

# 파이썬 기초 프로그래밍

- 파일 **open** 및 **read**

```
f = open("Testfile1.txt", 'r')
for line in f:          # 파일 객체 줄 단위로 항목이 순회
    print(line, end="") # == print(line.strip())
else:
    print('\n')

f.seek(0.0)
print(f.readlines())    # 파일의 내용을 리스트로 반환
f.close()
```

```
python programming
good
study
very very hot language
```

```
['python programming\n', 'good\n', 'study\n', 'very very hot language']
[Finished in 0.5s]
```

- 파일 임의 접근

```
f = open("D:\Python_Study\Testfile.txt", 'r+')
print(f.read())           # 파일 내용 전체 읽기
print(f.tell())           # 현재 파일 포인터 위치 출력
f.seek(0,0)               # 파일 포인터 위치를 맨 앞으로 이동

f.write("c/c++")           # 현재 위치에 "c/c++"문자열 저장
f.flush()                 # 쓰기 버퍼 비워주는 동작(실제 파일에 저장)
f.seek(0,0)
print(f.read())
f.close()
```

```
python programming
good
study
very very hot language
55
c/c++n programming
good
study
very very hot language
[Finished in 0.2s]
```

- 유니코드 인코딩 ,디코딩 / 바이트객체

```
mybytes = bytes([236, 130, 172,235,158, 145])
```

→ 바이트 타입 객체

```
mystr = mybytes.decode("utf-8")  
print(mystr)
```

→ 바이트객체를 **utf-8** 방식으로 디코딩

```
mystr = "파이썬 사랑"  
mytestbytes = mystr.encode("utf-8")  
print(mytestbytes)  
mystr1 = mytestbytes.decode("utf-8")  
print(mystr1)
```

→ 유니코드 문자 **utf-8** 방식으로 인코딩  
=> **Bytes** 객체로 변환

```
mytext = "python"  
f = open("mytext_utf8.txt", "wb")  
f.write(mytext.encode("utf-8"))  
f.close()
```

→ "python"문자열을 **utf-8**  
인코딩 방식으로 저장

```
f = open("mytext_utf16.txt","wb")  
f.write(mytext.encode("utf-16"))  
f.close()
```

→ "python"문자열을 **utf-16**  
인코딩 방식으로 저장

- 파일 내용 중 단어 구별

```
import sys
```

```
f = open("mytext.txt", "r") # "mytext.txt" == sys.argv[1]
```

```
mystr = f.read()
```

```
mylist = mystr.split(" ") →
```

읽어 들인 파일 내용을  
공백(" ") 문자로 문자  
열 구분하여 **list**

```
print(mylist)
```

```
mylist_cnv = {}
```

```
for s in mylist:
```

```
    tmp = s.strip() →
```

단어에 포함된 **\n or** 공백 제거

```
    if tmp in mylist_cnv:
```

```
        mylist_cnv[tmp] = mylist_cnv[tmp] + 1
```

```
    else:
```

```
        mylist_cnv[tmp] = 1
```

단어별로 사전에 포함시키되 사전에  
이미 단어가 있으면 사전 **value** 값  
**1** 증가

```
print(len(mylist_cnv))
```

```
print(mylist_cnv)
```

- 예외처리

```
try:  
    f = open("hellopython.txt","r")  
except FileNotFoundError:  
    print("no file")  
    raise SystemExit  
  
print("next code...")
```

- 버튼, Label

```
import tkinter
mywin = tkinter.Tk()
mywin.geometry("200x200")
bt = tkinter.Button(mywin, text = "Quit")
bt.grid(row = 0, column = 2)
lb_test = tkinter.Label(mywin, text="label Create")
lb_test.grid(row = 0, column = 1)

mywin.mainloop()
```

