PYTHON PROGRAMMING

GUI PROGRAMMING

ABOUT GUI Tkinter

ABOUT GUI TKINTER

◆ Tkinter 살펴보기

- Python 기본 내장된 **GUI용 파이썬 표준 라이브러리**
- 타 GUI 프레임워크나 툴킷에 비해 **지원 위젯 부족**
- UI도 그렇게 **예쁘지 않음**

- Tcl/Tk를 파이썬에 사용할 수 있도록 한 경량 **GUI 모듈**

  → **유닉스계열**에서 사용되던 Tcl/Tk 위에 **객체지향계층** 입힌 것

  → Tcl : Tool Command Language 약자, 프로그래밍 언어

  → Tk : 크로스 플랫폼에 사용되는 일종의 GUI 툴킷(Tool Kit)

ABOUT GUI TKINTER

◆ Tkinter 살펴보기

▪ GUI 구성

Main Window

+

UI(User Interface) 요소

Widget

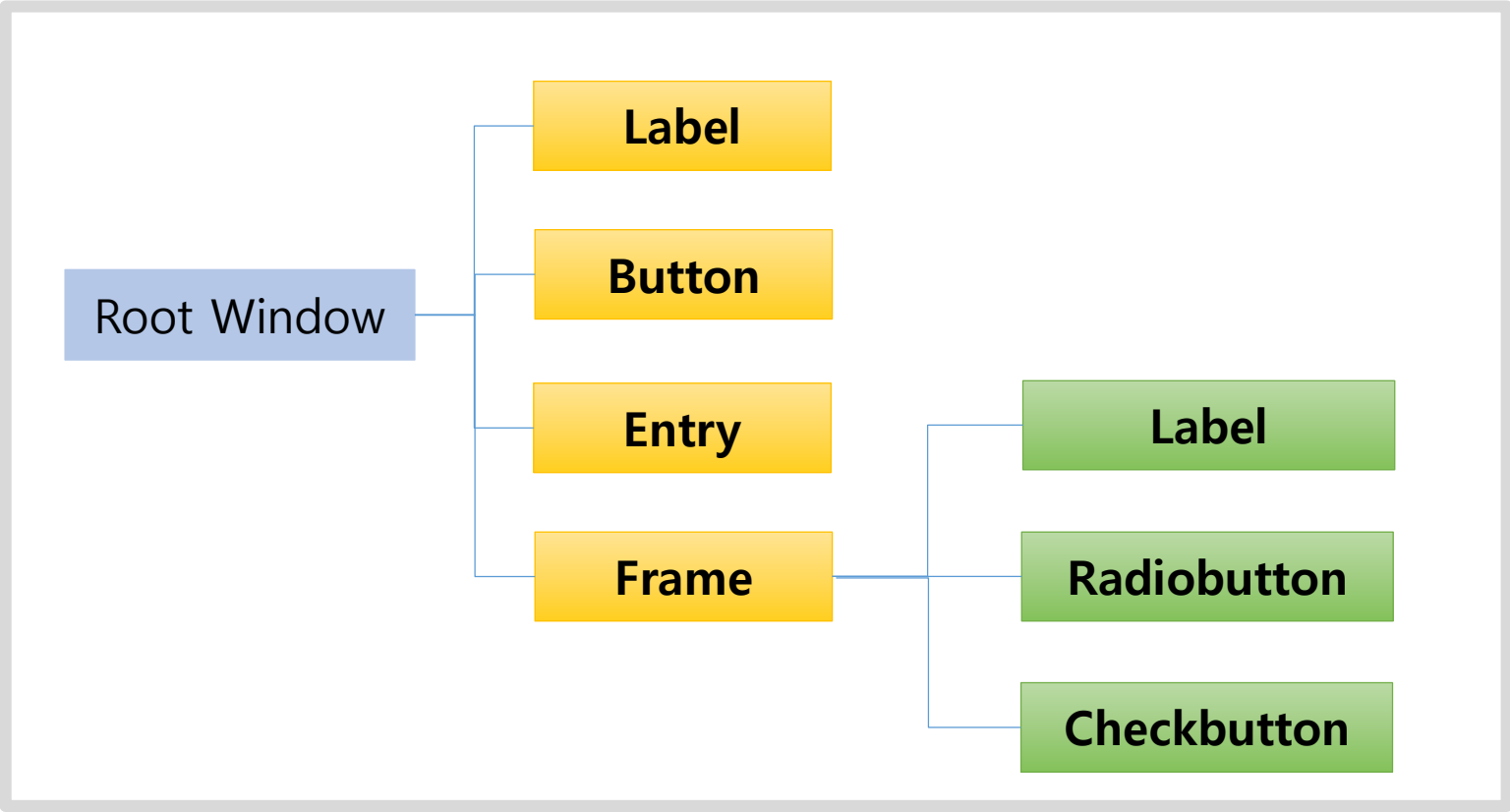
+

User Event

ABOUT GUI TKINTER

◆ Tkinter 살펴보기

- **UI(User Interface) – 계층적 구조**



ABOUT GUI TKINTER

◆ Tkinter 살펴보기

▪ 위젯(Widget)

