

# 코딩, Python으로 같이 시작해요



1주차 자료형, 연산자, if문

# week0-1.py

```
#week1.py
```

```
import time
```

```
r=1.2
print("풍")
time.sleep(r/2)
print("당")
time.sleep(r/2)
print("풍")
time.sleep(r/2)
print("당")
time.sleep(r/2)
print("돌")
time.sleep(r/2)
print("을")
time.sleep(r/2)
print("던")
time.sleep(r/4)
print("지")
time.sleep(r/4)
print("자")
time.sleep(r/2)

print("누")
time.sleep(r/2)
print("나")
time.sleep(r/2)
print("물")
time.sleep(r/2)
print("래")
time.sleep(r/2)
print("돌")
time.sleep(r/2)
print("을")
time.sleep(r/2)
print("던")
time.sleep(r/4)
print("지")
time.sleep(r/4)
print("자")
time.sleep(r/2)
```

# 주석 만들기

## <주석>

- '#' 문자로 시작하는 문장은 주석으로 처리 됩니다.
- 컴파일러에게 코드로 인식되지 않아, 소스코드에 아무런 영향을 주지 않습니다.
- 코드를 짜면서 우리가 메모장처럼 사용하기 위한 용도입니다!

#week1-1.py

```
def add (a,b): #덧셈 연산을 해주는 함수
    return a+b #덧셈 결과를 반환
```

```
result = add(3, 4) #반환 받은 덧셈 결과를 result라는 변수에 저장
```

# 변수 선언하기

<변수 선언하기>

'변수이름 = 값' 형식으로 변수를 만듭니다.

```
>>> x = 10
>>> x
10
>>> x = 12
>>> x
12
>>> y = "문자열"
>>> z = 3.4
>>> print(x, y, z)
12 문자열 3.4
```

```
x = 2
x = 4*x*(1-x)
print(x)
```

'='은 같다는 뜻이 아니라 왼쪽 변수에 오른쪽의 값을 저장하겠다는 뜻입니다.

정수, 실수, 문자열 등 어떤 값이든 저장 가능!

```
>>> x=2
>>> x
2
>>> x='a'
>>> x
'a'
```

왼쪽 코드의 실행결과를 예측해보세요!

# 변수 선언하기

## <변수 이름 정하는 규칙>

- 영문 문자, 숫자, 언더스코어(\_) 사용 가능
- 영문 문자 또는 언더스코어로 시작해야 하며, 숫자로 시작하면 안됨.
- 특수문자 사용 불가(+, \*, &, ^, @, # 등)
- 파이썬의 키워드(input, print, if, for, while, def 등)

안됨!

: 2020data, input

# 자료형

## <자료형>

- 숫자 자료형 : 정수 int, 실수 float
- 문자열 자료형 : str
- 리스트 자료형 : list
- bool 자료형 : True, False

## <type()로 자료형 확인!>

```
>>> type(5)
<class 'int'>
>>> type(3.4)
<class 'float'>
>>> type("string")
<class 'str'>
>>> type([1, 2, 3])
<class 'list'>
>>> type(True)
<class 'bool'>
```

# 기본 연산자

## <기본 연산자>

: + 더하기

: - 빼기

: \* 곱하기

: / 나눗셈(실수 반환)

```
>>> 4/2
```

```
2.0
```

: // 몫(정수 반환)

: % 나머지(모듈러)

: \*\* 거듭제곱

## <정수 간의 연산->정수>

```
>>> 5+2
```

```
7
```

```
>>> 5-2
```

```
3
```

```
>>> 5*2
```

```
10
```

```
>>> 5/2
```

```
2.5
```

```
>>> 5//2
```

```
2
```

```
>>> 5%2
```

```
1
```

```
>>> 5**2
```

```
25
```

## <정수와 실수 간의 연산->실수>

: 정수가 실수로 변환되어서 계산되므로 계산 결과는 실수

```
>>> 4.0+3
```

```
7.0
```

```
>>> 4.0-3
```

```
1.0
```

```
>>> 4.0*3
```

```
12.0
```

```
>>> 5.0/2
```

```
2.5
```

```
>>> 5.0//2
```

```
2.0
```

```
>>> 5.0**2
```

```
25.0
```

# 할당 연산자

<할당 연산자>

: +=  
: -=  
: \*=  
: /=  
: //=  
: %=  
: \*\*=

```
>>> x=2  
>>> x+10  
12  
>>> print(x)  
2
```

```
>>> a =2  
>>> a+=3  
>>> a  
5  
>>> a*=2  
>>> a  
10  
>>> a/=4  
>>> a  
2.5
```

x에 10을 더한 결과를 다시 변수 x에 저장해주어야 함.

