

리스트

```
numList = ["1", "2", "3"]
```

```
numList.append("4")
```

```
numList.insert(4, "5")
```

```
print(numList)
```

```
del numList[-1]
```

```
numList.pop(-1)    # return 값 있음
```

```
numList.insert(0, "3")
```

```
numList.remove("3") # 중복된 값이 있으면, 앞애가 먼저 삭제
```

```
print(numList)
```

```
print(numList.index("1"))
```

```
numList.append("1")
```

```
print(numList.count("1"))
```

```
numList_string = "/" .join(numList) # str 원소여야됨
```

```
print(numList_string)
```

```
numList_lists = numList_string.split("/")
```

```
print(numList_lists)
```

```
print("정렬전", numList)
```

```
numList.sort()    #인자로 reverse=True 하면 내림차순
```

```
print("정렬후", numList)
```

```
list_a = [2, 3, 1]
```

```
list_copy = sorted(list_a)
```

```
print(list_a, list_copy)
```

```
copy_a = [1, 2, 3, 4]
```

```
copy_b = copy_a[:]    # 슬라이싱 copy_a.copy()
```

```
# copy_b = copy_a    공유
```

```
copy_b[-1] = 5
```

```
print(copy_b)
```

확장

```
me = [1, 2, 3, 4, 5]
```

```
you = [6, 7, 8]
```

```
you.extend(me)
print(you)
```

#조건문

```
age = int(input("나이가 어떻게 되세요? "))
width_parents = input("보호자랑 같이 오셨나요? 네/아니오 ")
```

```
if age >= 15:
    print("15 세 이상 관람가 입장가능~")
elif width_parents == "네":
    print("보호자 계시면 입장가능~")
else:
    print("입장을 하실 수 없습니다!")
```

반복문

```
for i in range(0, 5, 2):          # 시작, 끝, 증감
    print(i, end=" ")
```

```
print()
words = "I love python"
for i in range(len(words)):
    print(words[i], end="")
```

```
# for i in words:
#     print(i)
```

format 포맷

```
format_a = 10
format_b = 20
format_c = 30
print()
print("a : {0}, b : {1}, c : {2}".format(format_a, format_b, format_c))
print("a : %d, b : %d, c : %d" % (format_a, format_b, format_c))
```

```
print()
# numbers = []
# for k in range(0, 10, 2):
#     numbers.append(k * k)
# print(numbers)
```

```
numbers = [k * k for k in range(0, 10, 2)]
print(numbers)
```

```
print()
n = [x for x in range(1, 11) if x % 3 == 0]
print(n)
```

```
print()
# 랜덤 random
import random
dice = random.randint(1, 6)    # 1 ~ 6 까지
print(dice)
```

```
print()
lotto = []
while(len(lotto) < 6):
    lotto.append(random.randint(1, 45))
    lotto = list(set(lotto))    # set 집합, 중복 제거
print(sorted(lotto))
```

```
# 딕셔너리 사전
test_list = ["ab", "de", "ad", "a1"]
print(dict(test_list))    # 각 원소 첫 글자를 key로 뒷 글자를 value로 2글자가 해당
```

```
before = {"apple": "사과", "watermelon": "수박"}
after = {"watermelon": "수수박"}
before.update(after)
print(before)
```

```
del before["apple"]    # 원소 삭제
#del before            # 딕셔너리 삭제
before["banana"] = "바나나"
print(before)
```

```
before.clear()    # 초기화
print(before)
```

```
# keys, values, items
dic = {"name": "이준용", "age": "22", "hobby": "eat"}
```

```
for k, v in dic.items():  
    print(k, v)
```

```
star = ["테란", "저그", "프로토스"]  
print(random.choice(star))  
print(random.choice(list(dic)))
```

```
before_dic = {"name": "카피", "method": "copy"}  
after_dic = before_dic.copy()  
after_dic["name"] = "카피 2"  
print(before_dic["name"])
```

```
# tuple 튜플  
# 튜플의 원소를 정의한 후에는 추가, 삭제, 수정 불가
```

```
# empty = ()  
# numbers = (1, -9, -7)  
# print(numbers[-1])
```

```
subjects = ("python", "c++", "english")  
for subject in subjects:  
    print(subject)
```

```
# 패킹 언패킹 기능  
kim, han, tom = subjects  
print(kim, han, tom)
```

