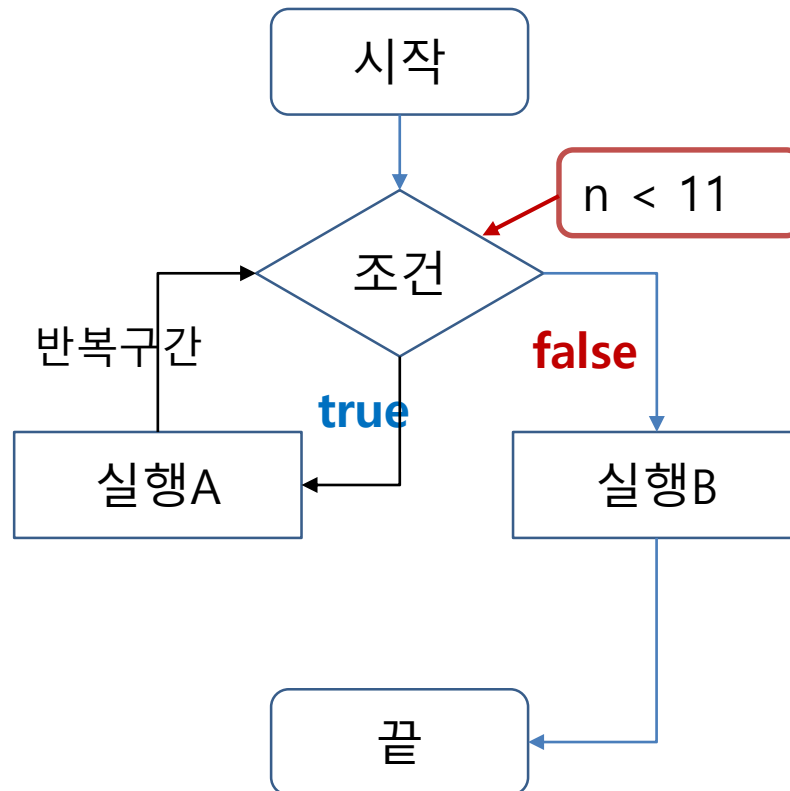
if age < 8:
    print("미취학 아동입니다.")
    admission_fee = 1000
elif age >= 8 and age < 14:
    print("초등 학생입니다.")
    admission_fee = 2000
elif age >= 14 and age < 20:
    print("중,고등 학생입니다.")
    admission_fee = 2500
else:
    print("일반인입니다.")
    admission_fee = 3000

#print("입장료는 " + str(admission_fee) + "원입니다.")
print(f"입장료는 {admission_fee}원 입니다.")
```

# 반복문

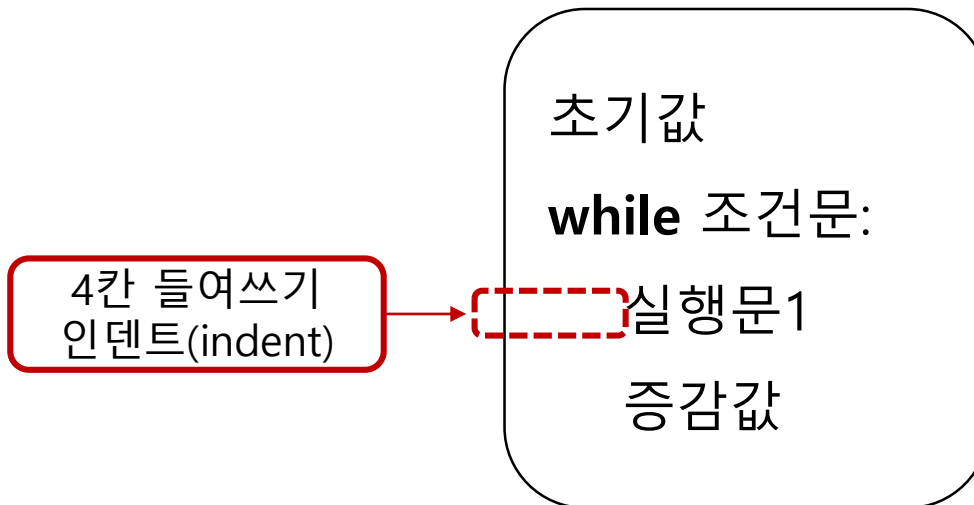
## ● 반복문

- 주어진 조건이 만족할 때까지 실행문을 반복적으로 수행
- while문과 for문이 대표적이다.



# 반복문

- while문



# 반복문 - while

- while문 예제