- 1)  $x = x + 10$
- 2)  $x += 10$

a=2

a = a+3 #5

a = a\*2 #10

a = a/4 #2.5



## week0-2.py

```
cake_kcal = int(input("케이크의 칼로리 : "))  
cake_num = int(input("먹은 케이크 갯수 : "))  
print()
```

```
rice_kcal = int(input("밥 한 공기 칼로리 : "))  
rice_num = int(input("먹은 공기밥 갯수 : "))  
print()
```

```
print("총 칼로리 : ", cake_kcal*cake_num + rice_kcal*rice_num)
```

# input()

input() : 사용자로부터 값을 입력 받는 함수

`x=input()` 형식으로 사용 -> input()함수의 결과가 변수 x에 **문자열**로 저장

`x=input(" ")` -> 괄호 안의 문구는 어떤 값을 입력해야 하는지 알려주는 안내문구 역할.

```
>>> GPA = input()
4.0
>>> school = input()
한양대학교
>>> print("저는 "+school+"을 다니고 학점은 "+GPA+"입니다.")
저는 한양대학교을 다니고 학점은 4.0입니다.
```

```
name = input("이름 : ")
studentID = input("학번 : ")
print("저는 " + studentID + ", " + name + "입니다.")
```

```
이름 : 윤다영
학번 : 20*****
저는 20*****, 윤다영입니다.
```

# input()

input()은 문자열로 값을 저장하기 때문에 **정수나 실수로 저장하려면 int(), float() 사용**

-> int() : 값을 정수로 변환하는 함수

-> float() : 값을 실수로 변환하는 함수

*x = int(input()), x = float(input())*

```
>>> a1 = input()
3
>>> a2=input()
4
>>> a1+a2
'34'
```

```
>>> a1=int(input())
3
>>> a2=int(input())
4
>>> a1+a2
7
```

# 연습!

- 1) 세 개의 정수를 입력 받아 변수 num1, num2, num3에 저장하고 num1xnum2+num3의 연산결과를 출력하는 코드를 쓰세요.

첫 번째 정수 : 3  
두 번째 정수 : 4  
세 번째 정수 : 7  
 $3 + 4 \times 7 = 31$

- 2) 두 개의 정수를 입력 받아서 몫과 나머지를 출력하는 코드를 쓰세요.

첫 번째 정수 : 5  
두 번째 정수 : 2  
몫 : 2  
나머지 : 1

# 연습!

```
1) num1 = int(input("첫 번째 정수 : "))  
    num2 = int(input("두 번째 정수 : "))  
    num3 = int(input("세 번째 정수 : "))
```

```
print(num1, "+", num2, "x", num3, "=", num1+num2*num3)
```

2) 두 개의 정수를 입력 받아서 몫과 나머지를 출력하는 코드를 쓰세요.

```
num1 = int(input("첫 번째 정수 :"))  
num2 = int(input("두 번째 정수 : "))
```

```
print("몫 :", num1//num2)  
print("나머지 :", num1%num2)
```

# split() 함수 활용

```
>>> first_name, last_name = input("Name : ").split()
Name : John Park
>>> first_name
'John'
>>> last_name
'Park'
```

split()함수는 input()으로 입력받은 문자열 'John Park'을 공백을 기준으로 'John', 'Park' 두개의 문자열로 분리합니다. 그 각각의 문자열은 순서대로 변수 first\_name, last\_name에 저장 됩니다!

## split() 함수 활용

```
>>> num1, num2 = input().split()
3 4
>>> num1+num2
'34'
```

```
>>> num1, num2 = input().split()
3 4
>>> num1 = int(num1)
>>> num2 = int(num2)
>>> num1+num2
7
```

## split() 함수 활용

```
>>> num1, num2 = map(int, input().split())  
3 4  
>>> num1  
3  
>>> num2  
4
```



# print() 특징 4가지

print 하나로 여러 개의 값을 출력하고, 우리가 원하는 대로 출력 형태를 설정하는 방법 배우기!