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

```
x, y = (y, x) # packing 과 unpacking 을 동시에 수행  
  
print(x, y)
```

```
# 함수 function  
def test(name):  
    '''help 하면 보임'''  
    return "hi" + name
```

```
print(test("파이썬"))
help(test)
print(test.__doc__)
```

가변 매개변수 (맨 뒤, 오직 하나)

```
def print_even(times, *values):
    for value in values:
        print(value * times)
```

#print_even(2, 5, 10, 15, 20)

기본 매개변수 (맨 뒤)

```
def print_even2(value, times=3):
    print(times * value)
```

```
print_even2(5, 2)
```

```
print()
```

factorial

```
def factorial_recursion(n):
```

"""

팩토리얼 by 재귀

*$f(n) = n * n - 1 * n - 2 * n - 3 * \dots * 1$*

*$f(5) = 5 * f(n-1)$*

"""

```
    if n == 0:
```

```
        return 1
```

```
    else:
```

```
        return n * factorial_recursion(n - 1)
```

```
print(factorial_recursion(4))
```

```
def factorial_loop(n):
```

```
    result = 1
```

```
    for i in range(1, n+1):
```

```
        result = result * i
```

```
    return result
```

```
print(factorial_loop(4))
```

```
# fibonacci
```

```
def fibo_recursion(n):
```

```
    """
```

```
     $f(n) = f(n-1) + f(n-2)$ 
```

```
     $f(1) = 1$ 
```

```
     $f(2) = 1$ 
```

```
    """
```

```
    if n == 1:
```

```
        return 1
```

```
    if n == 2:
```

```
        return 1
```

```
    else:
```

```
        return fibo_recursion(n - 1) + fibo_recursion(n - 2)
```

```
for k in range(1, 8):
```

```
    print("피보나치 {0} : {1}".format(k, fibo_recursion(k)))
```

```
# 매개변수로 함수를 전달하기
```

```
def print_hi(a):
```

```
    for i in range(5):
```

```
        a()
```

```
def hi():
```

```
    print("hi")
```

```
print_hi(hi)
```

```
# 표준함수 map
```

```
# map(함수, 순환가능한 자료구조)
```

```
# 리스트, 딕셔너리, 문자열, range
```

```
def square(n):
```

```
    return n*n
```

```
def odd(n):
```

```
    return n % 2 == 1
```

```
# for k in range(1, 6):
#     print(square(k))
print(list(map(square, [1,2,3,4,5])))
print(list(filter(odd, [1,2,3,4,5])))
```

```
# result = []
# for k in range(1, 6):
#     result.append(square(k))
# print(result)
```

모듈 module

```
import math
print(math.ceil(3.1))    # 올림
print(round(3.1))        # 반올림
print(math.floor(3.1))   # 내림
print(math.sqrt(16))     # 제곱근
```

```
import my_math
```

```
print(my_math.factorial_loop(5))
print(my_math.square(5))
print(my_math.power(2, 4))
```

```
from my_math import fibo_recursion  # from 모듈이름 import 가져오고 싶은 함수 또는 변수
print(fibo_recursion(7))
```

```
import my_math as mm              # 모듈명 줄이기
print(mm.power(2, 10))
```

```
from my_math import *            # 모두
from my_maty import power, square  #여러개
print(power(2, 10))
```

```
'''
```

file 파일

파일 객체 = open(파일 경로, 모드)