```
a = 1
print(a)

a += 1
print(a)

a += 1
print(a)
```

```
# "안녕~"을 10번 반복하기
n = 1 # 초기값
while n <= 10:
    print("안녕~", n)
    n += 1 #증가값
print()

n = 10 # 초기값
while n > 0:
    print("안녕~", n)
    n -= 1 #감소값
```

```
안녕~ 1
안녕~ 2
안녕~ 3
안녕~ 4
안녕~ 5
안녕~ 6
안녕~ 7
안녕~ 8
안녕~ 9
안녕~ 10

안녕~ 10
안녕~ 9
안녕~ 8
안녕~ 7
안녕~ 6
안녕~ 5
안녕~ 4
안녕~ 3
안녕~ 2
안녕~ 1
```

# 반복문 - while

- while문 예제

```
# 1부터 10까지 더하기
n = 1
total = 0 #합계
while n <= 10:
    total += n;
    print("n =", n, ", total =", total)
    n += 1
print("합계 : ", total, ", n =", n)
```

```
n = 1 , total = 1
n = 2 , total = 3
n = 3 , total = 6
n = 4 , total = 10
n = 5 , total = 15
n = 6 , total = 21
n = 7 , total = 28
n = 8 , total = 36
n = 9 , total = 45
n = 10 , total = 55
합계 : 55 , n = 11
```

# 반복문 - while

- 반복 조건문

반복문에서 break 문을 만나면 더 이상 반복을 수행하지 않고,  
반복문을 빠져 나옴

```
while True:
```

```
    수행문
```

```
    if 조건 :
```

```
        break
```

# 반복문 - while

- 반복 조건문

```
n = 1
while True:
    if n > 10:
        break
    print(n, end=" ")
    n += 1
print()
```

```
1 2 3 4 5 6 7 8 9 10
합계 : 55
```

```
n = 1
total = 0
while True:
    if n > 10:
        break
    total += n
    n += 1
print("합계 : ", total)
```



# 반복문 - while

- 반복 조건문

```
반복을 계속 할까요?(y/n)
y
반복을 계속 합니다.
반복을 계속 할까요?(y/n)
y
반복을 계속 합니다.
반복을 계속 할까요?(y/n)
k
정상 답변이 아닙니다.
반복을 계속 할까요?(y/n)
n
```

```
while True:
    answer = input('반복을 계속 할까요?(y/n)')

    if answer == 'y' or answer == 'Y':
        print('반복을 계속 합니다.')
    elif answer == 'n' or answer == 'N':
        print('반복을 중단 합니다.')
        break
    else:
        print('정상 답변이 아닙니다.')
```

# 반복문 - while

- 챗봇 프로그램

```
animal = "cat dog horse"  
print("cat" in animal)  
print("dog" in animal)  
print("horse" not in animal)
```

```
사용자: 이름이 뭐야  
챗봇: 저는 Python 챗봇입니다.  
사용자: 안녕  
챗봇: 안녕하세요!. 방가와요  
사용자: 오늘 날씨 어때?  
챗봇: 날씨는 날씨앱이나 검색 기능을 이용하세요  
사용자: 맛집 알려줘  
챗봇: 죄송해요, 잘 이해하지 못했어요.  
사용자: exit  
챗봇: 대화를 종료합니다. _안녕히 가세요!
```

# 반복문 - while

## ● 챗봇 프로그램

```
while True:
    user_input = input("사용자: ")

    if user_input == "exit":
        print("챗봇: 대화를 종료합니다. 안녕히 가세요!")
        break

    elif "안녕" in user_input:
        print("챗봇: 안녕하세요!. 방가와요")

    elif "이름" in user_input:
        print("챗봇: 저는 Python 챗봇입니다.")

    elif "날씨" in user_input:
        print("챗봇: 날씨는 날씨앱이나 검색 기능을 이용하세요")

    else:
        print("챗봇: 죄송해요, 잘 이해하지 못했어요.")
```

## 반복문 - for

- for문

순서열의 각 원소를 처음부터 순회하면서 반복변수에 담아 낸다.

순서열은 리스트, 튜플, 문자열 등을 사용

for - in range() 와 for - in 문을 사용함.

