

프로그램 제어문

- 프로그램은 작성 순서대로 실행
- 제어문: 프로그램의 실행 순서를 바꾸는 문장
 - 조건문
 - if, if-else, if-elif-else
 - 반복문
 - while
 - for
 - 기타 제어문
 - break
 - continue

if 제어문

조건문

■ if 문

- 조건을 만족할 때 정해진 문장 블록을 실행

if 문의 기본 구조

```
if Condition(조건) :  
    Block(블록)
```

Condition(조건)이 참이면 : 다음의 Block(블록) 부분을 실행함

- 조건이 참(True)이면 블록을 실행하고 아니면 블록을 건너뛴다

```
score = int(input("정수 입력:"))  
if score >= 90 : #if문 끝에 :을 붙인다  
    print("성적 : A") #들여쓰기  
    print("장학금 수여")
```

점수가 90점 이상이면
장학금 수여

```
age_0 = 22  
age_1 = 18  
age = int(input('your age? '))  
if age < age_0 and age > age_1: #복합 조건문  
    print("당신의 나이는 18~22세 사이입니다")
```

18 < 나이 < 22 인지 검사

조건문

■ if-else 문

if-else 문의 기본 구조

```
if Condition(조건) :  
    Block1(블록1)  
else :  
    Block2(블록2)
```

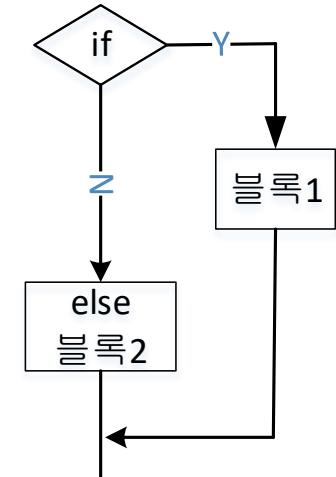
- 조건이 참(True)이면 Block1을 실행하고 else: 는 무시
- 참(True)이 아니면 else: 다음 Block2를 실행

```
if score >= 90 :  
    print("pass")  
else : #else도 끝에 :을 붙인다  
    print("fail") #들여쓰기
```

점수가 90점 이상이면
pass, 아니면 fail

- 한 줄로 표현하기

```
res = 'pass' if score >= 90 else 'fail'  
print(res)
```



조건문 예제

- 평년과 윤년 구분하는 프로그램
 - 년도가 4로 나뉘어지고 100으로 나뉘어지지 않거나
 - 400으로 나뉘어지면 윤년이다

```
year = int(input("Type a year :"))
if (year%4 == 0 and year%100 != 0) or (year%400) == 0 :
    print(year, "is a leap year")
else :
    print(year, "is not a leap year")
```

조건문

■ 중첩 if 문- if 문 속에 또 다른 if 문

```
n = int(input("type int number : "))  
if n > 0 :  
    print("positive")  
else :  
    if n < 0 :  
        print("negative")  
    else :  
        print("0")
```

수의 판별: 양수, 음수, 0

```
n = int(input("type int number : "))  
if n >= 0 :  
    if n == 0 :  
        print("0")  
    else :  
        print("positive")  
else :  
    print("negative")
```

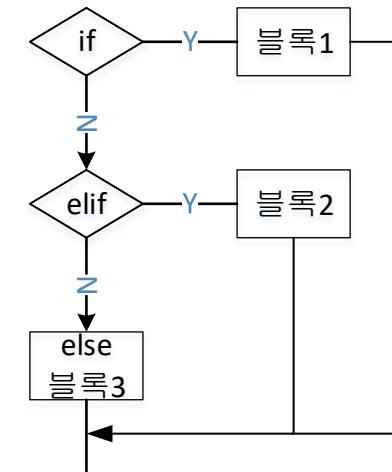
조건문

■ if-elif-else 문

- if 조건1이 참이면 블록1, if 조건1이 거짓이고 elif 조건2이 참이면 블록2, 조건이 모두 거짓이면 블록3이 실행된다

if-elif-else 문의 기본 구조

```
if Condition1(조건1) :  
    Block1(블록1)  
elif Condition2(조건2) :  
    Block2(블록2)  
else :  
    Block3(블록3)
```



```
n = int(input("type int number : "))  
if n > 0 :  
    print("positive")  
elif n < 0 :  
    print("negative")  
else :  
    print("0")
```

수의 판별: 양수, 음수, 0

- 다중 elif 문

- 여러 개의 elif를 사용한 문장
- if 또는 elif 조건 중 하나만 충족되면 나머지는 무시

```
age = 12
if age < 4:
    price = 0
elif age < 18:
    price = 5
elif age < 65:
    price = 10
else:
    price = 5

print("Your admission cost is $" + str(price) + ".")
```