윈도우 창 생성

윈도우 창 크기 변경

버튼 생성

버튼 배치

**Label** 생성

**Label** 배치

생성된 윈도우 와 그에 포  
함된 자식부품의 상태를 지  
속적으로 갱신

화면의 행(**row**), 열(**column**)  
지정

- 버튼 동작 ( **command** 처리 )

```
import tkinter
```

```
def change_bg():
    btn.configure(background = "green")
```

```
mywin = tkinter.Tk()
mywin.geometry("{}x{}".format(200,200))
frame_up = tkinter.Frame(mywin, height = 60, width = 90, background = "blue")
frame_down = tkinter.Frame(mywin, height = 30, width = 90, background = "red")
frame_up.pack()
frame_down.pack()
```

```
btn = tkinter.Button(frame_down, text = "click", command = change_bg,
    foreground = "white", background = "black",
    activeforeground = "blue",
    activebackground = "#FF007F")
btn.pack()
mywin.mainloop()
```

foreground	버튼 라벨의 색 지정
background	버튼 배경색 지정
activeforeground	버튼이 눌러졌을때 버튼의 라벨색
activebackground	버튼이 눌러졌을때 버튼의 배경색

버튼 클릭시 동작할 콜백 함수 지정  
(콜백함수는 인수 가질수 없음)



- 제어변수 설정

```
import tkinter
```

```
def increase():  
    number.set(number.get()+1)
```

```
mywin = tkinter.Tk()  
mywin.geometry("200x200")  
frame = tkinter.Frame(mywin)  
frame.pack()
```

정수값 속성을 갖는 클래스 인스턴스 생성  
또한 자신의 값이 변경 되었을때 참조하는  
부품들에게 값이 전달되고 갱신되게 하는  
**제어 변수**

```
number = tkinter.IntVar(value = 0)  
button = tkinter.Button(frame, text = "increase", command = increase)  
button.pack()  
label = tkinter.Label(frame, text = "start", textvariable = number)  
label.pack()
```

```
mywin.mainloop()
```

**Textvariable :**  
**Label** 에서 제어변수를 참조할 수 있는 옵션

- **Entry ( 간단한 텍스트 입력 부품 )**

```
import tkinter
```

```
from tkinter import messagebox #tkinter패키지 의 messagebox 모듈 등록
```

```
def CancelClick():
```

```
    text_id.set("")
```

```
    text_pw.set("")
```

```
def okClick():
```

```
    try:
```

```
        id_value = text_id.get()
```

```
        pw_value = text_pw.get()
```

```
        messagebox.showinfo("입력데이터확인", "ID : {}, pw : {}".format(id_value, pw_value))
```

```
    except:
```

```
        messagebox.showwarning("입력오류", "정확히 입력 하세요")
```

```
        text_id.set("")
```

```
        text_pw.set("")
```

제어변수 값 읽어옴

제어변수 값 ""설정해  
클리어

- **Entry ( 간단한 텍스트 입력 부품 )**

```
mywin = tkinter.Tk()
mywin.geometry("300x300")
frame = tkinter.Frame(mywin)
frame.pack()
```

```
text_id = tkinter.StringVar(value="")
text_pw = tkinter.IntVar(value = 0)
```

```
lb1 = tkinter.Label(frame, text = "ID(문자열 입력)")
lb1.grid(row = 0, column = 0)
txt = tkinter.Entry(frame, textvariable = text_id)
txt.grid(row = 0, column = 1)
```

**Entry** 생성과 동시 문자열 속성을 갖는 **text\_id** 제어 변수 설정

```
lb2 = tkinter.Label(frame, text = "pw(정수입력)")
lb2.grid(row = 1, column = 0)
txt_pw = tkinter.Entry(frame, textvariable = text_pw, show="*")
txt_pw.grid(row = 1, column = 1)
text_pw.set("")
```

**Entry** 생성과 동시 정수 속성을 갖는 **text\_pw** 제어 변수 설정

```
btn_ok = tkinter.Button(frame, text = "입력 확인", command = okClick)
btn_ok.grid(row = 2, column = 1)
```

```
btn_cancel = tkinter.Button(frame, text = "입력 취소", command = CancelClick)
btn_cancel.grid(row = 2, column = 2)
mywin.mainloop()
```