```
for 반복변수 in range(시작값, 종료값, 증감값):  
    실행문
```

```
for 반복변수 in 순서열:  
    실행문
```

## 반복문 - for

- range() 함수 사용하기

**range(시작값, 종료값, 증감값):**

✓ 시작값을 생략하면 0부터 시작하고, 종료값은 (종료값-1) 이다.

```
print(range(1, 10, 1))  
print(list(range(1, 10, 1)))  
print(list(range(1, 10)))  
print(list(range(10)))
```

```
range(1, 10)  
[1, 2, 3, 4, 5, 6, 7, 8, 9]  
[1, 2, 3, 4, 5, 6, 7, 8, 9]  
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

## 반복문 - for

- range() 함수 사용하기

```
# 1부터 10까지 출력하기
for i in range(1, 11):
    print(i, end = ' ')
print()

# 1부터 10까지 홀수 출력하기
for i in range(1, 11, 2):
    print(i)

for i in range(1, 11, 1):
    if i % 2 == 1:
        print(i)
```

```
1 2 3 4 5 6 7 8 9 10
1
3
5
7
9
1
3
5
7
9
```

# 반복문 - for

- range() 함수 사용하기

```
# 1부터 10까지 더하기
hap = 0
for i in range(1, 11):
    hap += i
    print("i =", i, ", hap = ", hap)
print("합계 :", hap)

# 1부터 10까지 짝수의 합계
hap = 0
for i in range(1, 11):
    if i % 2 == 0:
        hap += i
    print("i =", i, ", hap = ", hap)
print("합계 :", hap)
```

```
i = 1 , hap = 1
i = 2 , hap = 3
i = 3 , hap = 6
i = 4 , hap = 10
i = 5 , hap = 15
i = 6 , hap = 21
i = 7 , hap = 28
i = 8 , hap = 36
i = 9 , hap = 45
i = 10 , hap = 55
합계 : 55

i = 1 , hap = 0
i = 2 , hap = 2
i = 3 , hap = 2
i = 4 , hap = 6
i = 5 , hap = 6
i = 6 , hap = 12
i = 7 , hap = 12
i = 8 , hap = 20
i = 9 , hap = 20
i = 10 , hap = 30
합계 : 30
```

# 구구단

- 단을 입력받아 구구단 출력하기

```
단을 입력하세요 : 4  
4 x 1 = 4  
4 x 2 = 8  
4 x 3 = 12  
4 x 4 = 16  
4 x 5 = 20  
4 x 6 = 24  
4 x 7 = 28  
4 x 8 = 32  
4 x 9 = 36
```

```
x = int(input("단을 입력하세요 : "))  
dan = int(x)  
  
for i in range(1, 10):  
    #print(dan, 'x', i, '=', dan*i)  
    print("%d x %d = %d" % (dan, i, dan*i))
```



# continue문

- continue 문

```
for ~ in :  
    if 조건 :  
        continue  
수행문
```

- 반복하다 조건에 맞으면 이후 반복되는 부분을 수행하지 않고 조건식을 수행함.

# continue문

- continue 문

```
# 1 ~ 10 중 5, 7을 제외하고 출력
for i in range(1, 11):
    if i == 5 or i == 7:
        continue
    print(i)

# 1 ~ 10 중 홀수의 합계
hap = 0
for i in range(1, 11):
    if i % 2 == 0:
        continue
    hap += i;
print("홀수의 합 :", hap)
```

```
1
2
3
4
6
8
9
10
홀수의 합 : 25
```

# 반복문

- 이중 for문

- 행, 열 구현하기

```
for i in range(1, 6)
```

```
    for j in range(1, 6)
```

```
        실행문;
```

```
    print()
```

→ 행

→ 열

5행 5열

	열1	열2	열3	열4	열5
행1					
행2					
행3					
행4					
행5					

# 반복문

- 중첩 for문

- 행, 열 구현하기

```
가가가가가  
가가가가가  
가가가가가  
가가가가가  
가가가가가
```

```
for i in range(5):  
    for j in range(5):  
        print('가', end='')  
    print()
```

```
for i in range(1, 6):  
    for j in range(1, 6):  
        print('가', end='')  
    print()
```

