

[Opensource Basic Project]

오픈소스 기초 프로젝트

- 7주차 : Window programming

2024. 4. 24(수).

우창우

Dr.woo@chungbuk.ac.kr

□ 주차별 강의계획

※ 4.10(국회의원선거날)

주차(변경 전)	주차(변경 후)	강의계획	과제	번호	제출 마감일
1주(03.06)	1주(03.06)	Python Started	VScode 설치화면, "Hello World" 출력화면 및 작성코드	1	03.06~03.09(토)
2주(03.13)	2주(03.13)	Variables and simple data types	다이아몬드 모양, 진수 변환 프로그램 및 동전 변환 프로그램의 출력화면/작성코드	2	03.13~03.16(토)
3주(03.20)	3주(03.20)	Control Statement	종합계산기, 구구단출력 프로그램의 출력화면/작성코드	3	03.20~03.23(토)
4주(03.27)	4주(03.27)	Lists, Dictionaries, String	음식 궁합 출력, 문자열 거꾸로 출력 프로그램의 출력화면/작성코드	4	03.27~03.30(토)
5주(04.03)	5주(04.03)	Functions and module	GitHub Copilot Extension 설치화면, 로또번호 추첨 출력화면/작성코드	5	04.03~04.06(토)
6주(04.10)	6주(04.17)	Classes	팀프로젝트 과제 제출양식.hwp (조당 1개)	6	04.17~04.20(토)
7주(04.17)	7주(04.24)	Window programming	사진앨범 프로그램의 출력화면/작성코드	7	04.24~04.27(토)
8주(04.24)	8주(05.01)	Midterm (반영비율 20%)	-	-	-

□ 주차별 강의계획

※5.15(부처님오신날)

주차(변경 전)	주차(변경 후)	강의계획	과제	번호	제출 마감일
9주(05.01)	9주(05.08)	Files and Exceptions		8	05.08~05.11(토)
10주(05.08)	10주(05.22)	Database		9	05.22~05.25(토)
11주(05.15)	11주(05.29)	Data visualization		10	05.29~06.01(토)
12주(05.22)	12주(06.05)	Quiz (반영비율 10%), Team project implementation I	-	-	-
13주(05.29)	13주(06.17)	Team project implementation II	-	-	-
14주(06.05)	14주(06.18)	Team project implementation III	-	-	-
15주(06.12)	15주(06.20)	Project Presentation (반영비율 50%)	-	-	-

