

빠르게 활용하는 파이썬3 프로그래밍

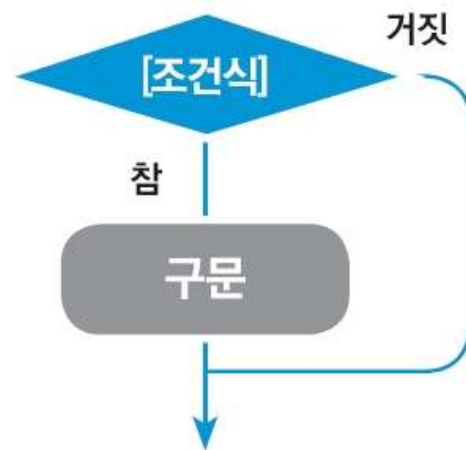
4. 제어



목 차

- if, elif, else
- 조건식의 참/거짓 판단
- 단축 평가
- while 문
- for 문
- break, continue, else

if 문



```
if 조건문:  
    수행할 문장1  
    수행할 문장2  
    ...  
else:  
    수행할 문장A  
    수행할 문장B  
    ...
```

- 조건식을 평가하고 참인 경우만 구문이 수행
- 2개 이상의 구문은 들여쓰기로 블록을 지정
 - 함수와 동일
 - 들여쓰기의 정도는 파일 전체를 통틀어 일치해야 함

※ 조건문 다음에 콜론(:)을 잊지 말자!



if 문

- 조건문이란 무엇인가?
 - 참과 거짓을 판단하는 문장

자료형	참	거짓
숫자	0이 아닌 숫자	0
문자열	"abc"	""
리스트	[1,2,3]	[]
터플	(1,2,3)	()
딕셔너리	{"a":"b"}	{}

```
>>> money=0
>>> if money:
    print("참입니다.")
```

```
>>> money=1
>>> if money:
    print("참입니다.")
```

참입니다.



if 문

- 비교연산자

비교연산자	설명
$x < y$	x가 y보다 작다
$x > y$	x가 y보다 크다
$x == y$	x와 y가 같다
$x != y$	x와 y가 같지 않다
$x \geq y$	x가 y보다 크거나 같다
$x \leq y$	x가 y보다 작거나 같다

```
>>> x = 3
>>> y = 2
>>> x > y
True
>>>
```

```
>>> x < y
False
```

```
>>> x == y
False
```

```
>>> x != y
True
```



if 문

- 예제 1
 - value 값은 10이고, 조건문은 value 값이 5보다 크면 "value is bigger than 5"라는 문자열을 출력합니다.

```
>>> value = 10
>>> if value > 5:
    print ("value is bigger than 5")
```

```
value is bigger than 5
>>> |
```



if 문

- 예제 2

- money 값은 2000이고, 조건문은 만약 3000원 이상의 돈을 갖고 있으면 "택시를 타세요" 그렇지 않으면 "걸어가세요"라는 문자열을 출력합니다.

```
>>> money = 2000
>>> if money >= 3000:
...     print("택시를 타고 가라")
... else:
...     print("걸어가라")
...
걸어가라
>>>
```




if 문

- and, or, not

연산자	설명
x or y	x와 y 둘중에 하나만 참이면 참이다
x and y	x와 y 모두 참이어야 참이다
not x	x가 거짓이면 참이다

- 예제 3
 - "돈이 3000원 이상 있거나 카드가 있다면 택시를 타고 그렇지 않으면 걸어 가라"

```
>>> money = 2000
>>> card = 1
>>> if money >= 3000 or card:
...     print("택시를 타고 가라")
... else:
...     print("걸어가라")
...
택시를 타고 가라
>>>
```




if 문

- `x in s`, `x not in s`

in	not in
<code>x in 리스트</code>	<code>x not in 리스트</code>
<code>x in 튜플</code>	<code>x not in 튜플</code>
<code>x in 문자열</code>	<code>x not in 문자열</code>

```
>>> 1 in [1, 2, 3]
True
>>> 1 not in [1, 2, 3]
False
```

```
>>> 'a' in ('a', 'b', 'c')
True
>>> 'j' not in 'python'
True
```