- 사용자에게 데이터 및 의사를 **전달 받는 통로**
- 사용자에게 정보를 **보여주는 통로**
- 클래스로 위젯을 보여주기위한 **많은 속성들 존재**
- **Geometry Manager** 사용하여 각 위젯 **위치 결정**

- **속성 / 특징 설정**

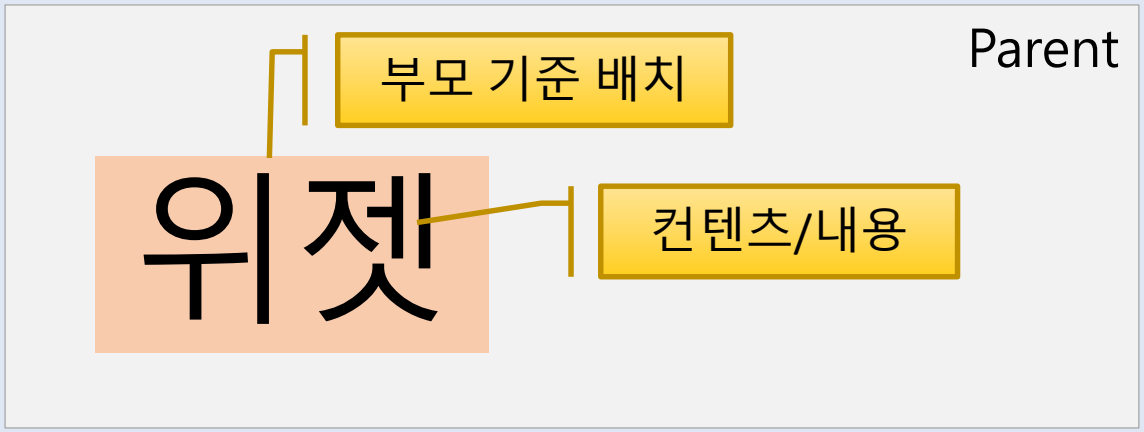
```
위젯명[ " 속성명" ]  = 속성값
```

ABOUT GUI TKINTER

◆ Tkinter 살펴보기

- **위젯(Widget)**

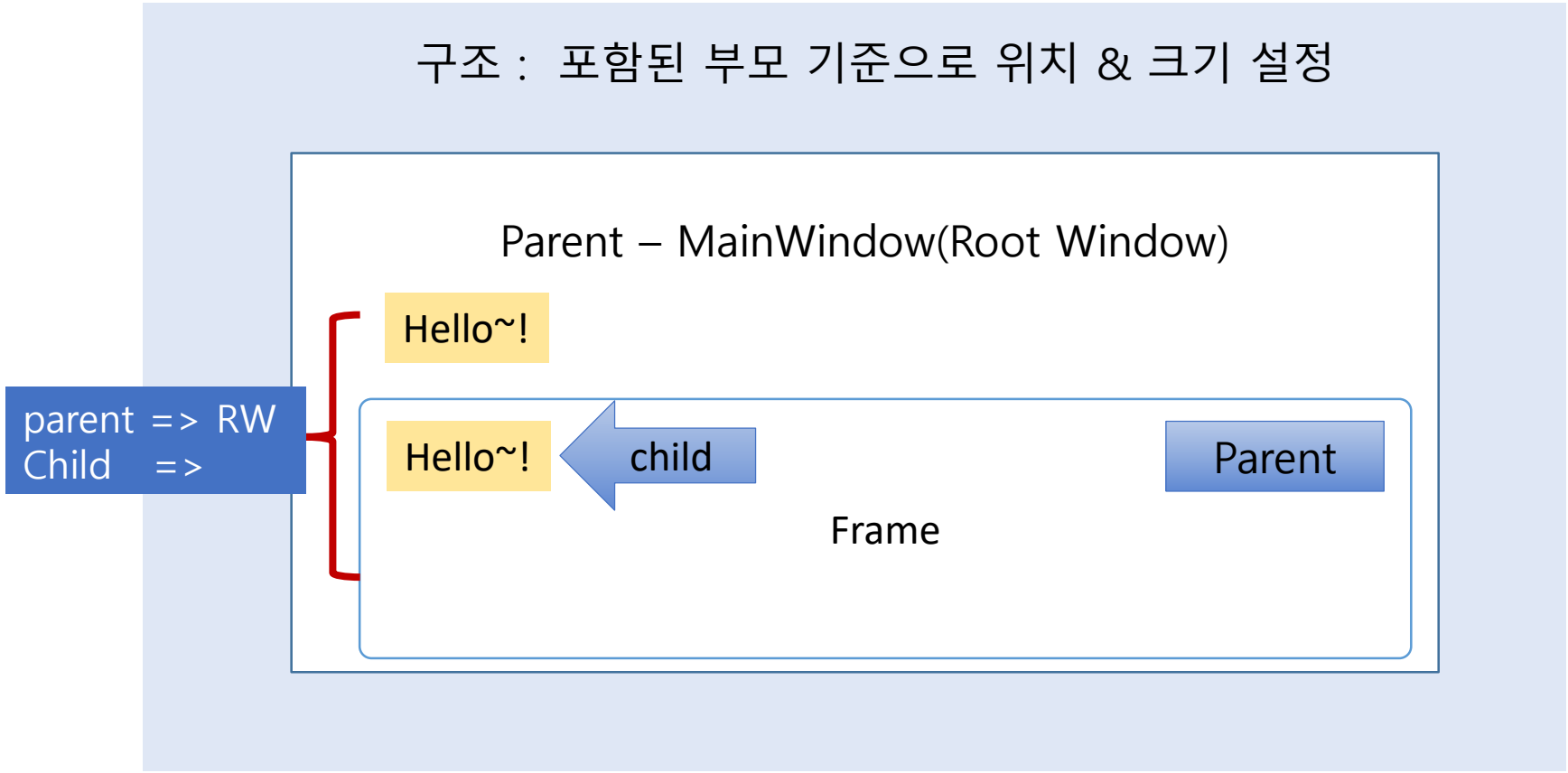
구조 : 포함된 부모 기준으로 위치 & 크기 설정



ABOUT GUI TKINTER

◆ Tkinter 살펴보기

- **위젯(Widget)**



ABOUT GUI TKINTER

◆ Tkinter 살펴보기

- Widget 기능에 따른 분류

단순위젯	• 하나의 기능 수행
컨테이너 위젯	• 다른 위젯을 담는 위젯
입력 위젯	• 사용자에게 데이터 및 의사 받아들이는 위젯
출력 위젯	• 사용자에게 정보 및 데이터를 표시하는 위젯
복합 위젯	• 여러 개 데이터 담아서 사용자에게 보여주는 위젯

ABOUT GUI TKINTER

◆ Tkinter 살펴보기

- Widget 포함 여부에 따른 분류