입장료 계산
프로그램
4세 미만: 무료
4 = < 나이 < 18:
5불
65 > 나이: 10불
나머지: 5불

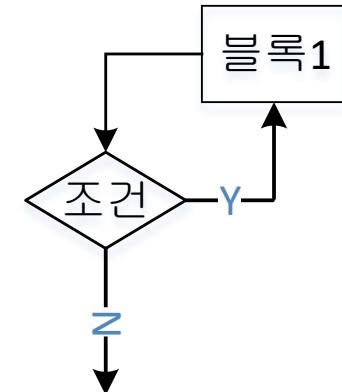
while 반복문

반복문

- 조건이 충족되면 동일한 일을 반복 수행
- while 문

while 문의 기본 구조

```
[초기화]  
while [조건] :  
    [블록]  
    [다음문장]
```



- 조건이 만족되면 블록을 실행하고 다시 조건을 조사
- 조건이 만족되지 않으면 다음 문장 실행

```
i = 1      #초기화  
sum = 0  
while i <= 100 :      #조건 판단. 끝에 :를 붙인다  
    sum = sum + i    #블록  
    i = i + 1 #조건 변경  
  
print("sum=", sum)    #블록이 끝나면 다시 들여쓰기를 원래 위치로
```

1~100까지의 합을
구하는 프로그램

while 연습문제

- 10진수를 입력 받아 2진수로 변환하여 출력하는 프로그램

```
n = int(input('Number: '))
result = " " #변환 결과 저장

#10진수를 2로 나누어 몫이 0이 될 때까지 반복해서 나머지를 구한다
while n != 0:
    m = n % 2 #나머지
    result = str(m) + result #숫자를 문자열로 변환
    n = n // 2 #몫

print("2진수 =", result)
```



for 제어문

반복문

■ for 문

```
for [변수] in [반복객체(리스트, 튜플, 문자열 등)] :  
    [블록]
```

- [반복 객체]의 각 요소의 값을 하나씩 [변수]에 저장하고 블록을 실행
- [반복 객체]에 더 이상의 요소가 없을 때까지 반복

```
for i in [1, 2, 3] :    # 끝에 :를 붙인다  
    print("숫자", i)
```

- 튜플, 문자열과 같은 반복 객체를 사용할 수도 있다.

```
for i in (1, 2, 3):  
    print("숫자", i)
```

```
for i in 'hello':  
    print(i)
```

반복문

■ for-range 문

for 문의 기본 구조 : range() 함수를 사용할 때

```
for [변수] in range(start, stop, step) :  
    [블록]
```

- range() 함수는 반복 객체를 생성하는 함수
- 변수 값을 start부터 시작하여 step 만큼 증가시키며 stop될 때까지 실행.
step=1이면 생략

```
for i in range(1, 4, 1) : #1부터 시작하여 1씩 증가하여 4가 되면 종료  
    print("Number", i)
```

Number 1
Number 2
Number 3

반복문

- for 문을 이용한 1부터 100까지의 합 구하기

```
sum = 0
for i in range(1, 101, 1) :
    sum = sum + i

print("sum = ", sum)
```

- x^n 을 구하는 프로그램

```
x = float(input('Type x : '))
n = int(input('Type n : '))

prod = 1
for i in range(1, n+1):
    prod = prod * x

print(prod)
```

반복문

▪ 중첩 반복문

- 내부 for 문이 먼저 실행되고 끝나면 외부 for 문이 실행됨
- 구구단 프로그램

```
for i in range(2, 10, 1) :  
    for j in range(1, 10, 1):  
        print("%d x %d = %d" %(i, j, i*j))
```

▪ 무한 반복문

```
while True :    #조건이 항상 참이다  
    print("last forever")
```

- 무한 반복문을 중지하려면 ctrl-C를 누른다

제어문 예제(반복문 안의 조건문)

■ 숫자 맞추기 게임 프로그램

- 1~100 사이의 난수를 만들고 사용자가 1~100 사이의 수를 입력하여 맞추는 프로그램. 5회까지 맞추지 못하면 종료

```
from random import randint
secret_num = randint(1,100)
num_guesses = 0    # 시도 횟수
guess = 0          # 예상 숫자
while guess != secret_num and num_guesses <= 4:
    guess = eval(input('Enter your guess (1-100): '))    # 수 입력
    num_guesses = num_guesses + 1    # 시도 횟수 증가
    if guess < secret_num:    # 정답이 클 때
        print('더 큽니다.', 5-num_guesses, '회 남았습니다.\n')
    elif guess > secret_num:    # 정답이 작을 때
        print('더 작습니다.', 5-num_guesses, '회 남았습니다.\n')
    else:    # 정답
        print('맞았습니다!')
if num_guesses==5 and guess != secret_num:    # 맞추지 못한 경우
    print('당신이 졌습니다. 정답은 ', secret_num, '입니다')
```

기타 제어문

기타 제어문

- break 문 : 반복문을 빠져나올 때 사용

```
n = 10
while n >= -10 :
    if n == 0 :
        break #n=0이면 while 블록을 빠져나온다
    inv = 1.0 / n
    print(inv)
    n = n - 1
```

- continue 문 : 블록의 나머지 부분은 생략하고 다시 while 조건 조사

```
n = 10
while n >= -10 :
    if n == 0 :
        n = n - 1
        continue #n=0이면 while 블록의 나머지 부분을 건너 뛴다
    inv = 1.0 / n
    print("n = %d, inv(n) = %0.2f"%(n, inv))
    n = n-1
```