

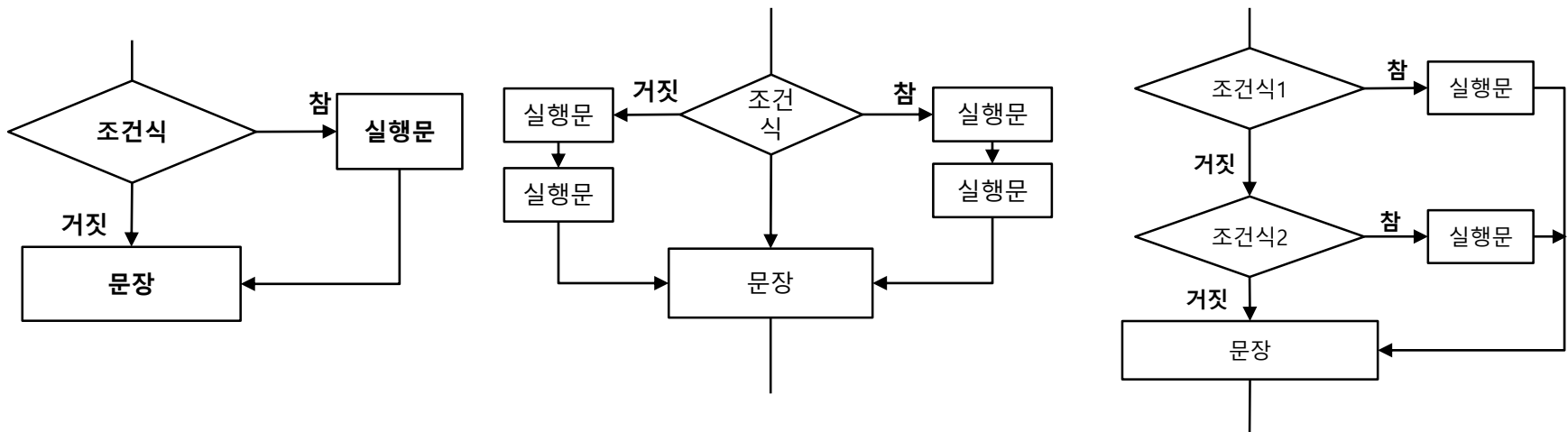
Python 제어구문



제어-선택

- 조건문

- 프로그램의 실행 흐름을 바꾸고자 할 때 사용하는 제어문
- 사용자가 지정한 조건에 따라 실행할 문장을 결정함



조건문 : if

- if 조건문 뒤에는 반드시 콜론(:)
 - 바로 아래 문장부터 if문에 속하는 모든 문장에 들여쓰기(indentation)를 해야 함
- 파이썬에는 다른 언어에 있는 switch 문이 존재하지 않음
 - if...elif...elif... 문으로 수행
- pass
 - def 문이나 if 문처럼 코드 블록을 본문으로 갖는 표현에서 본문을 비워 둘 때 사용

```
#조건식
x = 10

print("조건식 1:")
if x % 2 == 1 :
    print(x, ": 홀수")

print("조건식 2:")
if x % 2 == 0 :
    print(x, ": 짝수")
else:
    print(x, ": 홀수")

print("조건식 3:")
if x <= 0 :
    print("양수가 아님")
elif x % 2 == 0 :
    print(x, ": 짝수")
else:
    print(x, ": 홀수")
```

```
조건식 1:
조건식 2:
10 : 짝수
조건식 3:
10 : 짝수
```

비교연산자/논리연산자

비교연산자	의미
<code>x < y</code>	x가 y보다 작다.
<code>x > y</code>	x가 y보다 크다.
<code>x == y</code>	x와 y가 같다.
<code>x != y</code>	x와 y가 같지 않다.
<code>x >= y</code>	x가 y보다 크거나 같다.
<code>x <= y</code>	x가 y보다 작거나 같다.

논리연산자	의미
<code>and</code>	x가 y보다 작다.
<code>or</code>	x가 y보다 크다.
<code>not</code>	x와 y가 같다.