- **Canvas**

```
import tkinter
import random
```

```
def randxy():
    a = random.randint(1,250)
    return a
```

```
def click():
    cvs.create_line(randxy(), randxy(), randxy(), randxy())
```

```
mywin = tkinter.Tk()
cvs = tkinter.Canvas(mywin, width=250, height = 250)
cvs.pack()
```

```
button = tkinter.Button(text = "click", command = click)
cvs.create_window(150,200, window = button)
```

```
label = tkinter.Label(text = "Hellow")
cvs.create_window(100,200, window=label)
```

```
mywin.mainloop()
```



**Canvas**에 버튼 클릭시 임  
의의 **line** 생성

- **Canvas**

```
import tkinter
import time
mywin = tkinter.Tk()
cvs = tkinter.Canvas(mywin, width=600, height = 400)
cvs.pack()
x = 100
y = 100
img = tkinter.PhotoImage(file = "beachball.png")
ball1 = cvs.create_image(x,x, image = img)
```

Canvas에 이미지 출력

```
for x in range(0,60):
```

```
    cvs.move(ball1, 9 ,5)
```

```
    mywin.update()
```

```
    time.sleep(0.05)
```

ball1을 (9,5) 만큼 이동

화면 갱신

```
mywin.mainloop()
```

\* 문제 - 화면에 버튼 추가 , 버튼 클릭시 **img** 조금씩 이동 하는 프로그램 구현

- 튜플, 사전 가변 인수 처리 **format** 메소드

```
args =(5,6,2)
mystr = ""line {}
line {}
line {}
"".format(*args)
```

튜플을 받을때

```
print(mystr)
```

```
dic_args = {"orange":1, "apple":5, "banana":8}
```

```
mystrdic = ""line {orange}
line {apple}
line {banana}
"".format(**dic_args)
```

사전을 받을때

```
print(mystrdic)
```

- **grid**

```
import tkinter
from tkinter import messagebox
```

```
def btn1clicked():
    messagebox.showinfo(title = "Btn1", detail = "btn1 clicked")
```

```
mywin = tkinter.Tk()
myframe = tkinter.Frame(mywin)
myframe.pack()
button1 = tkinter.Button(text="click1", padx = 25, pady = 10, command = btn1clicked)
button1.grid(in_=myframe, row = 0, column = 0, padx =25, pady=10)
```

```
button2 = tkinter.Button(myframe , text = "====button2====")
button2.grid(row = 1, column = 1)
```

```
button3 = tkinter.Button(myframe, text = "button3")
button3.grid(row = 2, column = 1, sticky = 'we')
mywin.mainloop()
```

- 주의 : 같은 **frame** 내에서 **pack()** 메서드와 혼용해서 사용할 수 없다.

- **bind** 메소드로 이벤트 묶기

```
import tkinter
```

```
def callback():  
    mywin.title("Hello python")
```

```
mywin = tkinter.Tk()  
mywin.geometry("{}x{}".format(500,300))  
frame = tkinter.Frame(mywin, padx = 100, pady = 50)  
frame.pack()
```

```
button = tkinter.Button(frame, text = "Button click")  
button.pack()
```

```
label = tkinter.Label(frame, text = "Label click")  
label.pack()
```

```
button.bind("<ButtonPress-1>", lambda e: callback())  
button.bind("<Double-1>", lambda e: mywin.title("mouse Double click"))  
button.bind("<ButtonPress-3>", lambda e: mywin.title("mouse Right click"))
```

```
label.bind("<Double-3>", lambda e: mywin.title("label event"))
```

```
mywin.mainloop()
```

첫번째 인수 : 부품에 묶어줄 이벤트 이름  
두번째 인수 : 이벤트 핸들러(함수또는 람다함수), 이벤트에 대한 정보가 람다함수의 첫번째 인수로 전달