Chapter 10. 윈도 프로그래밍

Selection 01. 이장에서 만들 프로그램

Selection 02. 기본 위젯 활용

Selection 03. 위젯 배치와 크기 조절

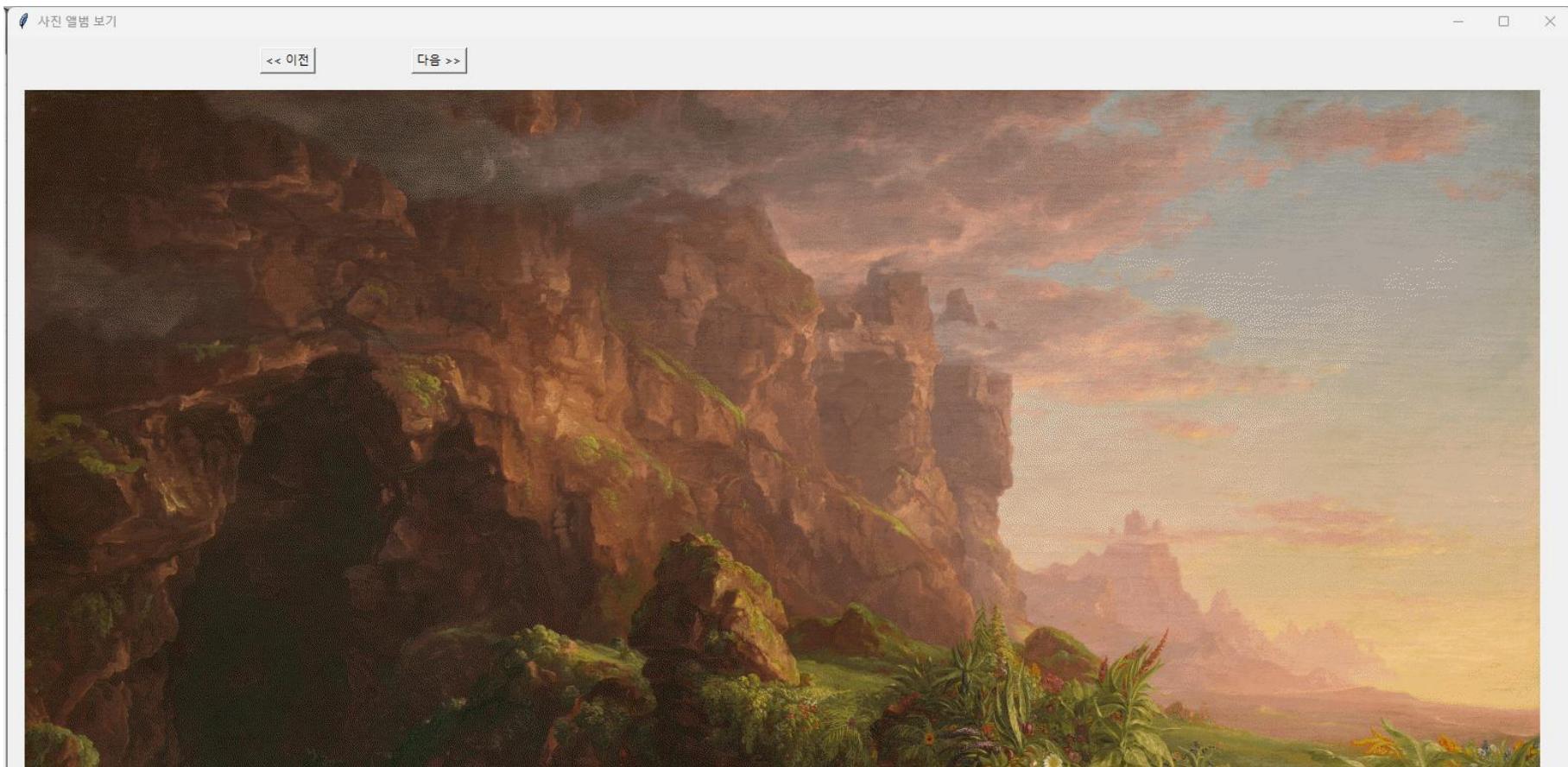
Selection 04. 키보드와 마우스 이벤트 처리

Selection 05. 메뉴와 대화상자

Selection 01. 이 장에서 만들 프로그램

1. 사진 앨범

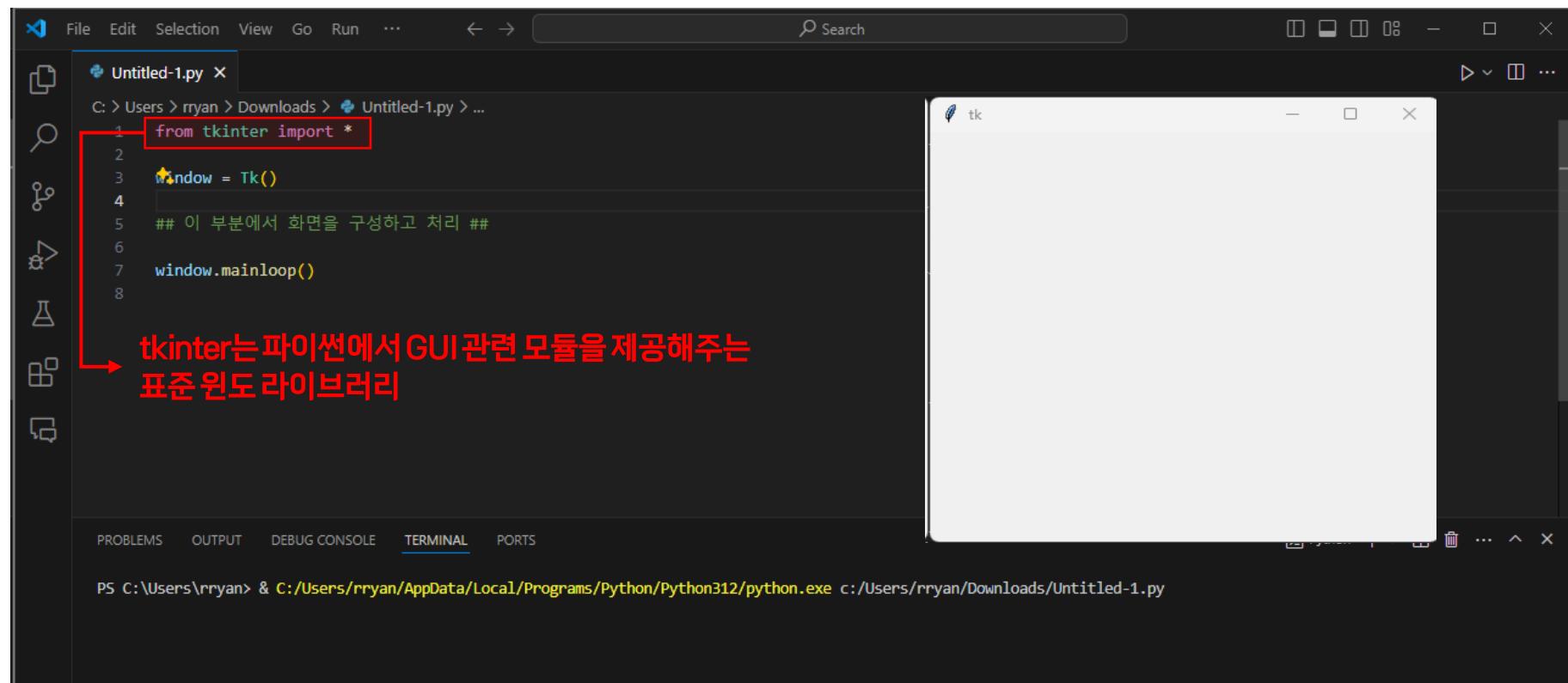
- <이전>과 <다음> 버튼을 누르면 앨범에 있는 사진 목록을 보여주는 프로그램
- 사진파일 4개는 eLMS에 업로드 되어있음 (The Voyage of Life.gif.zip)



Selection 02. 기본 위젯 활용

1. 기본 윈도 창의 구성

- 위젯(Widget)은 윈도 창에 나올 수 있는 문자, 버튼, 체크박스, 라디오 버튼 등을 의미
- (예시) 가장 기본적인 윈도 창을 구현하는 코드



The screenshot shows a code editor interface with a dark theme. On the left is a sidebar with various icons. The main area has a file named "Untitled-1.py" open. The code is as follows:

```
C: > Users > rryan > Downloads > Untitled-1.py > ...
1 from tkinter import *
2
3 window = Tk()
4
5 ## 이 부분에서 화면을 구성하고 처리 ##
6
7 window.mainloop()
8
```

A red arrow points from the text "tkinter는 파이썬에서 GUI 관련 모듈을 제공해주는 표준 윈도 라이브러리" to the first line of code, "from tkinter import *". To the right of the code editor is a small window titled "tk" which is currently empty.

At the bottom of the code editor, there are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is selected, showing the command: PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python312/python.exe c:/Users/rryan/Downloads/Untitled-1.py

Selection 02. 기본 위젯 활용

1. 기본 윈도 창의 구성

- (예시) 가장 기본적인 윈도 창 구현 + 윈도 창 조절 코드

The screenshot shows a Python code editor interface with a dark theme. On the left is a sidebar with various icons. The main area has a file named "Untitled-1.py" open, containing the following code:

```
C: > Users > rryan > Downloads > Untitled-1.py > ...
1  from tkinter import *
2
3  window = Tk()
4  window.title("윈도우 창 연습")
5  window.geometry("600x400")
6  window.resizable(width=False, height=False)
7
8  window.mainloop()
9
```

Below the code editor is a terminal window showing the command PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python312/python.

To the right of the code editor, a small window titled "윈도우 창 연습" is displayed. The window has a title bar and a white interior. In the center of the window, the text "600x400" is printed.