## 이중 for문 – 구구단 전체

### ▣ 구구단 전체 출력 프로그램

```
for i in range(2, 10):  
    print('[', i, '단']  
    for j in range(1, 10):  
        print(f"{i} * {j} = {i*j}")
```

```
[ 2 단]  
2 x 1 = 2  
2 x 2 = 4  
2 x 3 = 6  
2 x 4 = 8  
2 x 5 = 10  
2 x 6 = 12  
2 x 7 = 14  
2 x 8 = 16  
2 x 9 = 18
```

```
[ 3 단]  
3 x 1 = 3  
3 x 2 = 6  
3 x 3 = 9  
3 x 4 = 12  
3 x 5 = 15  
3 x 6 = 18  
3 x 7 = 21  
3 x 8 = 24  
3 x 9 = 27
```

## 이중 for문 – 구구단 전체

### ■ 구구단 전체 출력 프로그램

2 x 1 = 2	3 x 1 = 3	4 x 1 = 4	5 x 1 = 5
2 x 2 = 4	3 x 2 = 6	4 x 2 = 8	5 x 2 = 10
2 x 3 = 6	3 x 3 = 9	4 x 3 = 12	5 x 3 = 15
2 x 4 = 8	3 x 4 = 12	4 x 4 = 16	5 x 4 = 20
2 x 5 = 10	3 x 5 = 15	4 x 5 = 20	5 x 5 = 25
2 x 6 = 12	3 x 6 = 18	4 x 6 = 24	5 x 6 = 30
2 x 7 = 14	3 x 7 = 21	4 x 7 = 28	5 x 7 = 35
2 x 8 = 16	3 x 8 = 24	4 x 8 = 32	5 x 8 = 40
2 x 9 = 18	3 x 9 = 27	4 x 9 = 36	5 x 9 = 45

```
for i in range(1, 10):  
    for j in range(2, 10):  
        print(f"{j} * {i} = {j*i}", end=' | ')  
    print()
```

## 2중 for문 - 별 찍기

### ▣ 달리로 삼각형 모양만들기

```
$  
$$  
$$$  
$$$$  
$$$$$
```

```
$$$$$  
$$$$  
$$$  
$$  
$
```

```
for i in range(1, 6):  
    for j in range(1, i+1):  
        print('$', end='')  
    print()  
print()
```

## 2중 for문 - 별 찍기

### ▣ 별로 삼각형 모양만들기

```
      *
     **
    ***
   ****
  *****
 *****
  *****
   ****
    ***
     **
      *
```

```
for i in range(1, 6):
    for j in range(1, 6-i):
        print(' ', end='') #공백
    for j in range(1, i+1):
        print('*', end='') #별
    print()
```



## 2중 for문 - 별 찍기

▣ 1부터 증가하는 숫자 출력

```
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25
```

```
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
21 22 23
```

```
for i in range(0, 5):
    for j in range(1, 6):
        print(5*i+j, end=' ')
    print()
```

# 자리 배치도

## ▣ 자리 배치도(Seat Allocation)

```
*** 자리배치도 ***
입장객 수: 23
좌석 열 수: 5
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
21 22 23
```

```
print("*** 자리배치도 ***")

customer = int(input("입장객 수: "))
col_num = int(input("좌석 열 수: "))
row_num = 0 #좌석 줄 수

if customer % col_num == 0:
    row_num = customer / col_num
else:
    row_num = int(customer / col_num) + 1
#print(str(row_num) + "개의 줄이 필요합니다.")

for i in range(0, row_num):
    for j in range(1, col_num+1):
        seat_num = col_num*i+j
        if seat_num > customer:
            break
        print(seat_num, end=' ')
    print()
```

## 실습 문제 1 - 조건문

### 학점 계산 프로그램

아래의 표를 참고해서 프로그램을 구현해 보세요.

(파일이름: grade.py)

점수	학점
90 ~ 100	A
80 ~ 89	B
70 ~ 79	C
60 ~ 69	D
60 미만	F

👉 실행 결과

점수를 입력하세요: 86  
학점은 B입니다.

## 실습 문제2 – 중첩 for

구구단을 단보다 곱하는 수가 작거나 같은 경우까지 출력하는 프로그램  
(파일이름: test\_gugu.py)

👉 실행 결과

```
[ 2 단]
2 x 1 = 2
2 x 2 = 4

[ 3 단]
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9

[ 4 단]
4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
```