<제어문자>

: 화면에는 출력되지 않지만, 출력을 제어하는 역할을 함

- \n : 화면에는 출력되지 않지만, 줄을 바꾸게 함. 개행
- \t : tab

```
>>> print("first line\nsecond line")
```

```
first line
```

```
second line
```

```
>>> print("\tThis is Python programming beginner class.")
```

```
    This is Python programming beginner class.
```

# print() 특징 4가지

1. print는 기본적으로 출력하는 값 끝에 \n을 붙인다!

```
print(1)      1
print(2)      2
print(3)      3
|
```

2. print 하나 안에 값 여러 개를 콤마를 기준으로 구분해서 한 줄에 출력되게 할 수 있음. 콤마를 기준으로 공백 한 칸이 출력됨

```
>>> a = "교수님"
>>> b = "과제가"
>>> c = "안"
>>> d = "끝나요"
>>> print(a, b, c, d)
교수님 과제가 안 끝나요
```

## print() 특징 4가지

3. print 하나만 써서 값을 여러 개의 줄에 출력하고 싶을 때는 \n를 활용하세요!

```
print("1\n2\n3\n")
```

1  
2  
3

4. print를 여러 개 사용하면 어떻게 글자들을 한 줄에 출력되게 할까요?

풍당 풍당 돌을 던지자  
누나 몰래 돌을 던지자

# print() 특징 4가지

- 기본적으로 print의 end에는 '\n'이 저장!
- end를 우리가 원하는 문자열로 지정
  - > 줄 바꿈 없이 우리가 원하는 출력 형태 지정 가능

```
print(1, end = ' ' )      123
print(2, end = ' ' )
print(3)
```

```
a = "교수님"
b = "목소리가"
c = "들리지"
d = "않습니다."
print(a, end = ' ')
print(b, end = ' ')
print(c, end = ' ')
print(d)
print("교수님 마이크 고쳐 주세요.")
```

교수님 목소리가 들리지 않습니다.  
교수님 마이크 고쳐 주세요.

# week1-3.py

```
#week1-3.py
import time
```

```
r=1.2
print("퐁", end="")
time.sleep(r/2)
print("당", end=" ")
time.sleep(r/2)
print("퐁", end="")
time.sleep(r/2)
print("당", end=" ")
time.sleep(r/2)
print("돌", end = "")
time.sleep(r/2)
print("을", end=" ")
time.sleep(r/2)
print("던", end="")
time.sleep(r/4)
print("지", end = "")
time.sleep(r/4)
print("자")
time.sleep(r/4)
```

```
print("누", end="")
time.sleep(r/2)
print("나", end=" ")
time.sleep(r/2)
print("몰", end="")
time.sleep(r/2)
print("래", end=" ")
time.sleep(r/2)
print("돌", end = "")
time.sleep(r/2)
print("을", end=" ")
time.sleep(r/2)
print("던", end="")
time.sleep(r/4)
print("지", end = "")
time.sleep(r/4)
print("자", end=" ")
time.sleep(r/4)
```

퐁당 퐁당 돌을 던지자  
누나 몰래 돌을 던지자

# bool 자료형과 비교연산자

bool 자료형 : True, False

type(True) : <class 'bool'>

type(False) : <class 'bool'>

- 참, 거짓을 판단해야 할 때 사용
- True, False를 반환하는 연산자  
: 비교연산자, 논리연산자
- 비교연산자는 두 값을 비교해서 판단하고 True  
False를 반환합니다.

연산자	예제	설명
==	3 == 3	두 피 연산자 값을 비교하여 동일하면 True, 동일하지 않으면 False
!=	3 != 2	두 피 연산자 값을 비교하여 동일하면 False, 동일하지 않으면 True
>	3 > 2	두 피 연산자 값을 비교하여 왼쪽의 값이 크면 True, 그렇지 않으면 False
<	2 < 3	두 피 연산자 값을 비교하여 왼쪽의 값이 작으면 True, 그렇지 않으면 False
>=	3 >= 2	두 피 연산자 값을 비교하여 왼쪽의 값이 크거나 같으면 True, 그렇지 않으면 False
<=	3 <= 3	두 피 연산자 값을 비교하여 왼쪽의 값이 작거나 같으면 True, 그렇지 않으면 False

=과는 다릅니다!!!!