Selection 02. 기본 위젯 활용

2. (위젯) 레이블

- 레이블은 문자를 표현할 수 있는 위젯으로 'Label(부모 윈도, 옵션)'으로 사용
- 레이블과 같은 위젯들은 pack() 함수를 호출해야 화면에 표시

The screenshot shows a Python code editor with a dark theme. On the left is a sidebar with various icons. The main area displays a Python script named 'Untitled-1.py'.

```
C: > Users > rryan > Downloads > Untitled-1.py > ...
1  from tkinter import *
2  window = Tk()
3
4  label1 = Label(window, text = "COOKBOOK~~ Python을")
5  label2 = Label(window, text = "열심히", font = ("궁서체", 30), fg = "blue")
6  label3 = Label(window, text = "공부 중입니다.", bg = "magenta", width = 20, height = 5, anchor = SE)
7
8  label1.pack()
9  label2.pack()
10 label3.pack()
11
12 window.mainloop()
```

Annotations in red highlight specific parts of the code and the resulting window:

- A red bracket labeled "back ground" points to the `bg = "magenta"` line in the code and the magenta background of the window.
- A red bracket labeled "foreground" points to the `fg = "blue"` line in the code and the blue text of the second label.
- A red bracket labeled "anchor = SE" points to the `anchor = SE` line in the code and the bottom-right position of the third label.
- To the right of the window image, a legend lists 9 anchor positions with their corresponding directions:

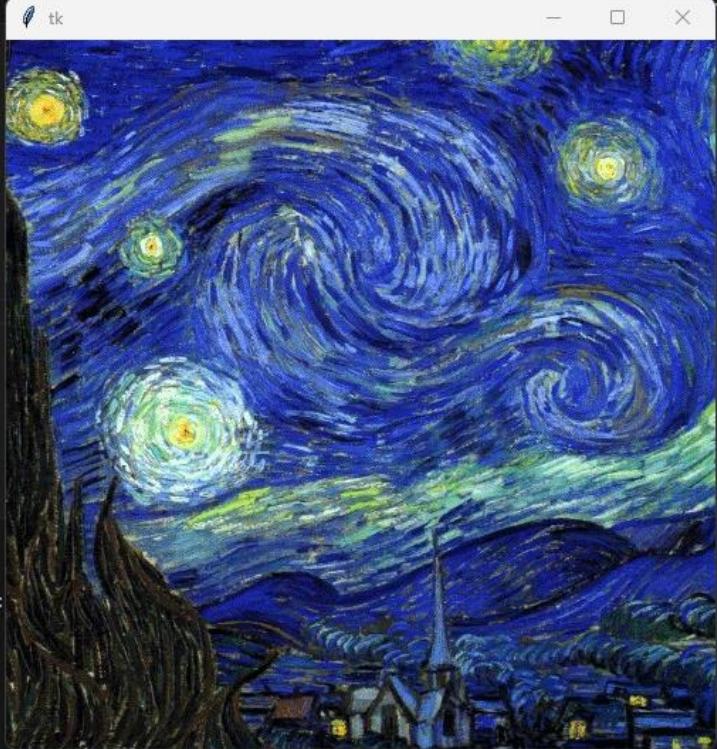
1. NW:북서	6. S:남
2. N:북	7. SW:남서
3. NE:북동	8. W:서
4. E:동	9. CENTER:중앙
5. SE:남동	

At the bottom of the editor, tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS are visible. The TERMINAL tab is selected, showing the command PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python312/python.exe c:/Users/rryan/Downloads/Untitled-1.py.

Selection 02. 기본 위젯 활용

2. (위젯) 레이블

- 글자 대신 이미지도 가능하지만 PhotoImage() 함수는 GIF 파일만 지원



The screenshot shows a Python code editor and a running application window. The code in the editor is:

```
C:\> Users > rryan > Downloads > Untitled-1.py > ...
1  from tkinter import *
2  window = Tk()
3  window.geometry("500x500")
4
5  photo = PhotoImage(file = "image01.gif")
6  label1 = Label(window, image = photo)
7  label1.pack();
8
9  window.mainloop()
10
```

The application window title is "tk". It displays a reproduction of Vincent van Gogh's painting "The Starry Night". The painting features a dark blue night sky filled with swirling stars of varying sizes and colors (yellow, white, green). In the foreground, there is a small town with buildings and a bridge over a river, all rendered in a style consistent with the original painting.

Selection 02. 기본 위젯 활용

3. (위젯) 버튼

- 클릭하면 눌리는 효과와 함께 작업을 실행, 'Button(부모 윈도, 옵션)'으로 사용
- (예시) 버튼을 눌렀을 때 파이썬이 종료되는 코드

The screenshot shows a Python code editor interface with a dark theme. On the left is a sidebar with various icons. The main area displays a Python script named 'Untitled-1.py' with the following code:

```
C: > Users > rryan > Downloads > Untitled-1.py > ...
1  from tkinter import *
2  window = Tk()
3
4  button1 = Button(window, text = "파이썬 종료", fg = "red", command = quit)
5
6  button1.pack()
7
8  window.mainloop()
9
```

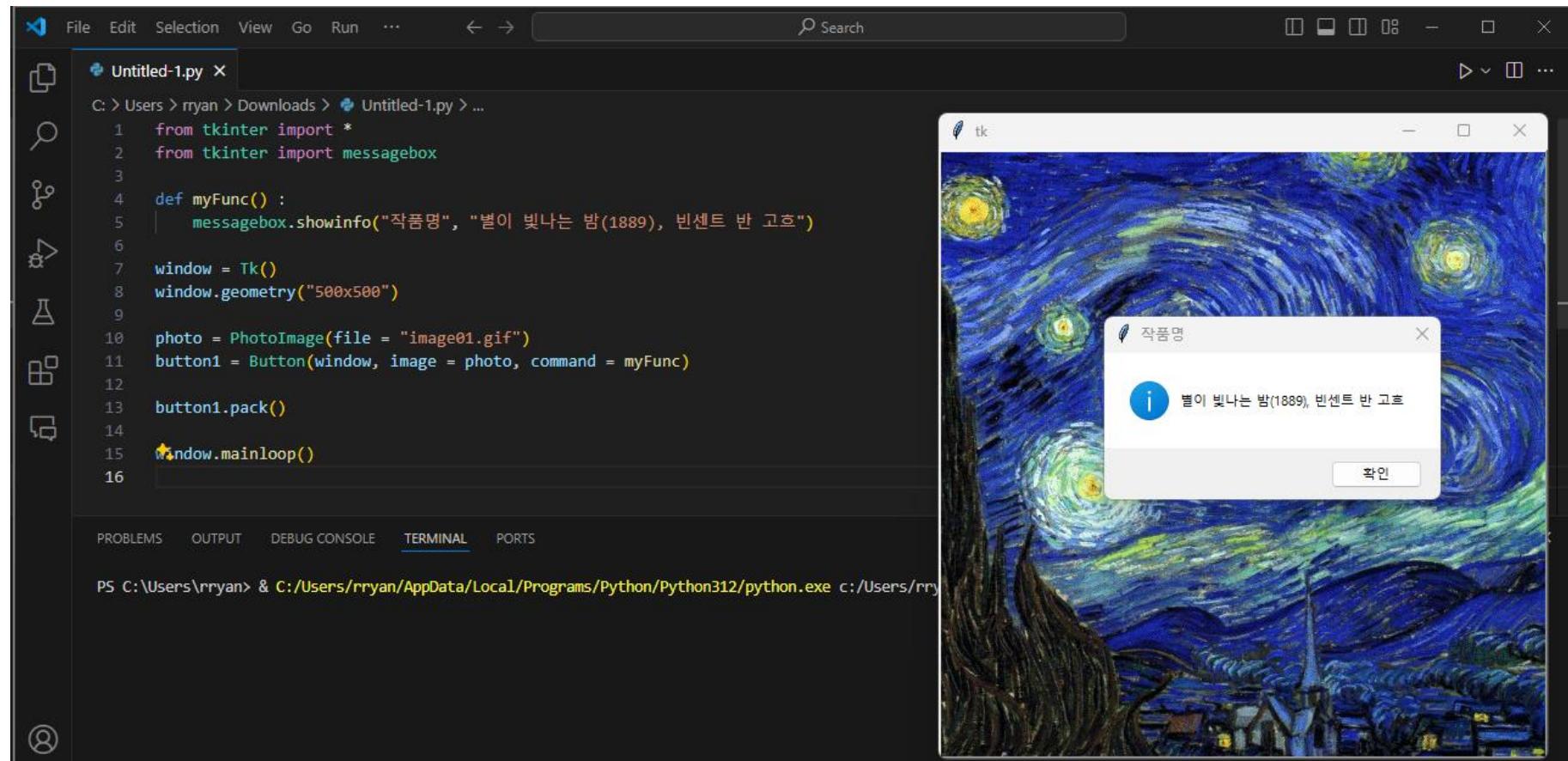
To the right of the code editor, a small window titled 'tk' is displayed. It contains a single button with the text '파이썬 종료' (Python Exit) in red. The entire application is running within a terminal window at the bottom of the screen.

```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python312/python.exe c:/Users/rryan/Downloads/Untitled-1.py
```