- **bind** 메소드로 이벤트 묶기

```
import tkinter
from tkinter import messagebox

def test(e):
    a = ""
    char : {}
    delta : {}
    hight : {}
    keycode : {}
    keysym : {}
    keysym_num : {}
    num : {}
    time : {}
    widget : {}
    width : {}
    x : {}
    y : {}
    x_root : {}
    y_root : {}"".format(e.char,
```

```
e.delta,
e.height,
e.keycode,
e.keysym,
e.keysym_num,
e.num,
e.time,
e.widget,
e.width,
e.x,
e.y,
e.x_root,
e.y_root)

info.set(a)
```

- **bind** 메소드로 이벤트 묶기

```
def mouseclick(e):  
    print("클릭 위치", e.x, e.y)  
    messagebox.showinfo(title = "mousexy", detail = "클릭 위치 {} {}".format(e.x, e.y))  
  
mywin = tkinter.Tk()  
info = tkinter.StringVar()  
  
frame = tkinter.Frame(mywin, width = 500, height = 400, padx = 100, bg = "blue")  
frame.grid()  
  
button = tkinter.Button(mywin, text = "Test")  
button.grid()  
  
label_title = tkinter.Label(frame, text = "===== test\  
Event===== ", justify = "left")  
label_title.grid()  
  
label = tkinter.Label(frame, textvariable = info, justify = "left")  
label.grid()  
print(id(label))  
  
frame.bind("<ButtonPress-1>", mouseclick)  
mywin.bind("<MouseWheel>", lambda e : test(e))  
mywin.bind("<KeyPress>", test)  
mywin.bind("<Motion>", test)
```

마우스 왼쪽 클릭  
좌표 출력

- **마우스 이벤트 처리 ( Canvas 동작 - 작은 원 )**

```
import tkinter
from tkinter import messagebox

def CallBack_Mouse(event):
    cvs.create_oval(event.x-10, event.y-10, event.x+10, event.y+10)

def Delete_pic(event):
    cvs.delete("all")

mywind = tkinter.Tk()

cvs = tkinter.Canvas(mywind, width = 500, height = 400)
cvs.pack()
cvs.bind("<Button-1>", CallBack_Mouse)
cvs.bind("<B1-Motion>", CallBack_Mouse)
cvs.bind("<Button-3>", Delete_pic)

mywind.mainloop()
```

- **마우스 이벤트 처리 ( Canvas 동작 - 마우스로 임의의 선 그리기 )**

```
import tkinter
from tkinter import messagebox
x0 = 0
y0 = 0

def Mouse_down(event):
    global x0, y0
    x0, y0 = event.x, event.y

def Mouse_draw(event):
    global x0, y0
    cvs.create_line(x0,y0,event.x,event.y)
    x0, y0 = event.x, event.y

def Mouse_up(event):
    global x0, y0
    if (x0,y0) == (event.x, event.y):
        cvs.create_line(x0,y0, x0+1,y0+1)
```

```
def Delete_pic(event):
    cvs.delete("all")

mywind = tkinter.Tk()

cvs = tkinter.Canvas(mywind, width = 500, height = 400)
cvs.pack()
cvs.bind("<Button-1>", Mouse_down)
cvs.bind("<B1-Motion>", Mouse_draw)
cvs.bind("<ButtonRelease-1>", Mouse_up)
cvs.bind("<Button-3>", Delete_pic)

mywind.mainloop()
```