# bool 자료형과 비교연산자

```
>>> a, b = 3, 2
>>> a == b
False
>>> a != b
True
>>> a > b
True
>>> a >= b
True
>>> a < b
False
>>> a <= b
False
```

- 문자열의 같고 다름도 ==, != 로 비교 가능합니다.
- 대소문자 구분합니다.

```
>>> "Apple" == "apple"
False
>>> "hanyang" == "hanyang"
True
```

# bool 자료형과 논리연산자

논리 연산자에는 and, or, not 세 가지가 있습니다.

```
>>> True and True
True
>>> True and False
False
>>> False and True
False
>>> False and False
False
```

```
>>> True or True
True
>>> True or False
True
>>> False or True
True
>>> False or False
False
```

```
>>> not True
False
>>> not False
True
```

False : 정수 0, 실수 0.0, 빈 문자열 "", None

True : 그 이외의 모든 값



# bool 자료형과 논리연산자

예시의 실행 결과를 예측해보세요.

```
num1 = 10
num2 = 12

result1 = (num1==10) and (num2==12)

result2 = (num1>10) or (num2<=12)

result3 = not(num1>=5)
print(result1, result2, result3)
```

```
money = 3000
hungry = False
is_happy = (money>2000) and (not hungry)
print("행복 : ", is_happy)

money -= 2000
hungry = True
is_sad = (money<1000) or hungry
print("슬픔 : ", is_sad)
```

# if, else, elif

*if 조건식:*

*조건식이 참인 경우에 실행할 코드*

```
x = 20
```

```
if x>=10:
```

```
    print("x는 10이상이다.")
```

```
    print("if문 아래 들여쓰기 되어있는 모든 코드가 실행됩니다.")
```

*if 조건식 1:*

*조건식1 이 참일 때 실행할 코드*

*else :*

*조건식1 이 거짓일 때 실행할 코드*

```
a = 12
```

```
if a>=10:
```

```
    print("10보다 크거나 같다")
```

```
else:
```

```
    print("10보다 작다")
```

**#변수 a,b에 값을 int로 입력받아서 두 정수 중 큰 값을 변수 big에 저장하는 코드를 써보세요!**

# if, else, elif

```
a=int(input("첫 번째 정수 : "))  
b=int(input("두 번째 정수 : "))  
if a>b:  
    big = a  
else:  
    big = b  
print(big)|
```

# if, else, elif

*if 조건식 1:*

*조건식1 이 참이면 실행*

*elif 조건식2:*

*조건식1 이 거짓일 때, 조건식2 를 확인해본다.*

*확인결과 참이면 여기 코드 실행*

*else :*

*조건식1, 조건식2 모두 거짓일 때 실행*

```
age=int(input("당신의 나이는 : "))  
if age>=10 and age<20:  
    print("당신은 10대")  
elif age>=20 and age<30:  
    print("당신은 20대")  
else:  
    print("당신은 30대 이상")
```

# if, else, elif

```
name_guess = input("이름을 맞춰보세요 : ")
age_guess = int(input("나이를 맞춰보세요 : "))

if name_guess == "윤다영" and age_guess == 22:
    print("맞았습니다!")
elif name_guess == "윤다영" and age_guess != 22:
    print("이름만 맞았습니다.")
elif name_guess != "윤다영" and age_guess == 22:
    print("나이만 맞았습니다.")
else:
    print("둘 다 틀렸습니다.")
```

이름을 맞춰보세요 : 윤다영  
나이를 맞춰보세요 : 21  
이름만 맞았습니다.

a=5, b=2라고 변수를 선언하고

+, -, \*, /, % 중에서 연산자 하나를 입력 받아서 a와 b를 연산한 값을 알려주는 코드를 짜보세요!