#비교 연산자

```
x = 10
y = 20
```

```
print("비교 연산자")
print(x, '>', y, ":", x > y)
print(x, '<', y, ":", x < y)
print(x, '>=', y, ":", x >= y)
print(x, '<=', y, ":", x <= y)
print(x, '==', y, ":", x == y)
print(x, '!=', y, ":", x != y)
```

```
print("논리 연산자")
if ( x >= 20 and y >= 20 ) : print ("두수는 20보다 크다")
elif ( x >= 20 or y >= 20 ) : print ("두수 중 20보다 큰수가 있다.")
else : print ("두수는 20보다 작다")
```

비교 연산자

```
10 > 20 : False
10 < 20 : True
10 >= 20 : False
10 <= 20 : True
10 == 20 : False
10 != 20 : True
```

논리 연산자

```
두수 중 20보다 큰수가 있다.
```

in연산자

- 멤버십 연산자

- 문자열이나 리스트나 튜플과 같이 연속적인 자료구조에 속한 멤버를 확인하기 위한 연산자

```
#in 연산자

str = "안녕하세요"
print(str, "에 안이란 글자가 있습니다.:", '안' in str)

#리스트
lst = list(range(1,5))
print("리스트에 10이 있습니다.:", 10 in lst)
```

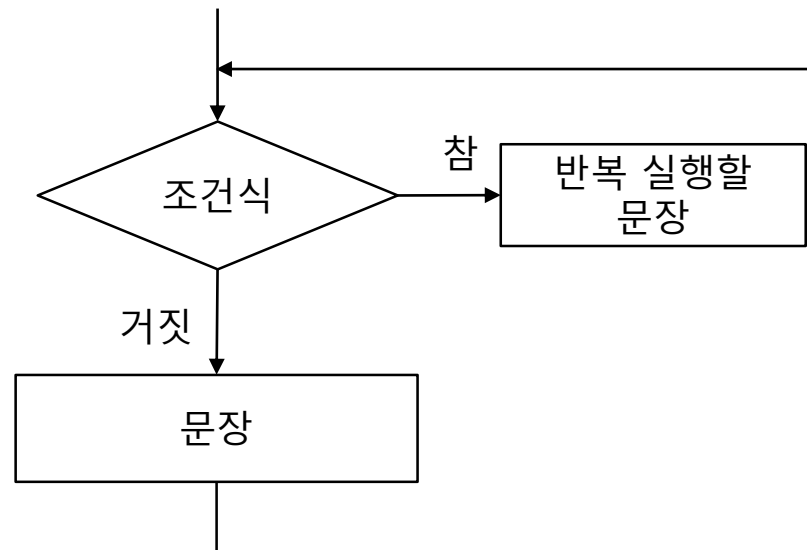
안녕하세요 에 안이란 글자가 있습니다.: True
리스트에 10이 있습니다.: False



제어-반복

- 반복문

- 특정한 부분의 코드가 반복적으로 수행



반복문 - for

- for

- 컬렉션으로부터 하나씩 요소(element)를 가져와, 루프 내의 문장들을 실행

```
#for문으로 collection 접근하기
#문자열
str = "안녕하세요"

print("문자열")
for item in str :
    print(item)

#리스트
lst = list(range(1,5))

print("리스트")
for item in lst :
    print(item)

#딕셔너리
dic = {'a':1, 'b':2}

print("딕셔너리")
for item in dic :
    print(item)

for item in dic.items():
    print(item)
```

문자열
안
녕
하
세
요
리스트
1
2
3
4
딕셔너리
b
a
('b', 2)
('a', 1)



range() 함수와 반복문

- range() 함수

- range(시작숫자, 종료숫자, step)
- 결과는 시작숫자부터 종료숫자 바로 앞 숫자까지 컬렉션 생성
- 값을 확인하기 위해서는 순서가 있는 리스트나 튜플 컬렉션으로 변환해야 함
- 시작숫자를 생략하면 0부터 생성
- step을 생략하면 1씩 증가

```
xlist = range(10)
print(type(xlist))
print(xlist)
```

```
for item in xlist:
    print(item, end = ' ')
```

```
print()
```

```
print('-'*50)
```

```
xlist = list(range(10))
print(type(xlist))
print(xlist)
```

```
for item in xlist:
    print(item, end = ' ')
```

```
<class 'range'>
range(0, 10)
0 1 2 3 4 5 6 7 8 9
```

```
-----
<class 'list'>
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
0 1 2 3 4 5 6 7 8 9
```