if 문

- `x in s`, `x not in s`
- 예제 4
 - "만약 주머니에 돈이 있으면 택시를 타고, 없으면 걸어 가라"

```
>>> pocket = ['paper', 'cellphone', 'money']
>>> if 'money' in pocket:
...     print("택시를 타고 가라")
... else:
...     print("걸어가라")
...
택시를 타고 가라
>>>
```



if 문

- `x in s`, `x not in s`
 - 조건문에서 아무 일도 하지 않게 설정하고 싶다면? **pass**
- 예제 5
 - "주머니에 돈이 있으면 가만히 있고 주머니에 돈이 없으면 카드를 꺼내라"

```
>>> pocket = ['paper', 'money', 'cellphone']
>>> if 'money' in pocket:
...     pass
... else:
...     print("카드를 꺼내라")
...
```

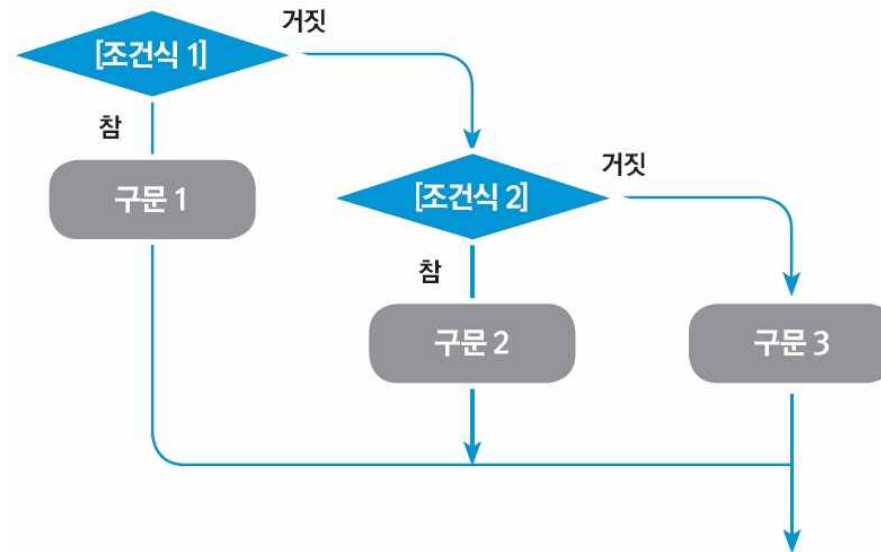


if 문

- `x in s`, `x not in s`
 - 조건문에서 아무 일도 하지 않게 설정하고 싶다면? **pass**
- 예제 5
 - "주머니에 돈이 있으면 가만히 있고 주머니에 돈이 없으면 카드를 꺼내라"

```
>>> pocket = ['paper', 'money', 'cellphone']
>>> if 'money' in pocket:
...     pass
... else:
...     print("카드를 꺼내라")
...
```

elif, else



```
If <조건문>:  
    <수행할 문장1>  
    <수행할 문장2>  
    ...  
elif <조건문>:  
    <수행할 문장1>  
    <수행할 문장2>  
    ...  
elif <조건문>:  
    <수행할 문장1>  
    <수행할 문장2>  
    ...  
...  
else:  
    <수행할 문장1>  
    <수행할 문장2>  
    ...
```