# if, else, elif

```
a,b = 5, 2
op = input("연산자 : ")

if op == '+':
    print(a+b)
elif op == '-':
    print(a-b)
elif op == '*':
    print(a*b)
elif op == '/':
    print(a/b)
elif op == '%':
    print(a%b)
else:
    print("잘못된 연산자 입력")
```

# if문의 중첩

*if 조건식 1:*

*if 조건식2:*

*조건식1 이 참, 조건식2 도 참일 때*

*else:*

*조건식1 이 참, 조건식2 가 거짓일 때*

*else :*

*조건식1 이 거짓일 때 실행*

**#현재 지갑에 있는 돈에 따라 점심 메뉴를 선정해주는 코드**

```
money = 6500
```

```
if money >= 5000:
```

```
    if money < 6000:
```

```
        print("짜장면 배달")
```

```
    else:
```

```
        print("돈까스 정식")
```

```
else:
```

```
    print("학식")
```

# 문자열 자료형 str

<문자열>

- "" 또는 " 로 묶인 텍스트

```
>>> my_string = "This is a string. type()로 자료형을 확인해보세요."
```

```
>>> print(my_string)
```

```
This is a string. type()로 자료형을 확인해보세요.
```

```
>>> type(my_string)
```

```
<class 'str'>
```

여러 줄로 된 문자열을 만들고 싶으면 '''또는 '''를 사용!

```
>>> multi_line = '''이름 : 윤다영  
소속 : 정보시스템학과'''
```

```
>>> print(multi_line)
```

```
이름 : 윤다영
```

```
소속 : 정보시스템학과
```



# 문자열 자료형 str

## <문자열 + 연산 : 연결>

- 문자열 끼리 '+' 연산을 하면 문자열들이 연결됩니다.

```
>>> string1 = "Good"
>>> string2 = " morning! ^@^"
>>> result = string1 + string2
>>> print(result)
Good morning! ^@^
```

- 문자열과 숫자 자료형(정수 int, 실수 float) 끼리는 연산을 할 수 없습니다.

```
>>> a = "갤럭시" + 8
Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    a = "갤럭시" + 8
TypeError: can only concatenate str (not "int") to str
```

# 문자열 자료형 str

<문자열 \* 연산 : 반복>

- 문자열 \* 횟수

-> 해당 횟수 만큼 문자열이 반복됨.

```
>>> my_str = "달고나"
```

```
>>> print(my_str*5)
```

```
달고나달고나달고나달고나달고나
```

# 문자열 자료형 str

## <문자열 인덱싱>

- 문자열의 각 문자에는 위치를 알려주는 인덱스가 있음!
- 문자열\_이름[인덱스]  
: 문자열에서 번호에 해당하는 문자를 뽑아낼 수 있음
- 단, 파이썬은 0부터 숫자를 센다!  
-> 인덱스도 0부터 시작
- a[0] : 첫 번째 문자
- a[-1] : 뒤에서 부터 첫 번째 문자

```
>>> a = "Python is easy!"
>>> a[0]
'P'
>>> a[1]
'y'
>>> a[2]
't'
>>> a[-1]
'!'
>>> a[-2]
'y'
```

# 문자열 자료형 str

## <문자열 슬라이싱>

- 문자열의 한 문턱이를 뽑아내고 싶을 때
- a[시작 번호 : 끝 번호]
  - > 문자열에서 시작 번호 부터 (끝 번호-1) 까지를 뽑아냄.
  - > 끝 번호를 포함하지 않는다!!!!
- a[: 끝 번호]
  - > 시작 번호를 생략하면  
문자열의 처음 부터 끝 번호 까지 추출
- a[시작 번호 : ]
  - > 끝 번호를 생략하면  
시작 번호 부터 끝까지 추출

```
>>> a = "Python is fun!"
>>> a[0:4]
'Pyth'
>>> a[2:5]
'tho'
>>> a[7:]
'is fun!'
>>> a[:6]
'Python'
>>> a[:7]
'Python '
>>> a[:]
'Python is fun!'
>>> b = a[:6]
>>> c = a[-4:-1]
>>> b+c
'Pythonfun'
```

# 리스트 자료형 list

<리스트>

- 여러 값들을 저장하는 목록
- *자료형\_이름 = [요소1, 요소2, 요소3]*

```
>>> prime_numbers = [2, 3, 5, 7, 11, 13]
>>> print(prime_numbers)
[2, 3, 5, 7, 11, 13]
```

```
>>> forif_study = ["C", "Python", "Django", "Android", "Godot", "React"]
>>> print(forif_study)
['C', 'Python', 'Django', 'Android', 'Godot', 'React']
```

```
>>> weird_list = [10, 7.3, "list", True, [0, 1, 2]]
>>> print(weird_list)
[10, 7.3, 'list', True, [0, 1, 2]]
```

# 리스트 자료형 list

range() 함수를 사용해 리스트를 만들어 봅시다.