w : 쓰기, r : 읽기, a : 이어쓰기

파일 닫을 때, 파일객체.close()

```
'''
```

```
fp = open("test.txt", "w")
```

```
print("1 니", file=fp) # 실제 쓰기
print("2 니", file=fp) # 실제 쓰기
print("3 니", file=fp) # 실제 쓰기
fp.write("4 니")
fp.close()
```

```
fp = open("test.txt", "r")      # r은 생략 가능
lines = fp.readlines()         # 파일을 1행 단위로 리스트 원소로 리턴
# print(lines)
for line in lines:
    # print(line.rstrip("\n"))
    # print(line.strip("\n"))
    # print(line[:-1])          # 슬라이싱 이용
    print(line, end="")
# for line in fp:
#     print(line, end="")
fp.close()
```

```
# with 자동으로 close 해줌
with open("test.txt") as fp:
    lines = fp.readlines()
    for line in lines:
        print(line[:-1])
```

```
import random
alcohol_foods = {}
with open("alcohols.txt", "r") as fp1:
    with open("foods.txt", "r") as fp2:
        alcohols = fp1.readlines()
        foods = fp2.readlines()

        for k in range(len(alcohols)):
            alcohol_foods[alcohols[k].strip("\n")] = foods[k][:-1]
```

```
while True:
    alcohol = input("주문하실 술(맥주/와인/소주/고량주/아무거나/결제)은?")

    if alcohol == "결제":
        break
```



```

if alcohol in alcohol_foods.keys():
    print("{0}에 어울리는 안주는 {1}입니다.".format(alcohol, alcohol_foods[alcohol]))
elif alcohol == "아무거나":
    any = random.choice(list(alcohol_foods))
    print("{0}을 추천합니다. 안주는 {1}입니다.".format(any, alcohol_foods[any]))
else:
    print("{0}는 판매하지 않습니다. 메뉴에서 골라주세요~".format(alcohol))

```

```

try :
    a = input()
    b = input()

    if a.isdigit() and b.isdigit() :
        print(a, b)
    else:
        print("입력된 수는 정수가 아닙니다.")
except :
    pass

```

```

try:
    a = int(input())
    b = int(input())

    print(a + b)
except:
    print("입력된 수는 정수가 아닙니다.")

```

```

try:
    c = list()
    c.append("사과")
    a = int(input())
    b = int(input())
    print(a / b)
    print(c[1])
except ZeroDivisionError:
    print("분모에 0 이 올 수 없습니다.")
except ValueError:
    print("입력된 수는 정수가 아닙니다.")

```

```

except IndexError:
    print("리스트의 범위를 벗어난 인덱스가 사용되었습니다")
except:
    print("무언가 에러가 발생했습니다.")
else:
    print("정상적으로 처리되었습니다.")
finally:
    print("예외 발생 여부에 상관없이 항상 실행됩니다")

```

```

try:
    a = [1, 2, 3]
    print(a[3])
except ZeroDivisionError as err:
    print("분모에 0 이 올 수 없습니다 : {0}".format(err))
except IndexError as err:
    print("인덱스 범위를 벗어났습니다 : {0}".format(err))
except Exception as err:
    print("무언가 에러가 발생했습니다. : {0}".format(err))

```

#내가 한 4 주차코드

#Module

```
import math
```

```
import random
```

```
print(math.ceil(3.1)) #4
```

```
print(math.floor(4.9)) #4
```

```
print(random.randint(1, 6)) #1~6 (랜덤 숫자)
```

```
math.sqrt(16) #4.0
```

#팩토리얼

```

def fac(n) :
    result = 1
    for i in range(1, n + 1) :
        result = result * i;
    return result

```

#피보나치

```
def fibo(n) :  
    if n == 1 :  
        return 1  
    elif n == 2 :  
        return 1  
    else :  
        return fibo(n - 1) + fibo(n - 2)
```

#거듭제곱

```
def power(i, j) :  
    result = 1  
    for k in range(j) :  
        result = result * i
```

#파일 IO

'''

w : 쓰기, r : 읽기, a : 이어쓰기

.close() 파일 닫기

'''

```
fp = open("=====.txt", "w")  
print("=====", file=fp) #실제쓰기  
fp.write("=====") #실제쓰기  
fp.close()
```