- 마우스 이벤트 처리 ( 버튼 추가 - **Canvas** 클리어 기능 )

```
import tkinter
from tkinter import messagebox

def mouseclick(e):
    print("클릭위치", e.x, e.y)
    global m_x, m_y
    m_x, m_y = e.x, e.y

def mousedraw(e):
    global m_x, m_y
    cvs.create_line(m_x, m_y, e.x, e.y)
    m_x, m_y = e.x, e.y

def mymouseup(e):
    global m_x, m_y
    if (m_x, m_y) == (e.x, e.y):
        cvs.create_line(m_x, m_y,
            m_x+1, m_y+1)

def delete_pic():
    cvs.delete("all")
```

```
mywin = tkinter.Tk()

cvs = tkinter.Canvas(mywin, width = 500, height = 400, bg =
    "blue")
cvs.grid()

button = tkinter.Button(mywin, text = "clear", command =
    delete_pic, foreground = "white", background = "black")
button.grid()

cvs.bind("<ButtonPress-1>", mouseclick)
cvs.bind("<ButtonRelease-1>", mymouseup)
cvs.bind("<B1-Motion>", mousedraw)

mywin.mainloop()
```

- **Checkbutton** 처리

```
import tkinter
from tkinter import messagebox

def btnresult():
    result = []
    print("check_A status : ", check_A.get())
    print("check_B status : ", check_B.get())
    print("check_C status : ", check_C.get())
    if check_A.get() == "on":
        result.append("AI 기술")
    if check_B.get() == "on":
        result.append("자율 주행")
    if check_C.get() == "on":
        result.append("5G 통신")

    selstr = " / ".join(result)
    messagebox.showinfo(title = "select tech", detail = selstr)
```

**CheckButton** 에 선택된 문자열  
을 합하여 메시지박스로 출력

```
mywind = tkinter.Tk()
mywind.geometry("300x300")
frame = tkinter.Frame(mywind)
frame.pack()

check_A = tkinter.StringVar()
check_B = tkinter.StringVar()
check_C = tkinter.StringVar()

check_A.set("off")
check_B.set("off")
check_C.set("off")
```

- **Checkbox 처리**

```
label_tile = tkinter.Label(frame, text = "관심 기술 분야")
label_tile.grid(row=0, column = 0)

cb_A = tkinter.Checkbutton(frame, text = "AI 기술", variable = check_A, onvalue="on", offvalue="off")
cb_A.grid(row=1, column = 0)

cb_B = tkinter.Checkbutton(frame, text = "자율 주행", variable = check_B, onvalue="on", offvalue = "off")
cb_B.grid(row=1, column = 1)

cb_c = tkinter.Checkbutton(frame, text = "5G 통신", variable = check_C, onvalue="on", offvalue = "off")
cb_c.grid(row=1, column = 2)

label = tkinter.Label(frame)
label.grid(row=2,column = 0)

buttonTest = tkinter.Button(frame, text = "result", command = btnresult)
buttonTest.grid(sticky = 'we')

mywind.mainloop()
```

두 옵션 값 중 하나의 값을 **variable** 옵션을 통해 제어 변수에 전달

- **Checkbox 처리 – offvalue 생략 시**

```
import tkinter

def result():
    r = ""
    for x in vargroup:
        if x.get() != '0':
            r = r + " " + x.get()
    var_result.set("I have checked" + r + ".")

def check_clear():
    for x in vargroup:
        x.set(0)

mywind = tkinter.Tk()
frame = tkinter.Frame(mywind)
frame.pack()

var1 = tkinter.StringVar()
var2 = tkinter.StringVar()
var3 = tkinter.StringVar()
var4 = tkinter.StringVar()
var5 = tkinter.StringVar()
var6 = tkinter.StringVar()
var7 = tkinter.StringVar()
```

제어 변수 리스트 그룹화

```
var_result.set("result")
var_result = tkinter.StringVar()
vargroup = [ var1, var2, var3, var4, var5, var6, var7 ]

for x in vargroup:
    x.set(0)
```