- elif
 - 2개 이상의 조건을 처리하는 경우
 - if는 가장 처음에만 사용할 수 있는 반면에, elif는 필요한 만큼 사용 가능
- else
 - 어떠한 조건에도 해당하지 않는 경우
 - 가장 마지막에만 사용 가능

elif, else

- 예제 1

- "주머니에 돈이 있으면 택시를 타고, 주머니에 돈은 없지만 카드가 있으면 택시를 타고, 돈도 없고 카드도 없으면 걸어 가라"

```
>>> pocket = ['paper', 'handphone']
>>> card = 1
>>> if 'money' in pocket:
...     print("택시를 타고가라")
... else:
...     if card:
...         print("택시를 타고가라")
...     else:
...         print("걸어가라")
...
택시를 타고가라
>>>
```



```
>>> pocket = ['paper', 'cellphone']
>>> card = 1
>>> if 'money' in pocket:
...     print("택시를 타고가라")
... elif card:
...     print("택시를 타고가라")
... else:
...     print("걸어가라")
...
택시를 타고가라
```



elif, else

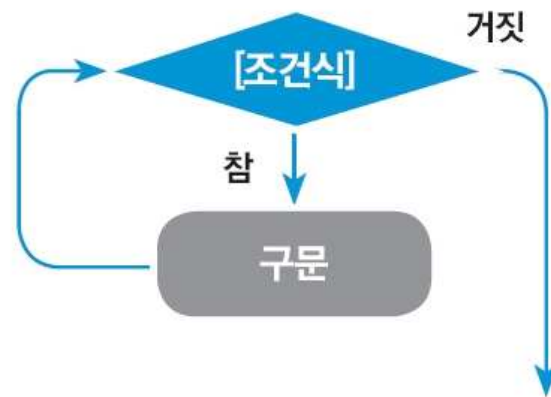
- 예제 2

```
# -*- coding: cp949 -*-
score = int(input('Input Score: ')) # 사용자로부터 정수값을 입력 받습니다
if 90 <= score <= 100:
    grade = "A"
elif 80 <= score < 90:
    grade = "B"
elif 70 <= score < 80:
    grade = "C"
elif 60 <= score < 70:
    grade = "D"
else:
    grade = "F"

print ("Grade is " + grade)
```

- 파이썬의 조건식 표현 방법
 - 70 <= score < 80**
 - grade >= 70 and grade < 80

while 문



```
while <조건문>:  
    <수행할 문장1>  
    <수행할 문장2>  
    <수행할 문장3>  
    ...
```

- 조건식이 참(True)인 동안 내부 구문을 반복 수행
 - 조건식은 구문이 수행되기 이전에 우선 평가
 - 구문을 모두 수행 이후 다시 조건식을 재평가
 - 조건식이 거짓(False)이면 while 문 구조를 벗어남



while 문

- 예제 1
 - 나무를 1번부터 10번까지 찍는 예제

```
>>> treeHit = 0
>>> while treeHit < 10:
...     treeHit = treeHit + 1
...     print("나무를 %d번 찍었습니다." % treeHit)
...     if treeHit == 10:
...         print("나무 넘어갑니다.")
...
나무를 1번 찍었습니다.
나무를 2번 찍었습니다.
나무를 3번 찍었습니다.
나무를 4번 찍었습니다.
나무를 5번 찍었습니다.
나무를 6번 찍었습니다.
나무를 7번 찍었습니다.
나무를 8번 찍었습니다.
나무를 9번 찍었습니다.
나무를 10번 찍었습니다.
나무 넘어갑니다.
```

while 문

- While문 강제로 빠져나가기
 - 에디터를 사용(ctrl+N 으로 새창 띄우기 -> 소스 작성 -> 파일명.py로 저장 -> F5로 실행)

»

```
# coffee.py

coffee = 10
while True:
    money = int(input("돈을 넣어 주세요: "))
    if money == 300:
        print("커피를 줍니다.")
        coffee = coffee - 1
    elif money > 300:
        print("거스름돈 %d를 주고 커피를 줍니다." % (money - 300))
        coffee = coffee - 1
    else:
        print("돈을 다시 돌려주고 커피를 주지 않습니다.")
        print("남은 커피의 양은 %d개 입니다." % coffee)
    if not coffee:
        print("커피가 다 떨어졌습니다. 판매를 중지 합니다.")
        break
```