컨테이너 위젯	• Frame, Toplevel, Label 등과 같이 • **다른 위젯을 내부에 담을 수 있는 위젯**
단순 위젯	• Button, Canvas, Checkbutton, Entry, Label, Message 등 • 하나의 기능 수행

ABOUT GUI TKINTER

◆ Tkinter 살펴보기

- 출력 Widget

Label	• 텍스트 / 이미지 표시
Message	• 텍스트 표시, Label과 달리 자동 래핑 가능

ABOUT GUI TKINTER

◆ Tkinter 살펴보기

- 입력 Widget

Button	• 단순 버튼 ➔ 동작/액션
CheckButton	• 체크박스 ➔ 선택
RadioButton	• 옵션 버튼 ➔ 선택
Scale	• 슬라이스 바 ➔ 숫자 값
Scrollbar	• 스크롤 바 ➔ 숫자 값
Entry	• 한 줄 글자 입력 받는 텍스트 박스 또는 텍스트 필드
Text	• 멀티 라인 텍스트 박스, 서식화된 텍스트 출력

ABOUT GUI TKINTER

◆ Tkinter 살펴보기

- 복합 Widget

ListBox	리스트 박스
Menu	메뉴 Pane
Menubutton	메뉴 버튼
Toplevel	새 윈도우 및 다이얼로그 생성할 때 사용
Frame	**컨테이너 위젯. 다른 위젯들을 그룹화**할 때 사용
Canvas	**그림 그릴 수 있는 요소, 커스텀 위젯** 생성 사용