Selection 02. 기본 위젯 활용

3. (위젯) 버튼

- (예시) 이미지 버튼을 눌렀을 때 간단한 메시지창이 나오는 코드



The screenshot shows a Python code editor with a script named `Untitled-1.py`. The code uses the `tkinter` module to create a window with a button that displays a message box. The message box contains the text "작품명" and "별이 빛나는 밤(1889), 빈센트 반 고흐". The application window is overlaid on a background image of Vincent van Gogh's "Starry Night".

```
C: > Users > rryan > Downloads > Untitled-1.py > ...
1  from tkinter import *
2  from tkinter import messagebox
3
4  def myFunc() :
5      messagebox.showinfo("작품명", "별이 빛나는 밤(1889), 빈센트 반 고흐")
6
7  window = Tk()
8  window.geometry("500x500")
9
10 photo = PhotoImage(file = "image01.gif")
11 button1 = Button(window, image = photo, command = myFunc)
12
13 button1.pack()
14
15 window.mainloop()
16
```

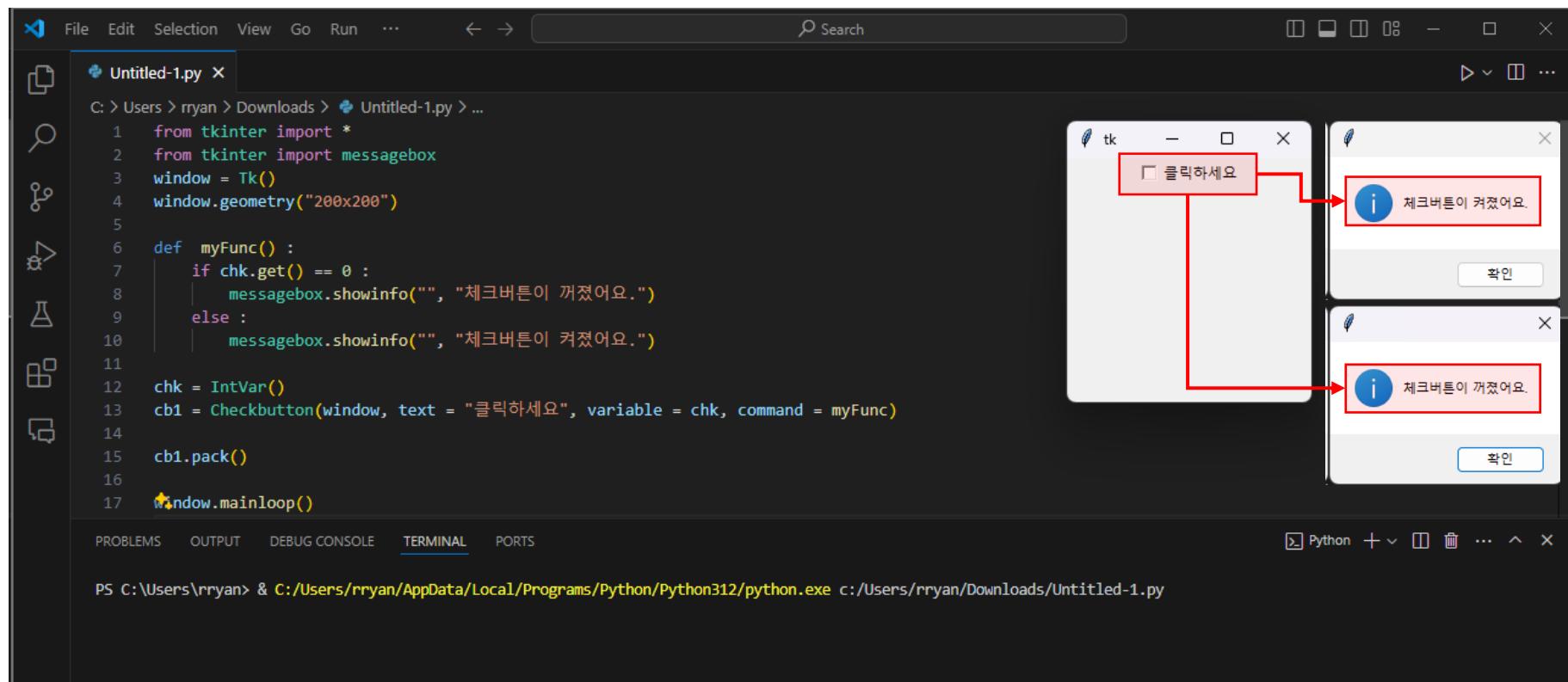
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python312/python.exe c:/Users/rry