<range>

- 연속된 숫자를 생성하는 함수
- *리스트\_이름 = range(시작 인덱스, 끝 인덱스)*
  - > 시작 값 부터 (끝 값-1) 까지의 정수를 갖는 리스트 생성
  - > 끝 인덱스는 포함을 하지 않는 것이죠!
- `range_list = list(range(5, 12))`  
: range 함수를 통해 5~11 까지의 정수가 생성되고,  
이 값들이 리스트 형식으로 변수 `range_list`에 저장!

```
>>> my_list = list(range(10))
>>> my_list
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> my_list = list(range(2,10))
>>> my_list
[2, 3, 4, 5, 6, 7, 8, 9]
```

# 리스트 자료형 str

<리스트 + 연산 : 연결>

- 리스트끼리 '+' 연산을 하면 리스트들이 연결됩니다.

<리스트 \* 연산 : 반복>

- 리스트\*횟수  
-> 해당 횟수 만큼 리스트를 반복

```
>>> a = [1,2,3]
>>> b = [4,5,6]
>>> result = a+b
>>> print(result)
[1, 2, 3, 4, 5, 6]
>>>
>>>
>>> print(a*3)
[1, 2, 3, 1, 2, 3, 1, 2, 3]
```

# 리스트 자료형 list

<리스트 인덱싱, 슬라이싱>

```
>>> a = [1, 3.4, "문자열", [1,2,3]]
>>> a[0]
1
>>> a[1]
3.4
>>> a[2]
'문자열'
>>> a[3]
[1, 2, 3]
>>> a[-1]
[1, 2, 3]
>>> a[-2]
'문자열'
>>> a[-1][0]
1
```

```
>>> a[:2] + a[-2:]
[1, 2, 4, 5]
```

```
>>> a = [1,2,3,4,5]
>>> a[0:2]
[1, 2]
>>> a[2:5]
[3, 4, 5]
>>> a[:4]
[1, 2, 3, 4]
>>> a[3:]
[4, 5]
>>> a[:]
[1, 2, 3, 4, 5]
```



# 리스트 자료형 list

## <리스트의 값 수정/삭제/추가>

- 문자열은 값을 변경할 수 없지만, 리스트에서는 값을 수정, 삭제 및 추가할 수 있습니다.
- 수정  
: `a[2] = 4` -> 세 번째 요소를 4로 수정
- 삭제  
: `del a[1]` -> 두 번째 요소를 삭제
- 추가  
: `a.append(5)` -> 리스트의 마지막에 5를 추가

```
>>> a = [1, 2, 3]
>>> a.append(4)
>>> a
[1, 2, 3, 4]
>>> a.append([5,6])
>>> a
[1, 2, 3, 4, [5, 6]]
>>> a = [1, 2, 3]
>>> a.append(4)
>>> a
[1, 2, 3, 4]
>>> a[0] = "first_element"
>>> a
['first_element', 2, 3, 4]
>>> del a[0]
>>> a
[2, 3, 4]
```

# 과제1

week1-1.py 로 저장

세 개의 정수를 입력 받은 후 큰 순서 대로 출력해주는 프로그램을 만드세요.

-> if문을 중첩해서 푸는 방법과 중첩하지 않는 두 가지 방법이 있는데, 두 방식 모두 해보셔도 좋아요!

```
Num1 : 7
```

```
Num2 : 2
```

```
Num3: 9
```

```
9 > 7 > 2
```

# 과제2

week1-2.py 로 저장

<문자열 인덱싱, 슬라이싱 연습>

- : 주민등록 번호를 입력 받아서 연월일 부분을 문자열 슬라이싱으로 추출해서 '생년월일 : 991028' 형식으로 출력하세요.
- : 성별을 뜻하는 뒷자리의 맨 첫 번째 숫자를 추출해서 성별을 판단한 결과를 출력하세요. (남 : 1, 여 :2)
- : 991028 - 2\*\*\*\*\* 형식으로 입력해주세요! 뒷자리의 맨 앞 번호만 알 수 있게! 그리고 꼭 본인 생일 입력 안하셔도 됩니다!! 개인 정보는 소중합니다!!!ㅎㅎ
- : ID = input("주민등록번호 : ") 형식으로 해서 문자열로 입력 받으시면 됩니다.