ABOUT GUI TKINTER

◆ Tkinter 살펴보기

- Geometry Manager

→ 화면에 Widget 배치하는 역할 수행

→ 배치 방법

Place	• 절대 좌표 배치	위젯.place()
Pack	• 부모 위젯에 모두 패킹, 불필요 공값 없앰	위젯.pack()
Grid	• 위젯들을 테이블 레이아웃에 배치	위젯.grid(row, col)

GUI Tkinter PROGRAMMING

GUI TKINTER PROGRAMMING

◆ Tkinter 사용 준비

- 설치

```
→ conda install -c conda-forge tk
```

- 사용

```
→ from tknter impot *
```

GUI TKINTER PROGRAMMING

◆ Main Window

▪ 생성

```
##- 모듈 로딩
from tkinter import *


##- 윈도우 창 생성
mainWin=Tk()


##- 윈도우에서 발생하는 이벤트 메시지 수신
##- 윈도우 종료될때까지 실행
mainWin.mainloop()
```

GUI TKINTER PROGRAMMING

◆ Main Window

▪ 설정

```
##- 모듈 로딩
from tkinter import *

##- 윈도우 창 생성
mainWin=Tk()

##- 윈도우 설정
mainWin.title("MY APP")                          ##- 타이틀 명
mainWin.geometry("300x700+100+100")     ##- 윈도우 크기 설정,
mainWin.resizable(False, False)                  ##- 크기 조절 여부, 상하, 좌우

##- 윈도우에서 발생하는 이벤트 메시지 수신
mainWin.mainloop()
```

GUI TKINTER PROGRAMMING

◆ 출력 Widget

▪ Label

→ Text, Image 출력 위젯

```
Label (   window,
          text     = "쓸 내용",
          font     = ("글꼴",크기),
          fg       = "글씨색깔",
          bg       = "뒷배경색깔" ,
          width    = 넓이,
          height = 높이,
          anchor ="기준점" )
```

GUI TKINTER PROGRAMMING

◆ 출력 Widget

▪ Label

텍스트 속성	의미	기본값	속성
text	라벨에 표시할 문자열	-	-
textvariable	라벨에 표시할 문자열을 가져올 변수	-	-
anchor	문자열 또는 이미지의 위치	center	n, ne, e, se, s, sw, w, nw, center
justify	문자열이 여러 줄 일 경우 정렬 방법	center	center, left, right
wraplength	자동 줄내림 설정 너비	0	상수

GUI TKINTER PROGRAMMING

◆ 출력 Widget

▪ Label

형태 속성	의미	기본값	속성
width	라벨의 너비	0	상수
height	라벨의 높이	0	상수
relief	테두리 모양	flat	flat, groove, raised, ridge, solid, sunken
borderwidth=bd	테두리 두께	2	상수
background=bg	배경 색상	SystemButtonFace	color
padx	테두리와 내용 가로 여백	1	상수
pady	테두리와 내용 세로 여백	1	상수

GUI TKINTER PROGRAMMING

◆ 출력 Widget

▪ Label

이미지 속성	의미	기본값	속성
bitmap	포함할 기본 이미지	-	info, warning, error question, questhead, hourglass, gray12, gray25, gray50, gray75
image	포함할 임의 이미지	-	-
compound	문자열과 이미지를 동시에 표시할 때 이미지 위치	none	bottom, center, left, none, right, top
font	라벨의 문자열 글꼴 설정	TkDefaultFont	font
cursor	라벨의 마우스 커서 모양	-	커서 속성

GUI TKINTER PROGRAMMING

◆ 출력 Widget

▪ Label

→ 화면 배치 : 라벨객체변수명.**pack(side = LEFT)**

→ 화면 배치 : 라벨객체변수명.**pack(side = RIGHT)**

GUI TKINTER PROGRAMMING

◆ 출력 Widget

▪ Label - Text

```
##- 모듈 로딩
from tkinter import *

##- 윈도우 창 생성
mainWin=Tk()

##- 라벨 인스턴스 생성
label=Label(mainWin, text="NAME")

##- 라벨 화면 배치
label.pack()

##- 윈도우에서 발생하는 이벤트 메시지 수신
mainWin.mainloop()
```

GUI TKINTER PROGRAMMING

◆ 출력 Widget

▪ Label - Text

```
# 메시지 출력하는 라벨 생성
label=Label(  mainWin,
              text="Hello GUI",
              width=30,
              height=5,
              bg='blue',
              fg='white',
              relief='solid' )

##- 라벨 화면 배치
label.pack()
```