For문을 이용한 리스트 변형

```
: st = ['1','2','3','4']  
print("리스트 :", lst)  
  
#for문을 이용한 리스트 항목의 형 변환  
lst = [int(item) for item in lst]  
print("리스트 :", lst)  
  
#for문을 이용한 리스트 항목추출  
lst = [item for item in lst if item % 2 == 0 ]  
print("리스트 :", lst)
```

반복문 while

- while

- while 키워드 다음의 조건식이 참일 경우 계속 while 안의 블록을 실행

```
: #for문과 while문  
  
print("for 문")  
for i in range(1, 6):  
    print(i)  
  
print("while 문")  
i = 0  
while i < 5:  
    i = i + 1  
    print(i)
```



break/continue

- break 문

- 반복문 안에서 루프를 빠져나오기 위해 을 사용

- continue문

- 루프 블록의 나머지 문장들을 실행하지 않고 다음 루프로 직접 돌아가게 함

```
#break/continue
for i in range(1, 100) :
    if ( i % 2 == 1 ) : continue
    if ( i > 5 ) : break
    print(i)
```

2
4



해결문제

- 단을 입력 받아서 해당하는 단의 구구단을 출력하시오.
 - 단은 2단에서 9단까지만 입력 그 외 입력되면 종료하고 그렇지 않을 경우 계속 입력

단을 입력하세요6

**** 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

단을 입력하세요2

**** 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

단을 입력하세요99
입력오류

Co



해결문제2

- 기상청에서 제공하는 지역별 식중독 지수를 참고하여 해당 지역의 단계를 설정하고 대응요령을 예시와 같이 출력하시오.

— https://www.weather.go.kr/weather/lifenindustry/jisudaymap_A01_2.jsp

서울시 : 주의 (57)

=> 조리음식은 중심부까지 75℃어패류 85℃로 1분 이상 완전히 익히고 외부로 운반할 때에는 가급적 아이스박스 등을 이용하여 10℃이하에서 보관 및 운반

부산시 : 주의 (62)

=> 조리음식은 중심부까지 75℃어패류 85℃로 1분 이상 완전히 익히고 외부로 운반할 때에는 가급적 아이스박스 등을 이용하여 10℃이하에서 보관 및 운반

대구시 : 주의 (59)

=> 조리음식은 중심부까지 75℃어패류 85℃로 1분 이상 완전히 익히고 외부로 운반할 때에는 가급적 아이스박스 등을 이용하여 10℃이하에서 보관 및 운반

인천시 : 주의 (59)

=> 조리음식은 중심부까지 75℃어패류 85℃로 1분 이상 완전히 익히고 외부로 운반할 때에는 가급적 아이스박스 등을 이용하여 10℃이하에서 보관 및 운반

강화군 : 주의 (58)

=> 조리음식은 중심부까지 75℃어패류 85℃로 1분 이상 완전히 익히고 외부로 운반할 때에는 가급적 아이스박스 등을 이용하여 10℃이하에서 보관 및 운반

서해5도 : 주의 (64)

=> 조리음식은 중심부까지 75℃어패류 85℃로 1분 이상 완전히 익히고 외부로 운반할 때에는 가급적 아이스박스 등을 이용하여 10℃이하에서 보관 및 운반

광주시 : 주의 (62)

=> 조리음식은 중심부까지 75℃어패류 85℃로 1분 이상 완전히 익히고 외부로 운반할 때에는 가급적 아이스박스 등을 이용하여 10℃이하에서 보관 및 운반

대전시 : 주의 (60)

=> 조리음식은 중심부까지 75℃어패류 85℃로 1분 이상 완전히 익히고 외부로 운반할 때에는 가급적 아이스박스 등을 이용하여 10℃이하에서 보관 및 운반

울산시 : 관심 (53)

=> 화장실 사용 후, 귀가 후, 조리 전에 손 씻기를 생활화

세종시 : 주의 (58)

=> 조리음식은 중심부까지 75℃어패류 85℃로 1분 이상 완전히 익히고 외부로 운반할 때에는 가급적 아이스박스 등을 이용하여 10℃이하에서 보관 및 운반