<출력 결과창>

```
돈을 넣어 주세요: 500
거스름돈 200를 주고 커피를 줍니다.
돈을 넣어 주세요: 300
커피를 줍니다.
돈을 넣어 주세요: 100
돈을 다시 돌려주고 커피를 주지 않습니다.
남은 커피의 양은 8개입니다.
돈을 넣어 주세요:
```



while 문

- 조건에 맞지 않는 경우 맨 처음으로 돌아가기
 - **continue**

```
>>> a = 0
>>> while a < 10:
...     a = a+1
...     if a % 2 == 0: continue
...     print(a)
...
1
3
5
7
9
```



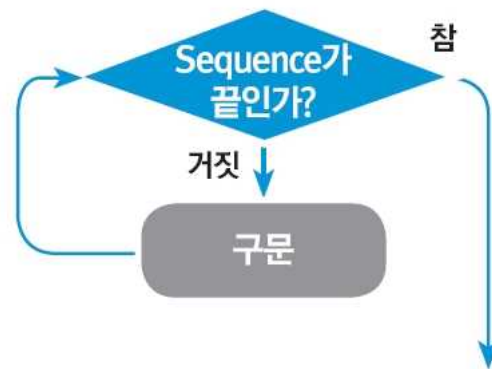
while 문

- 무한 루프 형태

```
while True:  
    수행할 문장1  
    수행할 문장2  
    ...
```

```
>>> while True:  
...     print("Ctrl+C를 눌러야 while문을 빠져나갈 수 있습니다.")  
...  
Ctrl+C를 눌러야 while문을 빠져나갈 수 있습니다.  
Ctrl+C를 눌러야 while문을 빠져나갈 수 있습니다.  
Ctrl+C를 눌러야 while문을 빠져나갈 수 있습니다.  
....
```

for문



```
for 변수 in 리스트(또는 튜플, 문자열):  
    수행할 문장1  
    수행할 문장2  
    ...
```

- 리스트(또는 튜플, 문자열)를 순차적으로 순회
 - 리스트(또는 튜플, 문자열)의 각 아이템을 변수에 할당
 - 할당된 아이템 변수를 가지고 문장을 수행
 - 모든 아이템을 순회하거나 break를 만나면 for문이 종료



for 문

1. 전형적인 for문

```
>>> test_list = ['one', 'two', 'three']  
>>> for i in test_list:  
...     print(i)  
...  
one  
two  
three
```




for 문

2. 다양한 for문의 사용

```
>>> a = [(1,2), (3,4), (5,6)]
>>> for (first, last) in a:
...     print(first + last)
...
3
7
11
```

for 문

3. for문의 응용

- "총 5명의 학생이 시험을 보았는데 시험 점수가 60점이 넘으면 합격이고 그렇지 않으면 불합격이다. 합격인지 불합격인지 결과를 보여주세요."

```
# marks1.py
marks = [90, 25, 67, 45, 80]

number = 0
for mark in marks:
    number = number + 1
    if mark >= 60:
        print("%d번 학생은 합격입니다." % number)
    else:
        print("%d번 학생은 불합격입니다." % number)
```

<출력 결과창>

```
1번 학생은 합격입니다.
2번 학생은 불합격입니다.
3번 학생은 합격입니다.
4번 학생은 불합격입니다.
5번 학생은 합격입니다.
```



for 문

- for문과 continue

```
# marks2.py
marks = [90, 25, 67, 45, 80]

number = 0
for mark in marks:
    number = number + 1
    if mark < 60: continue
    print("%d번 학생 축하합니다. 합격입니다. " % number)
```

<출력 결과창>

```
1번 학생 축하합니다. 합격입니다.
3번 학생 축하합니다. 합격입니다.
5번 학생 축하합니다. 합격입니다.
```



for 문

- for문과 range함수

```
>>> a = range(10)
>>> a
range(0, 10)
```

```
>>> a = range(1, 11)
>>> a
range(1, 11)
```



for 문

- for문과 range함수
 - 예제

```
>>> sum = 0
>>> for i in range(1, 11):
...     sum = sum + i
...
>>> print(sum)
55
```

```
#marks3.py
marks = [90, 25, 67, 45, 80]
for number in range(len(marks)):
    if marks[number] < 60: continue
    print("%d번 학생 축하합니다. 합격입니다." % (number+1))
```