GUI TKINTER PROGRAMMING

◆ 출력 Widget

▪ Label - Image

```
##- 이미지 파일 확장자에 따른 이미지 데이터 로딩
IMG_PATH = '../Image/cat.jpg'
ext = os.path.splitext(IMG_PATH)[1]

if ext.lower() in ['.png', '.gif', '.pgm', '.ppm']:
    photo=PhotoImage( file = IMG_PATH )
else:
    img = Image.open(IMG_PATH)
    photo = ImageTk.PhotoImage(img)

##- 이미지 라벨 인스턴스 생성
label=Label( mainWin, image=photo )

##- 라벨 화면 배치
label.pack()
```

GUI TKINTER PROGRAMMING

◆ 입력 Widget

▪ Button

→ Text, Image 출력 위젯

```
Button (  window,
          text              = " 버튼 위 글자",
          overrelief        = ("글꼴",크기),
          fg                = "글씨색깔",
          bg                = "뒷배경색깔" ,
          width             = 넓이,
          height            = 높이,
          repeatdelay    = 대기시간 ,
          repeatinterval= 반복시간)
```

GUI TKINTER PROGRAMMING

◆ 입력 Widget

▪ Button

텍스트 설정	의미	기본값	속성
text	버튼에 표시할 문자열	-	-
textvariable	버튼에 표시할 문자열을 가져올 변수	-	-
anchor	버튼안의 문자열 또는 이미지의 위치	center	n, ne, e, se, s, sw, w, nw center
justify	문자열이 여러 줄 일 경우 정렬 방법	center	center, left, right
wraplength	자동 줄내림 설정 너비	0	상수

GUI TKINTER PROGRAMMING

◆ 입력 Widget

▪ Button

형태 설정	의미	기본값	속성
width	• 버튼의 너비	0	상수
height	• 버튼의 높이	0	상수
relief	• 버튼의 테두리 모양	flat	flat, groove, raised, ridge, solid, sunken
overrelief	• 버튼에 마우스 올렸을 때 버튼 테두리 모양	raised	flat, groove, raised, ridge, solid, sunken
borderwidth	• 버튼의 테두리 두께	2	상수

GUI TKINTER PROGRAMMING

◆ 입력 Widget

▪ Button

형태 설정	의미	기본값	속성
background=bg	• 버튼의 배경 색상	SystemButtonFace	color
foreground=fg	• 버튼의 문자열 색상	SystemButtonFace	color
padx	• 버튼의 테두리와 내용의 가로 여백	1	상수
pady	• 버튼의 테두리와 내용의 세로 여백	1	상수

GUI TKINTER PROGRAMMING

◆ 입력 Widget

▪ Button

동작 설정	의미	기본값	속성
takefocus	• Tab 키 이용하여 위젯 이동 허용 여부	True	Boolean
command	• 버튼이 active 상태일 때 실행하는 메소드(함수)	-	메소드함수
repeatdelay	• 버튼 눌러진 상태 명령어 실행까지 대기시간	0	상수(ms)
repeatinterval	• 버튼 눌러진 상태 명령어 실행 반복 시간	0	상수(ms)

GUI TKINTER PROGRAMMING

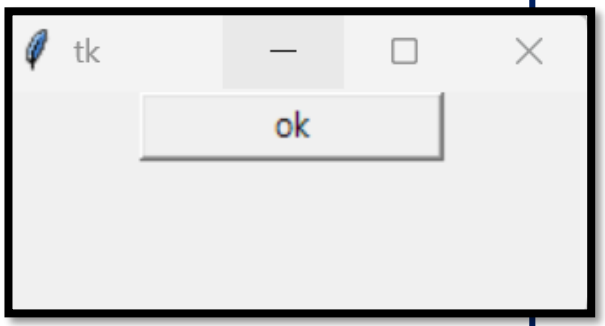
◆ 입력 Widget

▪ Button

```
##- 윈도우 창 생성
mainWin=Tk()


# 버튼 객체 생성
okBTN=Button( mainWin,
                    text='ok',
                    overrelief="sunken",    #  마우스 올렸을 때 버튼 테두리 모양
                    width=15,
                    repeatdelay=1000,     # 눌러진 상태. 명령어 실행까지 대기 시간
                    repeatinterval=100)    # 눌러진 상태. 명령어 실행의 반복 시간

okBTN.pack()
```



GUI Tkinter 프로그래밍 2

◆ 입력 위젯

▪ Entry

→ 1줄 입력, 텍스트 박스 또는 텍스트 필드

문자열 설정	의미	기본값	속성
show	• 기입창에 표시되는 문자	-	문자
textvariable	• 기입창에 표시할 문자열을 가져올 변수	-	-
justify	• 입력 문자열이 여러 줄 일 경우 정렬 방법	center	center, left, right