**주민등록번호를 입력하세요 : 991028-2\*\*\*\*\***

**생년월일 : 991028**

**성별 : 여**

# 과제3

week1-3.py

<리스트 인덱싱, 슬라이싱 연습>

- : movie\_rank = ["어벤져스", "쥬라기 공원", "기생충", "옥자", "살인의 추억"] 처럼 5개를 순위 대로 담은 리스트를 만드세요.
- : 1위는 어벤져스, 2위는 쥬라기 공원, 3위는 기생충....
- : 사용자가 2위 부터 5위 까지 알려 달라하면 리스트 슬라이싱을 해서 추출된 리스트를 보여주세요.
- : 리스트 슬라이싱을 할 때 끝 번호는 포함되지 않는다는 점 유의하세요!

**알고 싶은 영화 순위의 범위를 알려주세요 : 2 5**  
**['기생충', '옥자', '살인의 추억']**

- : 범위 입력 받을 때는 input().split() 활용하시고, 입력 받은 값은 각각 int형으로 변환하셔야 합니다.

# 과제4

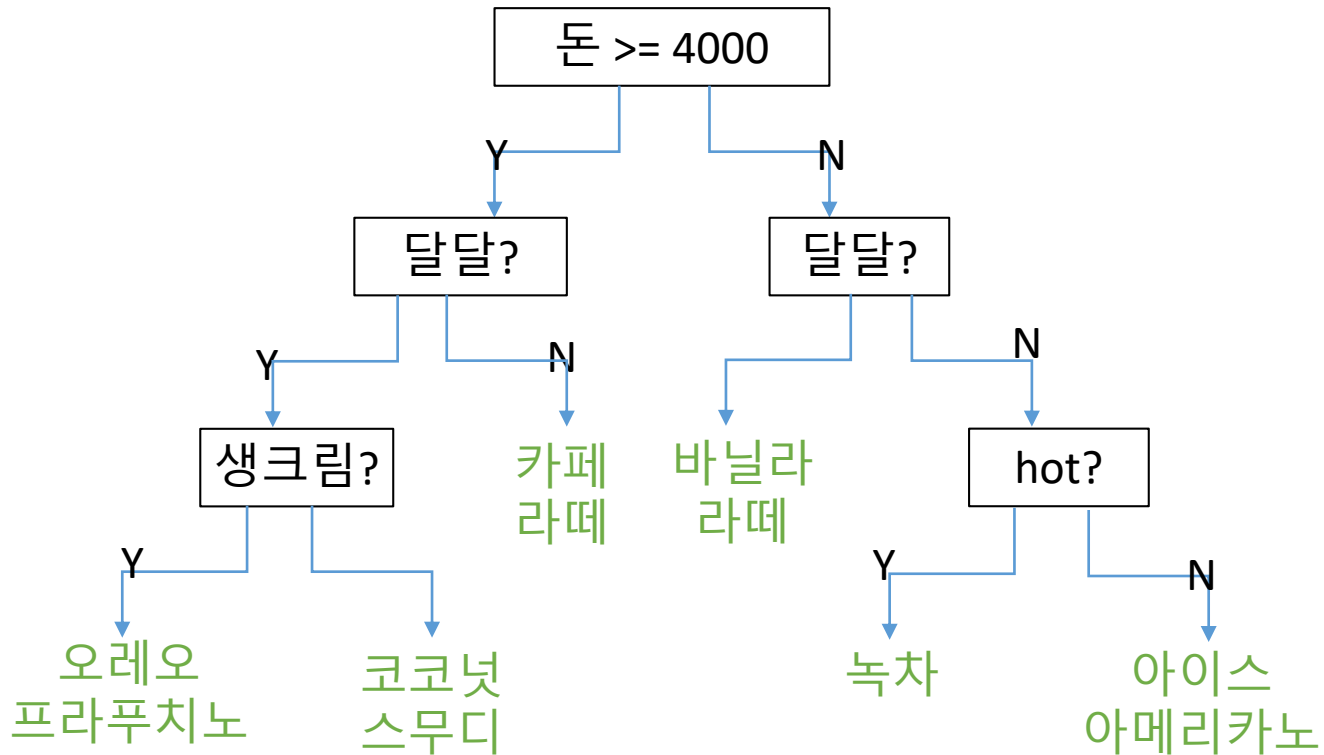
week1-4.py

메뉴 선정 프로그램!