Selection 02. 기본 위젯 활용

4. (위젯) 체크버튼

- ▣ 켜고 끄는데 사용하는 위젯으로, 'Checkbutton(부모 윈도, 옵션)'으로 사용
- ▣ (예시) 체크버튼을 켜거나 끌면 메시지창을 띄워주는 코드



```
C:\> Users > rryan > Downloads > Untitled-1.py > ...
1  from tkinter import *
2  from tkinter import messagebox
3  window = Tk()
4  window.geometry("200x200")
5
6  def myFunc() :
7      if chk.get() == 0 :
8          messagebox.showinfo("", "체크버튼이 꺼졌어요.")
9      else :
10         messagebox.showinfo("", "체크버튼이 켜졌어요.")
11
12 chk = IntVar()
13 cb1 = Checkbutton(window, text = "클릭하세요", variable = chk, command = myFunc)
14
15 cb1.pack()
16
17 window.mainloop()
```

PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python312/python.exe c:/Users/rryan/Downloads/Untitled-1.py

Selection 02. 기본 위젯 활용

5. (위젯) 라디오버튼

- 여러 개 중 하나를 선택할 때 사용하는 위젯, 'Radiobutton(부모 윈도, 옵션)'으로 사용
- (예시) 라디오 버튼을 선택하면 레이블을 변경해주는 코드

```
File Edit Selection View Go Run ... ← → Search ...  
Untitled-1.py X Untitled-1.py X  
C: > Users > rryan > Downloads > Untitled-1.py > ...  
1 from tkinter import *  
2 window = Tk()  
3 window.geometry("200x100")  
4  
5 def myFunc() :  
6     if var.get() == 1 :  
7         label1.configure(text = "파이썬")  
8     elif var.get() == 2 :  
9         label1.configure(text = "C++")  
10    else :  
11        label1.configure(text = "Java")  
12  
13 var = IntVar()  
14 rb1 = Radiobutton(window, text = "파이썬", variable = var, value = 1, command = myFunc)  
15 rb2 = Radiobutton(window, text = "C++", variable = var, value = 2, command = myFunc)  
16 rb3 = Radiobutton(window, text = "Java", variable = var, value = 3, command = myFunc)  
17  
18 label1 = Label(window, text = "선택한 언어 :")  
19 label1.pack()  
20 rb1.pack()  
21 rb2.pack()  
22 rb3.pack()  
23 label1.pack()  
24  
25 window.mainloop()  
26  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python312/python.exe c:/Users/rryan/Downloads/Untitled-1.py
```

The screenshot shows a code editor with two tabs: 'Untitled-1.py' and 'Untitled-1.py'. The code defines a function 'myFunc' that changes the text of a label 'label1' based on the selected radio button ('rb1', 'rb2', or 'rb3'). The 'rb1' button is initially selected. Three separate windows are shown on the right, each containing a radio button interface with three options: '파이썬', 'C++', and 'Java'. In the first window, '파이썬' is selected. In the second window, 'C++' is selected. In the third window, 'Java' is selected. Red arrows point from the code editor's radio button line to each of these three windows, illustrating how the code controls the state of the radio buttons.

Selection 03. 위젯의 배치와 크기 조절

1. 정렬

- pack() 함수의 옵션 중 side=를 사용하여 정렬 (top, bottom, left, right)

The screenshot shows a VS Code interface with a Python script named Untitled-1.py. The code creates a window and four buttons, each labeled with its position ('top', 'bottom', 'left', 'right') and packed using the specified side option. A red box highlights the pack() lines for buttons 1 through 4. To the right, a screenshot of the resulting Tkinter window shows four buttons arranged along the top, bottom, left, and right edges respectively.

```
C:\> Users > rryan > Downloads > Untitled-1.py > ...
1  from tkinter import *
2  window = Tk()
3  window.geometry("500x400")
4
5  button1 = Button(window, text = "top")
6  button2 = Button(window, text = "bottom")
7  button3 = Button(window, text = "left")
8  button4 = Button(window, text = "right")
9
10 button1.pack(side = TOP)
11 button2.pack(side = BOTTOM)
12 button3.pack(side = LEFT)
13 button4.pack(side = RIGHT)
14
15 window.mainloop()
16
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python312/python.exe

Selection 03. 위젯의 배치와 크기 조절

1. 정렬

- (예시) 리스트와 for 문을 활용한 버튼 생성 코드

The screenshot shows a Visual Studio Code (VS Code) interface with a Python file named "Untitled-1.py" open. The code uses a for loop to create three buttons and pack them onto the right side of a window. The resulting Tkinter window displays three buttons labeled "버튼3", "버튼2", and "버튼1" from left to right.