GUI Tkinter 프로그래밍 2

◆ 입력 위젯

▪ Entry

형태 설정 속성	의미	기본값	속성
width	• 기입창의 너비	0	상수
relief	• 기입창의 테두리 모양	flat	flat, groove, raised, ridge, solid, sunken
borderwidth=bd	• 기입창의 테두리 두께	2	상수
background=**bg**	• 기입창 배경 색상	SystemButtonFace	color
foreground=**fg**	• 기입창 문자열 색상	SystemButtonFace	color
insertwidth	• 기입창 키보드 커서 너비	2	상수
insertborderwidth	• 기입창 키보드 커서 테두리 두께	0	상수

GUI Tkinter 프로그래밍 2

◆ 입력 위젯

▪ Entry

이름	의미	기본값	속성
insertbackground	기입창 키보드 커서 색상	SystemWindowText	color
selectborderwidth	기입창 문자열 블록처리 테두리 두께	0	상수
selectbackground	기입창 문자열 블록처리 배경 색상	SystemHighlight	color
selectforeground	기입창 문자열 블록처리 문자열 색상	SystemHighlight	color

GUI Tkinter 프로그래밍 2

◆ 입력 위젯

▪ Entry - Text 입력

```
##- 1줄 텍스트 필드 객체 생성
msg=Entry( mainWin)

##- 1줄 텍스트 필드  화면 배치
msg.pack(side='right')

##- 1줄 텍스트 필드에 포커스 설정
msg.focus()

##- 윈도우에서 발생하는 이벤트 메시지 수신
mainWin.mainloop()
```

GUI Tkinter 프로그래밍 2

◆ 이벤트 이해

▪ 이벤트란

→ 무언가가 일어났다는 신호

→ 마우스 클릭, 키보드 입력, 창 크기 변경, 타이머 만료, 네트워크 응답처럼 **사용자·시스템·프로그램이 발생시키는 모든 사건**

GUI Tkinter 프로그래밍 2

◆ 이벤트 이해

▪ 이벤트 요소들

→ 이벤트 소스: 버튼·윈도우 같은 위젯

→ 이벤트 타입: 클릭, 키입력, 포커스 변화, 리사이즈 등.

→ 이벤트 핸들러(리스너): 특정 이벤트 발생 시 실행할 함수(콜백).

→ 이벤트 큐 : OS의 이벤트 저장 공간

→ 이벤트 루프: GUI 프레임워크에서 이벤트 꺼내 핸들러에 전달 루틴.

GUI Tkinter 프로그래밍 2

◆ 이벤트 처리

▪ 이벤트 Binding

→ 위젯들의 이벤트 < == > 이벤트 발생 시 실행할 함수 연결

→ [형식] 라벨객체변수명.**bind(** "이벤트", 처리함수명 **)**

GUI Tkinter 프로그래밍 2

◆ 이벤트 처리

▪ 다양한 이벤트

마우스

이름	의미
<Button-1>	마우스 왼쪽 버튼을 누를 때
<Button-2>	마우스 휠 버튼을 누를 때
<Button-3>	마우스 오른쪽 버튼을 누를 때
<Button-4>	스크롤 업
<Button-5>	스크롤 다운
<MouseWheel>	마우스 휠 이동

GUI Tkinter 프로그래밍 2

◆ 이벤트 처리

▪ 다양한 이벤트

마우스이벤트

이름	의미
〈ButtonRelease-1〉	마우스 왼쪽 버튼을 뗄 때
〈ButtonRelease-2〉	마우스 휠 버튼을 뗄 때
〈ButtonRelease-3〉	마우스 오른쪽 버튼을 뗄 때

이름	의미
〈Double-Button-1〉	마우스 왼쪽 버튼을 더블 클릭할 때
〈Double-Button-2〉	마우스 휠 버튼을 더블 클릭할 때
〈Double-Button-3〉	마우스 오른쪽 버튼을 더블 클릭할 때

GUI Tkinter 프로그래밍 2

◆ 이벤트 처리

▪ 다양한 이벤트

위젯 이벤트

이름	의미
<Enter>	위젯 안으로 마우스 포인터가 들어왔을 때
<Leave>	위젯 밖으로 마우스 포인터가 나갔을 때
<FocusIn>	위젯 안으로 Tab 키를 이용하여 들어왔을 때
<FocusOut>	위젯 밖으로 Tab 키를 이용하여 나갔을 때
<Configure>	위젯의 모양이 수정되었을 때