두 개의 판단 기준에 따라 if문, else문을 여러 갈래로 분기해서 카페에서 먹을 메뉴를 선정해주는 프로그램 만들기!

if문과 else문을 중첩해야 합니다.

메뉴는 6개 이상으로 해주세요!



돈 : 3500

달달한 게 땡기시나요?(Y/N) : N

뜨거운 음료가 땡기나요?(Y/N) : N

아이스 아메리카노

# 과제5

week1-5.py

매일 지각하는 사람을 위해 원래 설정 되어 있는 알람을 50분 앞서는 시간으로 바꿔서 설정하는 프로그램을 만드세요.  
사용자에게 일어나고 싶은 시간을 정수 두 개로 입력 받은 뒤 50분 앞선 시간을 계산해서 사용자에게 출력해주면 됩니다.

-> 시, 분 입력 받을 때 split()함수 활용

-> 23:55, 13:04 와 같이 24시간 표현을 사용 ( $0 \leq \text{시} \leq 23$ ,  $0 \leq \text{분} \leq 59$ )

-> 심화 과제(하고 싶으신 분들만!)

: 사용자가 몇 분 일찍 알람을 듣고 싶은지 입력 받아서 그 시간 만큼 일찍 알람을 설정해주는 프로그램!

**내일 몇시에 일어나시나요? 12:35**  
**11 시 45 분에 알람을 맞춰 드릴게요!**

# 과제6

week1-6.py 로 저장

- : 아이디와 비밀번호를 미리 ID, password 두개의 변수에 문자열로 저장합니다.
- : 사용자에게 아이디, 비번을 입력 받고 3가지의 경우의 수에 따라 다른 일을 수행합니다.
- : if문의 조건식에서 논리연산자를 사용해 주세요!(and, or, not) 논리연산자 3개 중 본인이 필요하신 것만 한 개 이상 사용해주시면 됩니당.

1. 아이디 : o, 비번 : o

```
ID : ektmf7890
password : &password&
ektmf7890님, 환영합니다!
```

2. 아이디 : o, 비번 : x  
-> 별은 4개만 출력!

```
ID : ektmf7890
password : 홍
비밀번호를 까먹으셨나요?

힌트를 드릴까요?(y/n) : y
&pas****
```

```
ID : ektmf780
password : 홍홍
비밀번호를 까먹으셨나요?

힌트를 드릴까요?(y/n) : n
로그인에 실패했습니다.
```

3. 아이디 : x

```
ID : ekeke
password : ekeke
존재하지 않는 계정입니다.
로그인에 실패하였습니다.
```

## 과제6 (심화과제)

궁금하고 풀어보고 싶으신 분들만 하시면 됩니다!!

week1-6adv.py 로 저장

아래 실행 화면 처럼 ID가 일치 하지 않으면 비밀번호를 입력 받지 않고 바로 오류 메시지를 띄우게 코드를 수정해보세요!

```
ID : ektnmf
존재하지 않는 계정입니다.
로그인에 실패했습니다.
```

이외의 모든 실행결과는 앞 슬라이드와 동일하게 해주시면 됩니다.



# 예습 범위 안내

## 1. 문자열과 리스트의 활용

: 점프 투 파이썬 위키독스 <https://wikidocs.net/13>

: 2-2장 문자열 자료형(1.문자열이란 ~ 4. 문자열 슬라이싱과 인덱싱)

: 2-3장 리스트 자료형(1.리스트는 어떻게 만들고 사용할까? ~ 5.1리스트에 요소추가(append))

## 2. 반복문(for, while)

: 코딩도장 <https://dojang.io/mod/page/view.php?id=2220>

: Unit13~18

## 3. 함수

: 코딩도장

: Unit29

## 과제 제출 안내

과제 제출은 다음 주 월요일(4.13) 23:59 까지 입니다!

제출 이메일 : [ektmf7890@hanyang.ac.kr](mailto:ektmf7890@hanyang.ac.kr)

(이번 주에 과제 제출하신 이메일과 달라진 점 유의해주세요!)

이번 주는 예습과제 없이 복습 과제만 나갑니다.