- **Checkbox 처리 - offvalue** 생략 시

```
check1 = tkinter.Checkbutton(frame, text = "Sun", variable = var1, onvalue = "Sun")
check2 = tkinter.Checkbutton(frame, text = "Mon", variable = var2, onvalue = "Mon")
check3 = tkinter.Checkbutton(frame, text = "Tue", variable = var3, onvalue = "Tue")
check4 = tkinter.Checkbutton(frame, text = "Wen", variable = var4, onvalue = "Wen")
check5 = tkinter.Checkbutton(frame, text = "Thu", variable = var5, onvalue = "Thu")
check6 = tkinter.Checkbutton(frame, text = "Fri", variable = var6, onvalue = "Fri")
check7 = tkinter.Checkbutton(frame, text = "Sat", variable = var7, onvalue = "Sat")

button2 = tkinter.Button(frame, text = "clear", command = check_clear)
button2.pack(side = "bottom")

button1 = tkinter.Button(frame, text = "result btn", command = result)
button1.pack(side = "bottom")

label = tkinter.Label(frame, textvariable = var_result, background = "white")
label.pack()

checkgroup = [check1, check2, check3, check4, check5, check6, check7]
for x in checkgroup:
    x.pack(side = "left")

mywind.mainloop()
```

**Offvalue** 생략 시 기본값 **0**이 전달되며 제어변수 속성이 문자열임으로 **'0'**문자로 전달

**frame** 맨 아래서 버튼 부품 배치

- **Text** 부품 ( 메모장이나 에디터 역할 )

```
import tkinter
from tkinter import filedialog

mywind = tkinter.Tk()
mywind.geometry("500x300")

text = tkinter.Text(mywind, width = 500, height = 300)
text.pack()

mystr = None

def openfile():
    f = filedialog.askopenfile("rt")
    mystr_read = f.read()
    text.insert("1.0",mystr_read)

def savefile():
    global mystr
    mystr = text.get("1.0","end") #텍스트 파일 전체 읽기
    print(type(mystr))
    if mystr == '\n':
        print("empty!!")
        return
    else:
        save_f = filedialog.asksaveasfile("wt")
        save_f.write(mystr)
        save_f.close()
```

메뉴에서 **open** 클릭시  
호출 되는 함수

**"1.0"** => 1행의 0번째 칸부터 읽  
어들인 파일 내용을 삽입

**Text** 부품에 작성된 내  
용을 처음부터 끝까지  
읽어 들임

읽어 들인 내용이 있으면  
새로운 파일 이름으로 파  
일 내용저장

- **Text** 부품 ( 메모장이나 에디터 역할 )

```
def EditClear():  
    text.delete("1.0","end")  
  
def exit():  
    mywind.quit()  
  
menubar = tkinter.Menu(mywind)  
mywind['menu'] = menubar  
  
menu_file = tkinter.Menu(menubar, tearoff = 0)  
menu_edit = tkinter.Menu(menubar, tearoff = 0)  
menu_help = tkinter.Menu(menubar, tearoff = 0)  
  
menubar.add_cascade(menu = menu_file, label = "File")  
menubar.add_cascade(menu = menu_edit, label = "Edit")  
menubar.add_cascade(menu = menu_help, label = "Help")  
  
menu_file.add_command(label = "open", command = openfile)  
menu_file.add_command(label = "save as", command = savefile)  
menu_file.add_command(label = "Edit Clear", command = EditClear)  
menu_file.add_command(label = "Exit", command = exit)  
  
mywind.mainloop()
```

메뉴에서 **Edit Clear** 클릭시  
호출 되는 함수, 전체 내용 삭제

메뉴 삽입할 메뉴바 생성 및  
삽입

메뉴바에 메인 메뉴 삽입

메뉴 파일에 서브 메뉴 삽입  
및 **command** 지정

# Thank you

(주)한컴MDS [www.hancommds.com](http://www.hancommds.com)

본사 13493 경기도 성남시 분당구 대왕판교로 644번길 49 한컴타워 3,4층 031-627-3000

연구소 13487 경기도 성남시 분당구 판교로 228번길 17 판교세븐벤처밸리 2단지 1동 9층 031-600-5000