GUI Tkinter 프로그래밍 2

◆ 이벤트 처리

▪ 다양한 이벤트

키 이벤트

이름	의미
<Key>	특정 키가 입력되었을 때
<Return>	Enter 키가 입력되었을 때
<Cancel>	Break 키가 입력되었을 때
<Pause>	Pause 키가 입력되었을 때
<Backspace>	백스페이스 키가 입력되었을 때
<Caps_Lock>	캡스 락 키가 입력되었을 때
<Escape>	이스케이프 키가 입력되었을 때
<Home>	Home 키가 입력되었을 때

GUI Tkinter 프로그래밍 2

◆ 이벤트 처리

▪ 다양한 이벤트

키 이벤트

이름	의미
<End>	End 키가 입력되었을 때
<Print>	Print 키가 입력되었을 때
<Insert>	Insert 키가 입력되었을 때
<Delete>	Delete 키가 입력되었을 때
<Prior>	Page UP 키가 입력되었을 때
<Up>	윗쪽 방향키가 입력되었을 때
<Down>	아랫쪽 방향키가 입력되었을 때
<Right>	오른쪽 방향키가 입력되었을 때
<Left>	왼쪽 방향키가 입력되었을 때

GUI Tkinter 프로그래밍 2

◆ 이벤트 처리

▪ 마우스 이벤트 처리

```
##- 이벤트 처리 함수 정의
def clickLeft(event):
    print('event : ', event )
    mainWin['bg'] = 'yellow'

def leave(event):
    print('event : ', event)
    mainWin['bg'] = 'white'

##- 윈도우 창에 이벤트 연결
mainWin.bind("<Button-1>", clickLeft)
mainWin.bind("<Leave>", leave)

##- 윈도우에서 발생하는 이벤트 메시지 수신
mainWin.mainloop()
```

GUI Tkinter 프로그래밍 2

◆ 이벤트 처리

▪ Entry 이벤트 처리

```
##- 이벤트 처리 함수 정의
def control(event):
    print("입력 이벤트", msg.get())
    msg.delete(first=0, last=100)

##- 1줄 텍스트 필드 객체 생성
msg=Entry( mainWin)

##- # 엔터키 입력 시 처리함수 연결
msg.bind("<Return>", control)

##- 1줄 텍스트 필드  화면 배치
msg.pack(side='right')

##- 1줄 텍스트 필드에 포커스 설정
msg.focus()
```

GUI Tkinter 프로그래밍 2

◆ 이벤트 처리

▪ Button 이벤트 처리

```
##- 버튼 이벤트 처리 함수 정의
def control():
    print("버튼 클릭")

##-  버튼 객체 생성
okBTN=Button( mainWin,
              text='ok',
              command=control)

##- 라벨 화면 배치
okBTN.pack()
```

