

Python Programming

For Beginner

Decision

For Python 3

if 절

1. 구문 형식

- If [조건식] :
- 조건식의 결과는 참 또는 거짓

2. 조건식의 유형

- 관계 연산자로 구성된 식
- 문자열 조건

#관계 연산

```
if people < cats:  
    print("고양이가 지배하는 세상")
```

```
if people > cats:  
    print("사람이 지배하는 세상")
```

```
if people < dogs:  
    print("세상은 개의 침으로 침몰!")
```

#문자열 조건

```
if bool("고양이"):  
    print("문자열 값이 있으면 True")
```

```
if "멍멍이":  
    print("문자열 값이 있으면 True")
```

```
if "":  
    print("문자열 값이 빈값이면 False, 실행안됨")
```

else와 if

1. 형식 1

```
If (조건식) :  
    구문 ...  
else :
```

2. 형식 2

```
If (조건식) :  
    구문 ...  
elif :  
    구문 ...  
else :  
    구문 ....
```

3. elif

- 다른 언어의 경우, else if
- 대안 실행을 위한 평가

#if ~ elif ~ else 문

```
people = 30
```

```
cars = 40
```

```
trucks = 15
```

```
if cars > people:
```

```
    print("차를 타야 합니다.")
```

```
elif cars < people:
```

```
    print("모두 다 차를 탈 수 없습니다.")
```

```
else:
```

```
    print("결정하지 못하겠습니다.")
```

```
if trucks > cars:
```

```
    print("트럭이 너무 많아요.")
```

```
elif trucks < cars:
```

```
    print("트럭을 더 구매해야 할까요.")
```

```
else:
```

```
    print("여전히 결정이 어렵네요.")
```

중첩 if 문

1. 한 분기에서 다른 분기로 가지치기
2. 선택지가 더 필요하다면, elif 추가.

```
if door == "1":
    print("선택하세요?")
    print("1. 땡.")
    print("2. 하늘.")
    bear = input("> ")
    if bear == "1":
        print("곰이 당신을 태웠습니다. 잘 했어요!")
    elif bear == "2":
        print("곰이 당신을 들이받아 하늘로 보냈네요. 저런!")
    else:
        print(f"음, {bear}은 선택할 수 없어요.")

elif door == "2":
    print("토끼굴로 들어왔어요.")
    print("1. 모자를 씩니다.")
    print("2. 노란 자켓을 입습니다.")
    print("3. 토끼를 따라 갑니다.")
    oz = input("> ")
    if oz == "1" or oz == "2":
        print("이상한 나라의 일원이 되었습니다. 잘했습니다!")
    else:
        print("다시 살던 곳으로 갑니다. 기회를 잃었습니다!")
else:
    print("이해력이 떨어지는 군요!")
```

if문 모범 사례

1. 모든 if 문에는 else를 꼭 쓴다.
2. else 문이 결코 실행되지 않을 것 같다면
 - else 내에 사용자 지정 프로그램 종료 함수 호출
3. if문은 2단계 이하로만 중첩
4. if문을 문단처럼 다룬다.
5. 부울 값 검사는 단순하게.

```
die(msg)
    print(msg, "잘 했어요")
    exit(0)
#msg는 die() 호출 이유 전달
```

Loop

For Python 3

for 문

1. 형식

- for in 문 형식만 가능

For 변수 in 반복 가능한 객체:
... 반복 실행 구문 ...

2. 반복 가능한 객체

- 리스트
- 딕셔너리
- 집합
- 튜플
- 바이트(bytes)
- range

```
#for in 반복문
the_count = [1, 2, 3, 4, 5]
fruits = ['사과', '오렌지', '배', '살구']
change = [1, '십원', 2, '백원', 3, '오백원']
```

```
# 리스트를 사용하는 for-in 문
for number in the_count:
    print(f"이번 회수 숫자는 {number}")
```

```
# 앞서와 동일
for fruit in fruits:
    print(f"과일 종류: {fruit}")
```

```
# 다른 데이터 형식이 혼합된 리스트
for i in change:
    print(f"받은 돈: {i}")
```