#파일 읽기

```
fp = open("=====.txt", "r")  
for line in fp :  
    print(line) #줄바꿈 포함하여 출력후 다시 줄바꿈  
    print(line, end="") #한번만 줄바꿈
```

```
line = fp.readlines() #파일을 1 행 단위로 리스트의 원소로 리턴  
print(line) #['=====\n', '=====\n']
```

```
for line in line :  
    print(line.rstrip('\n')) #한줄씩 엔터되서 출력  
    print(line[0:-1]) #한줄씩 엔터되서 출력  
    print(line, end="") #한줄씩 엔터되서 출력
```

#with

```
with open("=====.txt") as fp :  
    line = fp.readlines()  
    for line in line :  
        print(line[:-1]) #한줄씩 엔터되서 출력
```

#안주추천 프로그램 v 0.5

```
import random
```

```
import tkinter as tk
```

```
alcohol_foods = {'맥주' : '치킨', '와인' : '치즈', '고량주' : '짬뽕', '소주' : '골뱅이소면'}
```

```
def sul(lbl_alchol=None) :
```

```
    alcohol_foods = {'맥주': '치킨',  
                     '와인': '치즈',  
                     '고량주': '짬뽕',  
                     '소주': '골뱅이소면'}
```

```
    while True :
```

```
        alchol = en_input.get()
```

```
        if alchol == '결제':
```

```
            lbl_alchol.configure(text='다음에 또 오세요.')
```

```
            break
```

```
        if alchol in alcohol_foods.keys():
```

```
            lbl_alchol.configure(text='{0}에 어울리는 안주는 {1}입니다.'.format(alchol,  
alcohol_foods[alchol]))
```

```
        elif alchol == '아무거나':
```

```
            lbl_alchol.configure(random.choice(list(alcohol_foods)))
```

```
        else:
```

```
            lbl_alchol.configure(text='{0}는 판매하지 않습니다 메뉴에서  
골라주세요'.format(alchol))
```

```
w = tk.Tk()
```

```
w.title('3 학년 A 반 Quiz')
```

```
w.geometry('500x200')
```

```
lbl_name = tk.Label(w, text='주문하실 술(' + '/' .join([alcohol.rstrip('\n') for alcohol in alcohols]) +  
)은?')
```

```
lbl_alchol = tk.Label(w)
```

```
en_input = tk.Entry(w)
```

```

btn_sul = tk.Button(w, text='확인', command=sul)
lbl_name.pack()
lbl_alchol = tk.Label(w)
en_input.pack()
btn_sul.pack()
en_input.focus()
w.mainloop()

```

```

#try except finally

```

```

try :

```

```

    raise Exception('내가만든 예외')

```

```

    #코드부분

```

```

except ValueError as e:

```

```

    pass

```

```

    #에러처리부분

```

```

    #print("==== {0}".format(e))

```

```

except Exception as e :

```

```

    print("{0}".format(e)) #내가만든 예외

```

```

finally :

```

```

    pass

```

```

    #예외 발생 여부에 상관없이 실행

```

```

'''

```

```

'''

```

```

#A 받

```

```

import random

```

```

import tkinter as tk

```

```

def sul(lbl_alchol=None) :

```

```

    alcohol_foods = {'맥주': '치킨',
                     '와인': '치즈',
                     '고량주': '짬뽕',
                     '소주': '골뱅이소면'}

```

```

    while True :

```

```

        alchol = en_input.get()

```

```

        if alchol == '결제':
            lbl_alchol.configure(text='다음에 또 오세요.')
            break
        if alchol in alcohol_foods.keys():
            lbl_alchol.configure(text='{0}에 어울리는 안주는 {1}입니다.'.format(alchol,
alcohol_foods[alchol]))
        elif alchol == '아무거나':
            lbl_alchol.configure(random.choice(list(alcohol_foods)))
        else:
            lbl_alchol.configure(text='{0}는 판매하지 않습니다 메뉴에서
콜라주세요'.format(alchol))

```

```

w = tk.Tk()
w.title('3 학년 A 반 Quiz')
w.geometry('500x200')