break

- break을 만나면 반복문 내부 블록을 벗어남

```
>>> L = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
>>> for i in L:
    if i > 5:
        break
    print("Item: {}".format(i))
```

```
Item: 1
Item: 2
Item: 3
Item: 4
Item: 5
```



continue

- continue 이후 반복문 내부 블록을 수행하지 않고, 다음 아이템을 선택하여 내부 블록의 시작 지점으로 이동

```
>>> L = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
>>> for i in L:
    if i % 2 == 0:
        continue
    print("Item: {}".format(i))
```

```
Item: 1
Item: 3
Item: 5
Item: 7
Item: 9
```


과제

- 과제 1.
 - 나이가 24세 초과이면 “성인입니다”를 출력,
9세이상 24세 이하이면 “청소년입니다.”를
출력, 9세 미만이면 “어린이 입니다.”를
출력하는 코드를 작성한다.
- (제출시 나이가 5,9,24,60세 일때를 각각
캡쳐하시오)

과제

- 과제 2. 2단부터 9단까지 전부 출력하는 구구단을 찍는 프로그램을 작성한다.
(range함수 사용 X)

과제

- 과제 3. 2단부터 9단까지 전부 출력하는 구구단을 찍는 프로그램을 작성한다.
(range함수 사용)

과제

- 과제 4. **for**문을 이용하여 다음과 같은 별찍기를 실행 한다.

```
★  
★ ★  
★ ★ ★  
★ ★ ★ ★  
★ ★ ★ ★ ★  
★ ★ ★ ★ ★ ★  
★ ★ ★ ★ ★ ★ ★  
★ ★ ★ ★ ★ ★ ★ ★  
★ ★ ★ ★ ★ ★ ★ ★ ★  
★ ★ ★ ★ ★ ★ ★ ★ ★ ★
```

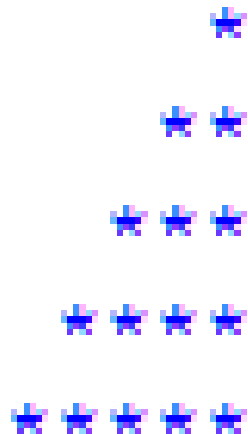
과제

- 과제 5. for문을 이용하여 다음과 같은 별찍기를 실행 한다.

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

과제

- 과제 6. for문을 이용하여 다음과 같은 별찍기를 실행 한다.



과제

- 과제 7. `for`문을 이용하여 다음과 같은 별찍기를 실행 한다.



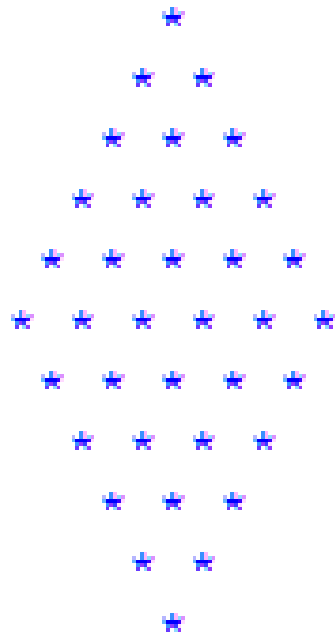
과제

- 과제 8. `for`문을 이용하여 다음과 같은 별찍기를 실행 한다.

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

과제

- 과제 6. `for`문을 이용하여 다음과 같은 별찍기를 실행 한다.



과제 제출 방법

- ▶ 과제캡처 후 워드or한글파일에 첨부/정리하여 제출
- ▶ 파일형식 : [과제번호3]_이름(조이름)_학번
 - ▶ 제출 형식 어길 시 감점처리
- ▶ 제출 : **dbcyy1@gmail.com**로 제출
- ▶ 제출기간 : 3월29일 화요일 23시59분까지
- ▶ 첨부파일 넣었는지, 메일 반송되지 않았는지 꼭 확인하세요!



감사합니다.