```
C:\> Users > rryan > Downloads > Untitled-1.py > ...
1  from tkinter import *
2  window = Tk()
3  window.geometry("500x400")
4
5  btnList = [""] * 3
6
7  for i in range(0, 3) :
8      btnList[i] = Button(window, text = "버튼" + str(i + 1))
9
10 for btn in btnList :
11     btn.pack(side = RIGHT)
12
13 window.mainloop()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python312/python.exe c:/Users/
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python312/python.exe c:/Users/
PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python312/python.exe c:/Users/
```

Selection 03. 위젯의 배치와 크기 조절

2. 폭 및 여백 조정

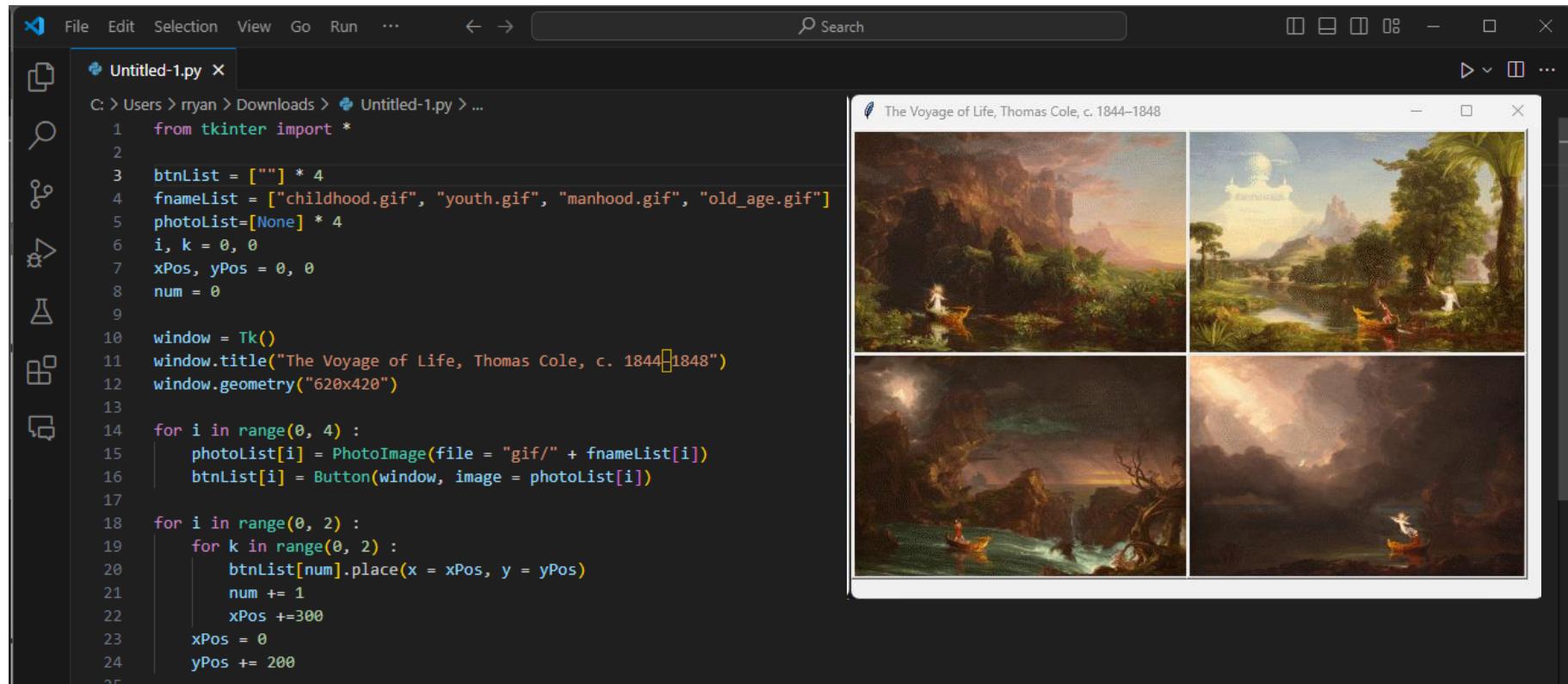
- (폭 조정) pack() 함수의 옵션 중 fill=X를 사용하여 윈도 창에 폭을 맞춤
- (여백) padx, pady를 통해 위젯 사이를, ipadx, ipady를 통해 위젯 내부 여백을 지정

```
File Edit Selection View Go Run ... ← → Search tk tk tk tk  
Untitled-1.py X C: > Users > rryan > Downloads > Untitled-1.py > ...  
1  from tkinter import *  
2  window = Tk()  
3  window.geometry("300x200")  
4  
5  btnList = [""] * 3  
6  
7  for i in range(0, 3) :  
8      btnList[i] = Button(window, text = "버튼" + str(i + 1))  
9  
10 for btn in btnList :  
11     btn.pack(side = TOP, fill = X)  
12     btn.pack(side = TOP, fill = X, padx = 10, pady = 10)  
13     btn.pack(side = TOP, fill = X, ipadx = 10, ipady = 10)  
14     btn.pack(side = TOP, fill = X, ipadx = 10, ipady = 10, padx = 10, pady = 10)  
15  
16 window.mainloop()  
17
```

Selection 03. 위젯의 배치와 크기 조절

3. 고정 위치에 배치

- 위젯을 고정 위치에 배치하려면 pack() 대신 place() 함수를 사용
- (예시) 그림 4개를 2차원으로 배치하는 코드



The screenshot shows a code editor window with a Python script named 'Untitled-1.py'. The script uses the Tkinter library to create a window titled 'The Voyage of Life, Thomas Cole, c. 1844-1848' with a geometry of '620x420'. It defines four buttons ('btnList') and four image files ('fnameList'). A list ('photoList') is initialized with four None values. The script then creates a window and places four images ('photoList[i]') at specific coordinates using the 'place' method. The images are arranged in a 2x2 grid. The top-left image depicts a scene with a small boat on a river, while the other three images show various stages of life or scenes from the painting.

```
C: > Users > rryan > Downloads > Untitled-1.py > ...
1  from tkinter import *
2
3  btnList = [""] * 4
4  fnameList = ["childhood.gif", "youth.gif", "manhood.gif", "old_age.gif"]
5  photoList=[None] * 4
6  i, k = 0, 0
7  xPos, yPos = 0, 0
8  num = 0
9
10 window = Tk()
11 window.title("The Voyage of Life, Thomas Cole, c. 1844-1848")
12 window.geometry("620x420")
13
14 for i in range(0, 4) :
15     photoList[i] = PhotoImage(file = "gif/" + fnameList[i])
16     btnList[i] = Button(window, image = photoList[i])
17
18 for i in range(0, 2) :
19     for k in range(0, 2) :
20         btnList[num].place(x = xPos, y = yPos)
21         num += 1
22         xPos +=300
23     xPos = 0
24     yPos += 200
25
```

Selection 04. 키보드와 마우스 이벤트 처리

1. 마우스 이벤트 기본 처리

- 키보드나 마우스를 누르는 것을 이벤트(event)라고 하며, 클릭할 때, 드래그할 때, 버튼을 눌렀다 뗄 때 등 다양한 상황의 이벤트 처리가 가능
- 마우스 이벤트 처리 형식

