

# Python을 활용한 데이터 분석 강의

제어문(if, while, for)

# 제어문 - if

- if문의 형태

```
if 조건문:  
    수행할 문장1  
    수행할 문장2  
else:  
    수행할 문장1  
    수행할 문장2
```

**indentation 중요!**

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

걸어 가라

# 제어문 - if

- elif

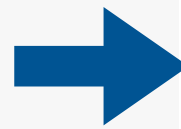
주머니에 돈이 있으면 택시를 타고,

주머니에 돈은 없지만 카드가 있으면 택시를 타고,

돈도 없고 카드도 없으면 걸어 가라

```
pocket = ['paper', 'cellphone']  
card = 1  
if 'money' in pocket:  
    print('택시를 타고 가라')  
else:  
    if card:  
        print('택시를 타고 가라')  
    else:  
        print('걸어 가라')
```

택시를 타고 가라



```
pocket = ['paper', 'cellphone']  
card = 1  
if 'money' in pocket:  
    print('택시를 타고 가라')  
elif card:  
    print('택시를 타고 가라')  
else:  
    print('걸어 가라')
```

택시를 타고 가라

# Practice 2-1 🧐

- 두 정수 비교

- input() 함수를 사용해 사용자로부터 두 정수 값을 입력 받아 x와 y 변수에 할당
- 두 변수를 비교:
  - 만약 x가 y보다 크면 'x가 y보다 큼니다' 출력
  - 만약 x와 y가 같다면 'x와 y가 같습니다' 출력
  - 만약 x가 y보다 작다면 'x가 y보다 작습니다' 출력

```
x를 입력하세요: 7
y를 입력하세요: 5
x가 y보다 큼니다
```

# Practice 2-2 🧐

- 나머지 구하기
  - input() 함수를 사용해 사용자로부터 두 정수 값을 입력 받아 x와 y 변수에 할당
  - x를 y로 나눈 후
    - 만약 나머지가 0이면 'zero' 출력
    - 만약 나머지가 0이 아니고 짝수이면 'even' 출력/홀수이면 'odd' 출력

```
x를 입력하세요: 15  
y를 입력하세요: 7  
odd
```

# Practice 2-3 🧐

- 항목 검색

- 아래와 같이 튜플을 생성

`t = (1, 3, 5, 7, 'a', 'b', 'c')`

- 5가 t에 있으면 '5 is in t' 출력
- d가 t에 있으면 'd is in t' 출력

5 is in t

# 제어문 - while

- 반복해서 문장을 수행해야 할 경우 반복문을 사용(while, for)

열 번 찍어 안 넘어가는 나무 없다

```
treeHit = 0
while treeHit < 10:
    treeHit = treeHit + 1
    print("나무를 {}번 찍었습니다.".format(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.

# 제어문 - while

- 리스트에서 while문 활용하기

```
L = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
i = 0
while i < len(L):
    print(L[i])
    i += 1
```

0  
1  
2  
3  
4  
5  
6  
7  
8  
9



# 제어문 - while

- while문 강제로 빠져나가기 - break
  - ex) 커피 자판기에서 커피가 떨어졌을 때 판매를 중단하게 만드는 경우

```
coffee = 5  #자판기에 커피가 10개 있음
money = 10000
while money > 300:
    print("돈을 받았으니 커피를 줍니다.")
    coffee = coffee - 1
    print("남은 커피의 양은 {}입니다.".format(coffee))
    if coffee == 0:
        print("커피가 다 떨어졌습니다. 판매를 중지합니다.")
        break
```

돈을 받았으니 커피를 줍니다.  
남은 커피의 양은 4입니다.  
돈을 받았으니 커피를 줍니다.  
남은 커피의 양은 3입니다.  
돈을 받았으니 커피를 줍니다.  
남은 커피의 양은 2입니다.  
돈을 받았으니 커피를 줍니다.  
남은 커피의 양은 1입니다.  
돈을 받았으니 커피를 줍니다.  
남은 커피의 양은 0입니다.  
커피가 다 떨어졌습니다. 판매를 중지합니다.

# 제어문 - while

- 조건에 맞지 않는 경우 while문의 맨 처음으로 돌아가기 - continue

1에서 10까지의 자연수 중 홀수만 출력