CONTAINER WIDGET & LAYOUT

CONTAINER WIDGET & LAYOUT

◆ 컨테이너 위젯

▪ Frame

→ 윈도우 창에 **표 형태로 위젯 배치**

→ **셀 단위로 배치**

→ 한번에 여러 셀 건너 뛰어 배치 불가

→ **pack()과 함께 사용 불가**

CONTAINER WIDGET & LAYOUT

◆ 컨테이너 위젯

▪ Pack

→ 윈도우 창에 **상대적인 위치로 Widget 배치**

→ Left - Rigth, Top-Bottom 각 방향으로 상대성 동작

→ **side 객체 속성값으로 설정**

- 속성값 : LEFT, RIGHT, TOP, BOTTOM

CONTAINER WIDGET & LAYOUT

◆ 컨테이너 위젯

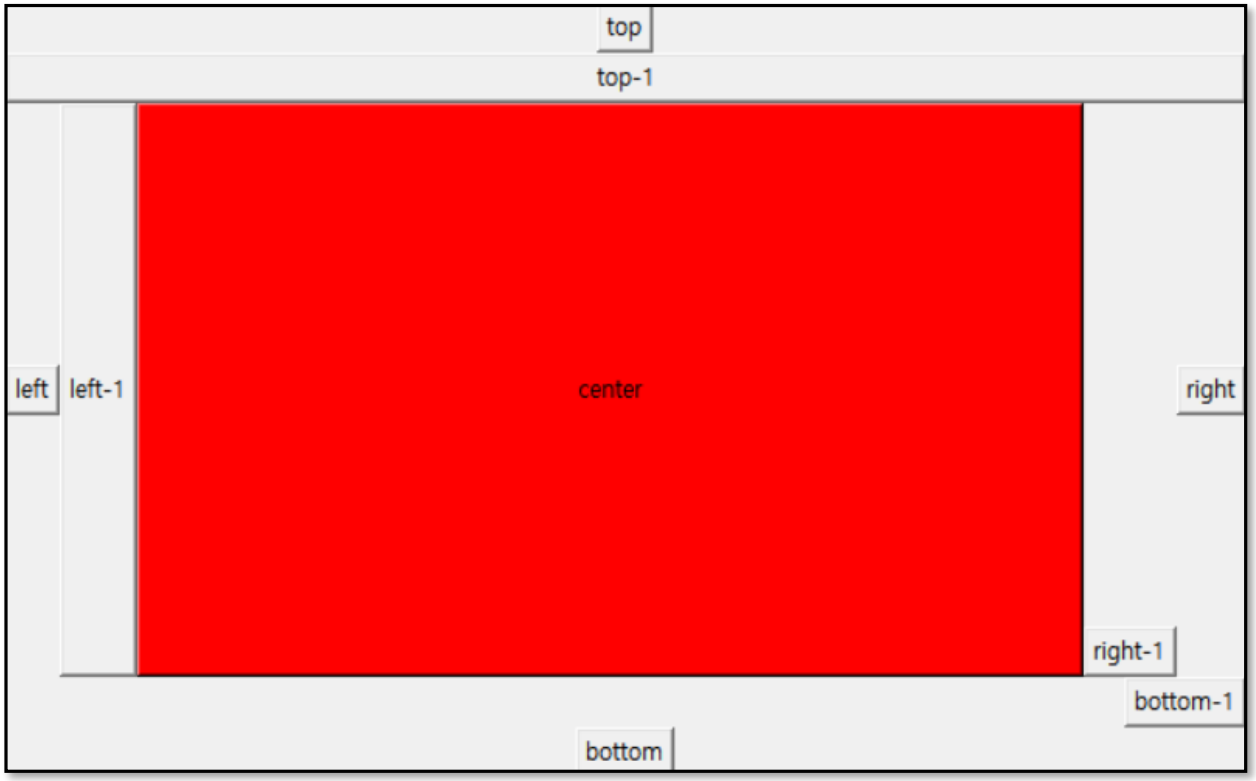
▪ Pack

속성	의미	기본값	속성값
side	• 정렬방향	top	left, right, top, bottom
fill	• 지정방향으로 공간 늘리기 • window resizing 자동 조절하고 싶으면 expand=YES, fill=BOTH	none	x, y, both, none
expand	• 모든 공간 사용 설정 • fill, anchor 함께 사용	True	True, False
anchor	• 위치지정	center	NW, N, NE, E, SE S, SW, W, CENTER

CONTAINER WIDGET & LAYOUT

◆ 컨테이너 위젯

- **Pack**



CONTAINER WIDGET & LAYOUT

◆ 컨테이너 위젯

▪ Grid

→ 윈도우 창을 **표로 나누어 Widget 배치**

→ Left - Rigth, Top-Bottom 각 방향으로 상대성 동작

→ 같은 열에 크기가 다르다면, 가장 큰 길이 기준

→ **절대 위치 배치**

CONTAINER WIDGET & LAYOUT

◆ 컨테이너 위젯

▪ Frame

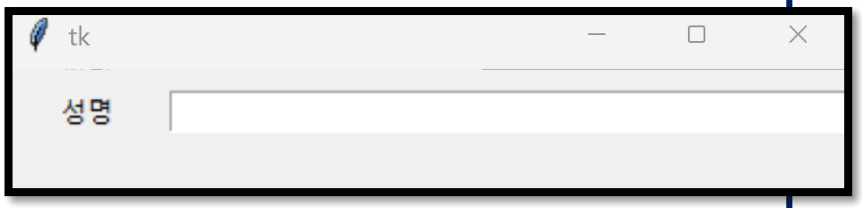
```
##- 윈도우 창 생성
mainWin=Tk()

##- 프레임 객체 생성 및 배치
frame1 = Frame(mainWin)
frame1.pack(fill=X)

##- 프레임에 UI 요소 추가
lblName = Label(frame1, text="성명", width=10)
lblName.pack(side=LEFT)

##- 텍스트 입력 필드 배치
entryName = Entry(frame1)
entryName.pack(fill=X, pady=10)

##- 윈도우에서 발생하는 이벤트 메시지 수신
mainWin.mainloop()
```



CONTAINER WIDGET & LAYOUT

◆ [실습]

▪ 기능

→ 이름, 회사명, 특징 입력 받기

→ 특징은 여러 줄 입력 받기

→ 저장 버튼으로 입력된 내용 파일로 저장하기

CONTAINER WIDGET & LAYOUT

◆ [실습]

- **UI 구성**