```
def 이벤트처리함수(event) :  
    # 이 부분에 마우스 이벤트가 발생할 때 작동할 내용 작성  
  
    위젯.bind("마우스이벤트", 이벤트처리함수)
```

- 마우스 이벤트가 발생할 때 처리하기 위한 함수를 만들어 놓아야 하며, 이때 함수는 event를 매개변수로 전달받음
- 이벤트를 처리할 위젯에 bind() 함수를 사용할 마우스 이벤트와, 이벤트가 발생할 때 작동할 함수명을 입력함으로써 이벤트를 마우스 이벤트를 처리

Selection 04. 키보드와 마우스 이벤트 처리

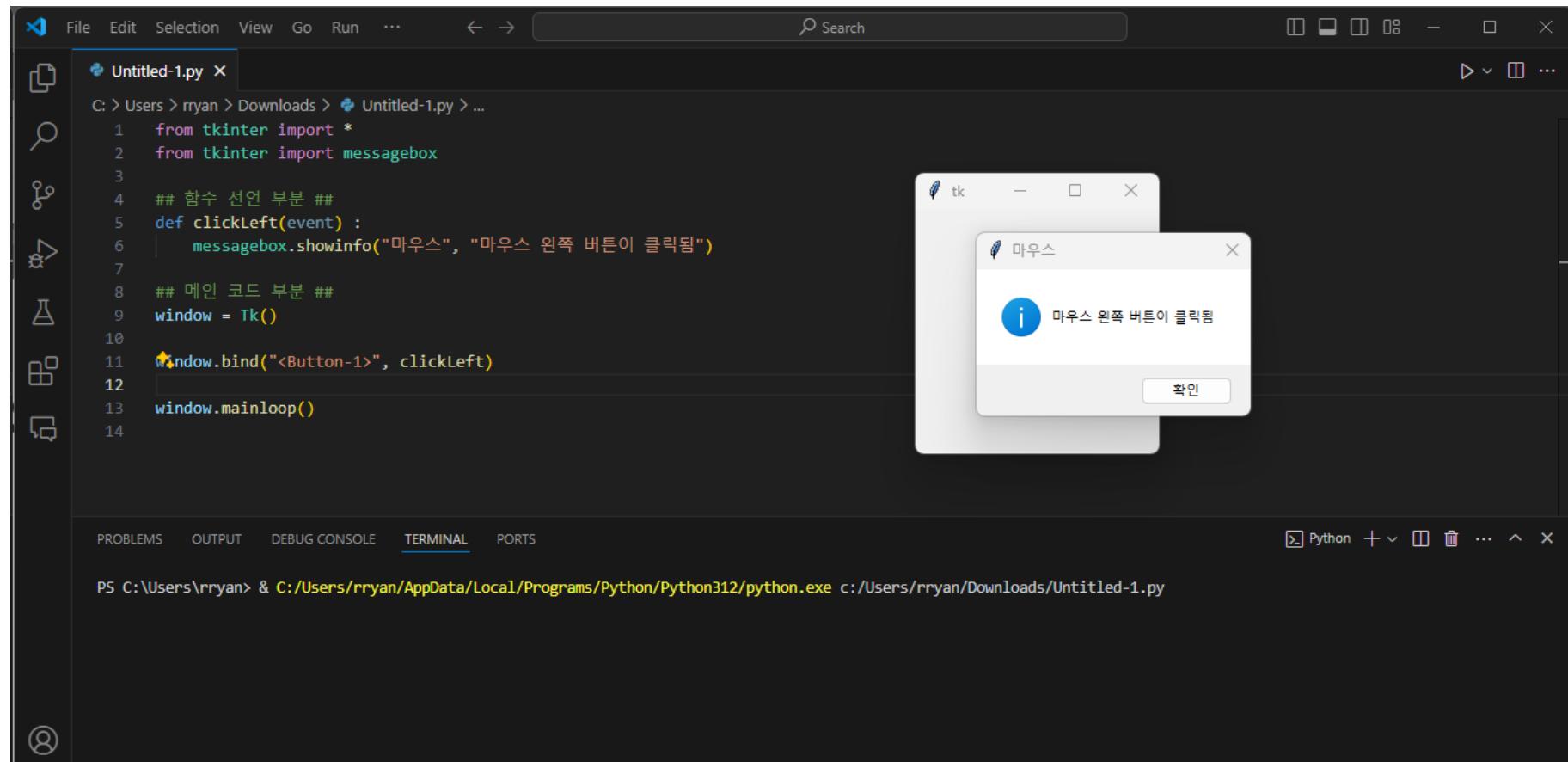
1. 마우스 이벤트 기본 처리

마우스작동	마우스 버튼	이벤트 코드
클릭할 때	마우스 버튼 마우스 왼쪽 버튼 마우스 가운데 버튼 마우스 오른쪽 버튼	〈Button〉 〈Button-1〉 〈Button-2〉 〈Button-3〉
떼었을 때	버튼을 놓았을 때 발생 왼쪽 버튼을 놓았을 때 발생 가운데 버튼을 놓았을 때 발생 오른쪽 버튼을 놓았을 때 발생	〈ButtonRelease〉 〈ButtonRelease-1〉 〈ButtonRelease-2〉 〈ButtonRelease-3〉
더블 클릭할 때	더블 클릭하면 발생	〈DoubleClick〉
드래그 할 때	마우스가 움직이면 발생 왼쪽 버튼을 누르면서 움직이면 발생 가운데 버튼을 누르면서 움직이면 발생 오른쪽 버튼을 누르면서 움직이면 발생	〈Motion〉 〈B1-Motion〉 〈B2-Motion〉 〈B3-Motion〉
마우스 커서가 위젯 위로 올라왔을 때		〈Enter〉
마우스 커서가 위젯을 떠날 때		〈Leave〉

Selection 04. 키보드와 마우스 이벤트 처리

1. 마우스 이벤트 기본 처리

- (예시) 마우스 왼쪽 버튼을 클릭했을 때 이벤트를 처리하는 코드



The screenshot shows a Python code editor with a dark theme. On the left, there's a sidebar with various icons. The main area has a file named "Untitled-1.py" open. The code is as follows:

```
C: > Users > rryan > Downloads > Untitled-1.py > ...
1  from tkinter import *
2  from tkinter import messagebox
3
4  ## 함수 선언 부분 ##
5  def clickLeft(event):
6      messagebox.showinfo("마우스", "마우스 왼쪽 버튼이 클릭됨")
7
8  ## 메인 코드 부분 ##
9  window = Tk()
10
11 window.bind("<Button-1>", clickLeft)
12
13 window.mainloop()
```

To the right of the code editor, a small window titled "마우스" (Mouse) is displayed. It contains the message "마우스 왼쪽 버튼이 클릭됨" (Left mouse button clicked) and a "확인" (Confirm) button. At the bottom of the screen, there's a terminal window showing the command "PS C:\Users\rryan> & C:/Users/rryan/AppData/Local/Programs/Python/Python312/python.exe c:/Users/rryan/Downloads/Untitled-1.py".

Selection 04. 키보드와 마우스 이벤트 처리

1. 마우스 이벤트 기본 처리

- (예시) 지정된 위치를 클릭했을 때 다른 함수(messagebox)를 호출하는 코드

The screenshot shows a Python IDE interface with a code editor and a running application window.

Code Editor:

```
C: > Users > rryan > Downloads > Untitled-1.py > ...
1  from tkinter import *
2  from tkinter import messagebox
3
4  ## 함수 선언 부분 ##
5  def clickImage(event) :
6      messagebox.showinfo("마우스", "The Voyage of Life: Youth")
7
8  ## 메인 코드 부분 ##
9  window = Tk()
10 window.geometry("820x560")
11
12 photo = PhotoImage(file = "gif/youth-500.gif")
13 label1 = Label(window, image = photo)
14
15 label1.bind("<Button>", clickImage)
16
17 label1.pack( expand = 1, anchor = CENTER)
18 window.mainloop()
19
```

Running Application:

A Tkinter window titled "마우스" is displayed over a landscape painting. The window contains an information message: "The Voyage of Life: Youth". It includes an icon, a message text area, and a "확인" (Confirm) button.

Selection 04. 키보드와 마우스 이벤트 처리

2. event 매개변수를 활용한 마우스 이벤트 처리

- event 매개변수는 마우스에 관련된 다양한 정보(좌표, 번호 등)가 포함되어 있음
- (예시) 마우스를 클릭할 때마다 어떤 버튼이 어디서 클릭했는지 좌표를 출력하는 코드

The screenshot shows a code editor with a dark theme. On the left is the code for 'Untitled-1.py':