For와 range

1. 형식 – range(시작 값, 종료 값, 스텝)
2. 리스트를 조각 낸 것과 비슷하다.
3. 시작 값과 종료 값 바로 전 단계 숫자까지 포함
4. 값 확인은 다른 순서 있는 컬렉션으로 변환

```
##range 값 출력하기
for i in range(10,20,3):
    print(i)
```

```
range(10,20,2)
list(range(10,20,2))
tuple(range(10,20,2))
```

```
## 빈 리스트로 시작해서 리스트 만들기
elements = []
```

```
# 0-5까지 세는 range 함수 사용
for i in range(0, 6):
    print(f"리스트에 {i}값을 추가합니다.")
    # append는 리스트의 함수 입니다.
    elements.append(i)
```

```
# 역시 출력할 수 있습니다.
for i in elements:
    print(f"원소 {i} 번째 값: {i}")
```

```
##range 값 출력하기
for i in range(10,20,3):
    print(i)
```

for 와 enumerate

1. 몇 번째 반복문인지 확인이 필요할 때
2. 인덱스 번호와 원소를 tuple로 반환

#기본 tuple 반환

```
e = [1, 5, 7, 33, 39, 52]
```

```
for t in enumerate(e):  
    print(t)
```

#활용

```
for i, v in enumerate(e):  
    print("인덱스 : {}, 값: {}".format(i,v))
```

While 문

1. 형식

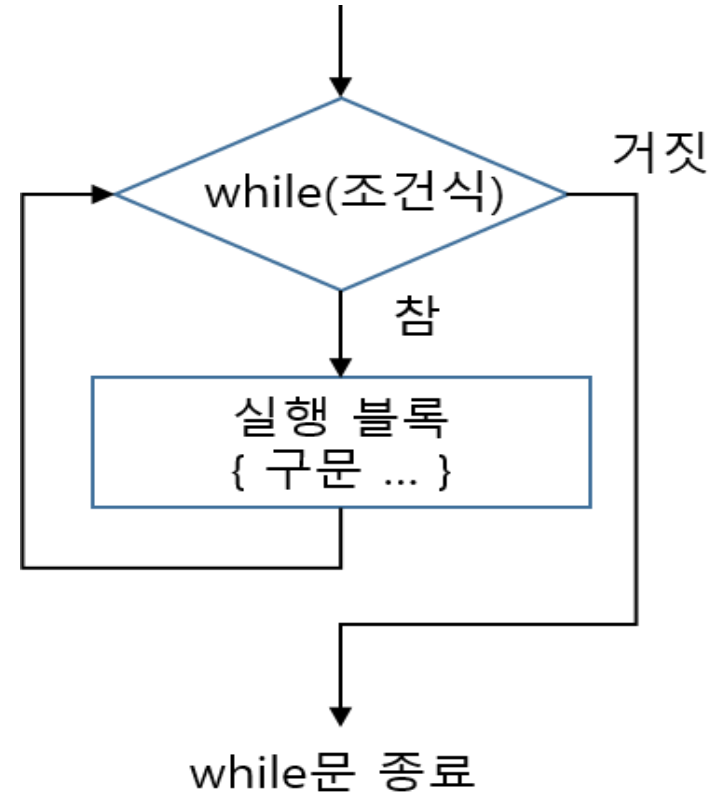
```
while 조건:  
    ... 반복 실행 구문 ...
```

2. 조건이 참인 동안 반복 수행

- 조건 제어는 while 문 내에서 처리

3. for문과 while 문 선택 기준

- 반복 횟수가 정해지지 않을 때
- 반복 횟수가 정해진 경우



While 문 모범 사례

1. For 문으로 가능한지 먼저 검토
2. 조건이 언제 거짓이 되는지 확인
3. 의심스러울 땐 순환 블록 위 아래
검사 변수 출력
4. 무한 루프 조심
 - CTRL + C로 종료

```
i = 0
numbers = []

while i < 6:
    print(f"반복문 상단에서 i는 {i}")
    numbers.append(i)

    i = i + 1
    print("이제 숫자는: ", numbers)
    print(f"반복문 하단에서 i는 {i}")
```

break와 continue

1. break 문

- 특정 조건에 반복 문 중단 시키고 빠져 나올 때 사용

2. continue 문

- 반복문을 중단 시키지 않고 다음 반복으로 넘어 간다.

#break 문

```
while True:
    response = input('숫자를입력하세요:')
    if int(response) % 10 == 0 :
        print('10으로 나누었을때 나머지가 0입니다.')
        break
```

#continue 문

```
i=0
j=0
while True:
    j=j+1
    response = input(f"숫자를 입력하세요(입력횟수[{j}]):")
    result = int(response) % 10
    if result == 0 :
        continue
    i = i + 1
    print("10으로 나눈 나머지는 {}".format(result))
    print(f"반복횟수: {i}")
```

수고했습니다!