```

```

lbl_name = tk.Label(w, text='주문하실 술(맥주/와인/소주/고량주/아무거나/결제)은?')
lbl_alchol = tk.Label(w)
en_input = tk.Entry(w)
btn_sul = tk.Button(w, text='확인', command=sul)

```

```

lbl_name.pack()
lbl_alchol = tk.Label(w)
en_input.pack()
btn_sul.pack()

```

```

en_input.focus()

```

```

w.mainloop()

```

```

import tkinter as tk

def f2cEnter(ev) :
    f2c()

def f2c():
    """ 화씨 온도를 섭씨 온도로 바꿔주는 함수(엔트리 객체로 부터 입력, 레이블
    객체에 출력) """
    
$$\#(32^{\circ}\text{F} - 32) \times 5/9 = 0^{\circ}\text{C}$$

    try:
        f = en_input.get()
        c = (float(f) - 32) * 5 / 9;
        lbl_temp.configure(text=f"화씨 {f}도는 섭씨 {round(c, 4)}도 입니다")
    except ValueError as e:
        lbl_temp.configure(text="{0}는 숫자가 아닙니다. 숫자를
        입력해주세요\n{1}".format(en_input.get(), e))
        en_input.delete(0, "end")
        
$$\# \text{print}(f\text{"화씨"}\{f\}\text{도는 섭씨}\{round(c, 4)}\text{도 입니다})$$


def c2f():
    """ 섭씨 온도를 화씨 온도로 바꿔주는 함수(엔트리 객체로 부터 입력, 레이블
    객체에 출력) """
    
$$\# (32^{\circ}\text{C} \times 9/5) + 32 = 89.6^{\circ}\text{F}$$

    try:
        c = en_input.get()
        f = (float(c) * 9/5) + 32;
        lbl_temp.configure(text=f"섭씨 {c}도는 화씨 {round(f, 4)}도 입니다")
    except ValueError as e:
        lbl_temp.configure(text="{0}는 숫자가 아닙니다. 숫자를
        입력해주세요\n{1}".format(en_input.get(), e))
        en_input.delete(0, "end")

if __name__ == "__main__":
    help(f2c)
    w = tk.Tk()
    w.title("온도변환프로그램0.1")
    w.geometry("300x100")

```

```
en_input = tk.Entry(w)
btn_f2c = tk.Button(w, text="화씨->섭씨", command=f2c)
btn_c2f = tk.Button(w, text="섭씨->화씨", command=c2f)
lbl_temp = tk.Label(w, text="온도변환")

en_input.bind("<Return>", f2cEnter)

en_input.pack()
btn_f2c.pack(fill="x")
btn_c2f.pack(fill="x")
lbl_temp.pack()

en_input.focus()

w.mainloop()
```



```
import tkinter as tk

def star() :
    n = int(inputstar.get())
    strStar = ""

    for line in range(1, n + 1):
        for space in range(n - line, 0, -1):
            strStar = strStar + " "
        for star in range(1, 2 * line):
            strStar = strStar + "*"
        strStar = strStar + "\n"
    lblstar.configure(text = strStar)

if __name__ == '__main__' :
    w = tk.Tk()

    w.title("별 찍기")
    w.geometry("300x100")
    inputstar = tk.Entry(w)
    btnstar = tk.Button(w, text="별 찍기", command=star)
    lblstar = tk.Label(w)

    inputstar.pack()
    btnstar.pack()
    lblstar.pack()

    w.mainloop()
```

```

import random
import tkinter as tk

b3 = ['송민섭', '간명해', '이교범', '김준영', '김현호', '조민호', '전유진', '강성모', '고대현',
'김광래', '임진수', '김주연', '한승민', '황태결', '이준용']

w = tk.Tk()
w.title("KJY_201944036")
w.geometry("300x200")

def plus() :
    try :
        result = int(inputnum1.get()) + int(inputnum2.get())
        lblresult.configure(text=result)
    except ValueError as e :
        result = "숫자만 입력해주세요. \n{0}".format(e)
        lblresult.configure(text=result)

def divide() :
    try :
        result = int(inputnum1.get()) / int(inputnum2.get())
        lblresult.configure(text=result)
    except ValueError as e :
        result = "숫자만 입력해주세요. \n{0}".format(e)
        lblresult.configure(text=result)
    except ZeroDivisionError as e :
        result = "분모에는0이 올 수 없습니다. \n{0}".format(e)
        lblresult.configure(text=result)

def random_num() :
    try :
        num1 = random.randint(0, 9)
        num2 = random.randint(0, 9)