```
a = 0
while a < 10:
    a = a + 1
    if a % 2 == 0:
        continue
    print(a)
```

a를 2로 나누었을 때  
나머지가 0이면 맨 처음으로 돌아간다

1  
3  
5  
7  
9

- 무한 루프

>>> while True:

## Practice 2-4 🧐

- while/else문 직접 사용해보기
  - number = 0 으로 할당
  - number가 10보다 작으면 '10보다 작습니다'를 출력
  - number를 1씩 증가시키면서 number가 10 이상이 되면 '종료'를 출력하고 while문을 종료

```
10보다 작습니다.  
10보다 작습니다.  
10보다 작습니다.  
10보다 작습니다.  
10보다 작습니다.  
10보다 작습니다.  
10보다 작습니다.  
10보다 작습니다.  
10보다 작습니다.  
10보다 작습니다.  
종료
```

# Practice 2-5 🧐

- 세 정수의 최소공배수 찾기
  - 서로 다른 임의의 정수 3개를 입력 받는다.
  - 이 세 정수의 최소공배수를 찾아내고, '최소공배수는 ~입니다.'를 출력하면서 종료

첫번째 숫자를 입력하세요: 2  
두번째 숫자를 입력하세요: 3  
세번째 숫자를 입력하세요: 9  
최소공배수는 18입니다.

## Practice 2-6 🧐

- while문과 문자열 출력
- 아래 문자열을 담고 있는 리스트를 생성해서 변수 colors에 할당  
'red', 'blue', 'pink', 'green', 'yellow', 'purple'
- 'p'로 시작하는 문자열만 출력

```
pink  
purple
```

# 제어문 - for

- for문의 형태

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

```
one  
two  
three
```

```
fruits = ['apple', 'banana', 'orange', 'cherry', 'grape']  
  
for fruit in fruits:  
    print(fruit)
```

```
apple  
banana  
orange  
cherry  
grape
```

# 제어문 - for

- for문과 함께 자주 사용하는 range 함수

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

- 1부터 10까지 더하기

```
sum = 0

for i in range(1, 11):
    sum = sum + i

print(sum)
```

55

# Practice 2-7 🧐

- 홀수만 출력하는 for문
  - 정수 1부터 10까지 포함하고 있는 리스트를 생성해서 변수 integers에 할당
  - for문으로 리스트를 순환해서 홀수만 출력

1
3
5
7
9



# Practice 2-8 🧐

- 구구단 출력

- for문과 range() 함수를 활용한 구구단 출력
- 아래와 같은 형식으로 구구단을 출력하기

\* print() 함수에 매개변수 end의 값을 ' '로 설정하면 줄바꿈 대신 한칸으로 출력  
ex) print(n, end=' ')

2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

# Practice 2-9 🧐

- 중첩 for문 구성하기

- 다음과 같이 자료형을 구성

color = ['하얗고', '까맣고', '빨갳고', '노랗고']

taste = ['달콤한', '짹짹한', '매콤한', '고소한']

food = ['라면', '피자', '치킨', '햄버거']

- color + taste + food 형식으로 표현 가능한 모든 음식을 출력

하얗고	달콤한	라면
하얗고	달콤한	피자
하얗고	달콤한	치킨
하얗고	달콤한	햄버거
하얗고	짹짹한	라면
하얗고	짹짹한	피자
하얗고	짹짹한	치킨

...

노랗고	매콤한	치킨
노랗고	매콤한	햄버거
노랗고	고소한	라면
노랗고	고소한	피자
노랗고	고소한	치킨
노랗고	고소한	햄버거