```

```

        plus = num1 + num2
        div = num1 / num2
        result = "첫번째 숫자: " + str(num1) + ", 두번째 숫자: " + str(num2) + "\n
더하기: " + str(plus) + ", 나누기: " + str(div)
        lblresult.configure(text=result)
    except ZeroDivisionError as e :
        result = "첫번째 숫자: " + str(num1) + ", 두번째 숫자: " + str(num2) + "\n
더하기: " + str(plus) + "\n분모에는0이 올 수 없습니다. {0}".format(e)
        lblresult.configure(text=result)

```

```

def random_name() :
    with open("quiz.txt", 'w') as fp :
        for i in range(len(b3)) :
            num = random.randint(1, 100)
            name = b3[-i - 1]
            result = str(num) + ", " + name + "\n"
            fp.write(result)
        lblresult.configure(text="quiz.txt 생성 완료")

```

```

inputnum1 = tk.Entry(w)
inputnum2 = tk.Entry(w)
btnplus = tk.Button(w, text="더하기", command=plus)
btndiv = tk.Button(w, text="나누기", command=divide)
btnrannum = tk.Button(w, text="랜덤번호() 각각 0~9사이의 수", command=random_num)
btnrananame = tk.Button(w, text="성명, 랜덤번호", command=random_name)
lblresult = tk.Label(w)

```

```

inputnum1.pack(fill="x")
inputnum2.pack(fill="x")
btnplus.pack(fill="x")
btndiv.pack(fill="x")
btnrannum.pack(fill="x")
btnrananame.pack(fill="x")
lblresult.pack(fill="x")

```

```

w.mainloop()

```

```

import tkinter as tk
from tkinter import messagebox
from tkinter import ttk

def clickPhoto(ev) :
    messagebox.showinfo("좌표", f"{ev.x}, {ev.y}")

def enterNext(ev) :
    clickNext()

def enterBefore(ev) :
    clickBefore()

def clickNext() :
    global idx
    idx += 1
    if idx > len(photos) - 1 :
        idx = 0

    photos[idx]
    p.configure(file=photos[idx])
    lblPage.configure(text=f"{idx + 1}/{len(photos)}")

def clickBefore() :
    global idx
    idx -= 1
    if idx < 0 :
        idx = len(photos) - 1

    photos[idx]
    p.configure(file=photos[idx])
    lblPage.configure(text=f"{idx + 1}/{len(photos)}")

```

#전역변수

```
photos = ['michael.PNG', 'franklin.PNG', 'trevor.PNG']  
idx = 0
```

```
w = tk.Tk()  
w.title("포토뷰어v0.2")  
w.geometry("500x500")
```

```
p = tk.PhotoImage(file = photos[0])
```

```
lblPhoto = ttk.Label(w, image=p)  
btnNext = ttk.Button(w, text = "다음->", command = clickNext)  
btnBefore = ttk.Button(w, text = "<- 이전", command = clickBefore)  
lblPage = ttk.Label(w, text=f"{idx + 1}/{len(photos)}")
```

```
w.bind('<Next>', enterNext)  
w.bind('<Prior>', enterBefore)  
lblPhoto.bind('<Button-1>', clickPhoto)
```

```
lblPhoto.grid(row=0, column=0, columnspan=3)  
btnBefore.grid(row=1, column=0, sticky=tk.EW)  
lblPage.grid(row=1, column=1)  
btnNext.grid(row=1, column=2, sticky=tk.EW)
```

```
w.mainloop()
```

```

import tkinter as tk

def popup() :
    if selected.get() == 0 :
        lblDisplay.configure(image=p1)
    elif selected.get() == 1 :
        lblDisplay.configure(image=p2)
    else :
        lblDisplay.configure(image=p3)

w = tk.Tk()
w.title("라디오버튼 실습")
w.geometry("500x500")

p1 = tk.PhotoImage(file = "michael.PNG")
p2 = tk.PhotoImage(file = "franklin.PNG")
p3 = tk.PhotoImage(file = "trevor.PNG")

selected = tk.IntVar()
rdoBtn1 = tk.Radiobutton(w, text = "마이클", command=popup, variable=selected,
value=0)
rdoBtn2 = tk.Radiobutton(w, text = "프랭클린", command=popup, variable=selected,
value=1)
rdoBtn3 = tk.Radiobutton(w, text = "트레버", command=popup, variable=selected,
value=2)

lblDisplay = tk.Label(w, text = "플레이어 선택")
lblDisplay.configure(image=p1)

rdoBtn1.grid(row = 0, column=0)
rdoBtn2.grid(row = 0, column=1)
rdoBtn3.grid(row = 0, column=2)
lblDisplay.grid(row=1, column=0, columnspan=3)

```

```
w.mainloop()
```

```
import tkinter as tk
```

```
from tkinter import messagebox
```

```
def popup() :
```

```
    if checked.get() == 0 :
```

```
        lblDisplay.configure(text = "체크버튼OFF")
```

```
        messagebox.showinfo("체크안됨", "체크버튼OFF")
```

```
    elif checked.get() == 1 :
```

```
        lblDisplay.configure(text="체크버튼ON")
```

```
        messagebox.showinfo("체크됨", "체크버튼ON")
```

```
    else :
```

```
        messagebox.showerror("오류", "실행될 일 없음")
```

```
w = tk.Tk()
```

```
w.title("체크버튼 실습")
```

```
w.geometry("300x100")
```

```
checked = tk.IntVar()
```

```
chbOnoff = tk.Checkbutton(w, text = "출석체크", variable = checked, command=popup)
```

```
lblDisplay = tk.Label(w, text = "")
```

```
chbOnoff.pack()
```

```
lblDisplay.pack()
```

```
w.mainloop()
```

```
import tkinter as tk
from tkinter import messagebox

w = tk.Tk()
w.title("세 번째 GUI 프로그램")

def popup() :
    #messagebox.showinfo("클릭", "버튼이 눌렸습니다.")
    #messagebox.showwarning("주의", "버튼이 눌렸습니다.")
    #messagebox.showerror("오류", "버튼이 눌렸습니다.")
    #messagebox.askokcancel("버튼2개", "버튼이 눌렸습니다.")
    messagebox.askyesnocancel("버튼3개", "버튼이 눌렸습니다.")

p1 = tk.PhotoImage(file = "michael.PNG")
p2 = tk.PhotoImage(file = "franklin.PNG")
p3 = tk.PhotoImage(file = "trevor.PNG")

lbl_disp1 = tk.Label(w, image = p1)
lbl_disp2 = tk.Label(w, image = p2)
btn_disp3 = tk.Button(w, image = p3, command=popup)

lbl_disp1.pack(side = 'left')
lbl_disp2.pack(side = 'left')
btn_disp3.pack()

w.mainloop()
```



```
import tkinter as tk
```

```
w = tk.Tk()
```

```
w.title("세 번째 GUI 프로그램")
```

```
p1 = tk.PhotoImage(file = "michael.PNG")
```

```
p2 = tk.PhotoImage(file = "franklin.PNG")
```

```
p3 = tk.PhotoImage(file = "trevor.PNG")
```

```
lbl_disp1 = tk.Label(w, image = p1)
```

```
lbl_disp2 = tk.Label(w, image = p2)
```

```
lbl_disp3 = tk.Label(w, image = p3)
```

```
lbl_disp1.pack(side = 'left')
```

```
lbl_disp2.pack(side = 'left')
```

```
lbl_disp3.pack()
```

```
w.mainloop()
```