```
C: > Users > rryan > Downloads > Untitled-1.py > ...
1  from tkinter import *
2
3  def clickMouse(event) :
4      txt = ""
5      if event.num == 1 :
6          txt += "마우스 왼쪽 버튼이 (" + str(event.y) + "," + str(event.x) + ")에서 클릭됨"
7      elif event.num == 3 :
8          txt += "마우스 오른쪽 버튼이 (" + str(event.y) + "," + str(event.x) + ")에서 클릭됨"
9
10     label1.configure(text = txt)
11
12
13
14 window = Tk()
15 window.geometry("400x400")
16
17 label1 = Label(window, text = "이곳이 바뀜")
18
19 window.bind("<Button>", clickMouse)
20
21 label1.pack(expand = 1, anchor = CENTER)
22 window.mainloop()
23
```

On the right, there are two windows titled 'tk' showing the output of the code. The first window shows the message '마우스 왼쪽 버튼이 (103,101)에서 클릭됨'. The second window shows the message '마우스 오른쪽 버튼이 (93,278)에서 클릭됨'.

Selection 04. 키보드와 마우스 이벤트 처리

3. 키보드 이벤트 기본 처리

- 키보드 이벤트는 위젯에서 키보드가 눌리면 발생하는 이벤트
- (예시) 위젯에서 어떤 키를 눌렀는지 확인하는 이벤트 처리 코드

The screenshot shows a Python code editor with a dark theme. On the left, there is a sidebar with various icons. The main area contains a code editor with the following Python script:

```
C:\> Users > rryan > Downloads > Untitled-1.py > ...
1  from tkinter import *
2  from tkinter import messagebox
3
4  ## 함수 선언 부분 ##
5  def keyEvent(event) :
6      messagebox.showinfo("키보드 이벤트", "눌린 키 : " + chr(event.keycode))
7
8  ## 메인 코드 부분 ##
9  window = Tk()
10
11 window.bind("<Key>", keyEvent)
12
13 window.mainloop()
```

To the right of the code editor, four separate message boxes are displayed, each showing the result of a different key press:

- Top-left message box: 놀린 키 : G
입력 키 : G
- Top-right message box: 놀린 키 : 5
입력 키 : 5
- Bottom-left message box: 놀린 키 : A
입력 키 : `
- Bottom-right message box: 놀린 키 : ¶
입력 키 : Capslock

Selection 04. 키보드와 마우스 이벤트 처리

3. 키보드 이벤트 기본 처리

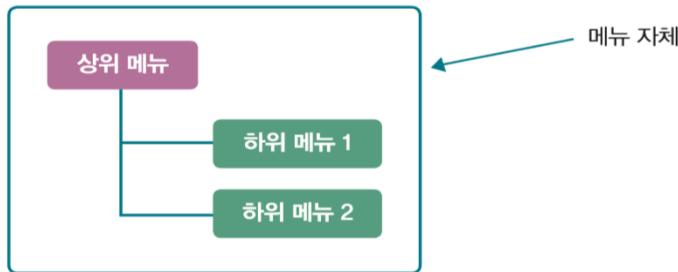
키보드 작동	이벤트 코드
모든 키를 누를 때	<code><Key></code>
특수 키를 누를 때	<code><Return>, <BackSpace>, <Tab>, <Shift_L>, <Control_L>, <Alt_L>, <Pause>, <Caps_Lock>, <Escape>, <End>, <Home>, <Left>, <Right>, <Up>, <Down>, <Num_Lock>, <Delete>, <F1>~<F12> 등</code>
일반 키를 누를 때	<code>a~z, A~Z, 0~9, <space>, <less></code>
화살표 키와 조합	<code><Shift-Up>, <Shift-Down>, <Shift-Left>, <Shift-Right> 등</code>

- ‘Enter’을 처리하기 위해서는 `<Return>`을 사용하고, ‘SpaceBar’는 `<Space>`로

Selection 05. 메뉴와 대화상자

1. 메뉴의 생성

- **메뉴를 생성하면 하위 메뉴가 나타나고, 하위 메뉴 중 하나를 선택하면 해당 작업이 실행**
(예시) 파일-열기 메뉴를 선택하면 파일을 선택하는 대화창이 열림
- **메뉴의 구성 개념과 형식**



메뉴자체 = Menu(부모윈도)
부모윈도.config(menu = 메뉴자체)

상위메뉴 = Menu(메뉴자체)
메뉴자체.add_cascade(label = "상위메뉴텍스트", menu = 상위메뉴)
상위메뉴.add_command(label = "하위메뉴1", command = 함수1)
상위메뉴.add_command(label = "하위메뉴2", command = 함수2)

Selection 05. 메뉴와 대화상자

1. 메뉴의 생성

- (예시) [파일] 메뉴 아래에 [열기] 와 [종료] 하위 메뉴가 있는 코드

The screenshot shows a Python code editor with a dark theme. On the left is a sidebar with various icons. The main area displays a Python script named 'Untitled-1.py' with the following code:

```
C: > Users > rryan > Downloads > Untitled-1.py > ...
1  from tkinter import *
2
3  window = Tk()
4
5  mainWindow = Menu(window)
6  window.config(menu = mainWindow)
7
8  fileMenu = Menu(mainWindow)
9  mainWindow.add_cascade(label = "파일", menu = fileMenu)
10 fileMenu.add_command(label = "열기")
11 fileMenu.add_separator()
12 fileMenu.add_command(label = "종료")
13
14 window.mainloop()
15
```

To the right of the code editor is a small window titled 'tk'. The window has a menu bar with the word '파일' (File). A red box highlights the '파일' menu, which contains two items: '열기' (Open) and '종료' (Exit).

Selection 05. 메뉴와 대화상자

1. 메뉴의 생성

- (예시) [파일] 메뉴 아래에 [열기] 와 [종료] 하위 메뉴가 있는 코드 + 실행함수 구현

The screenshot shows a Python code editor with a dark theme. On the left, there's a sidebar with various icons. The main area contains the following Python code:

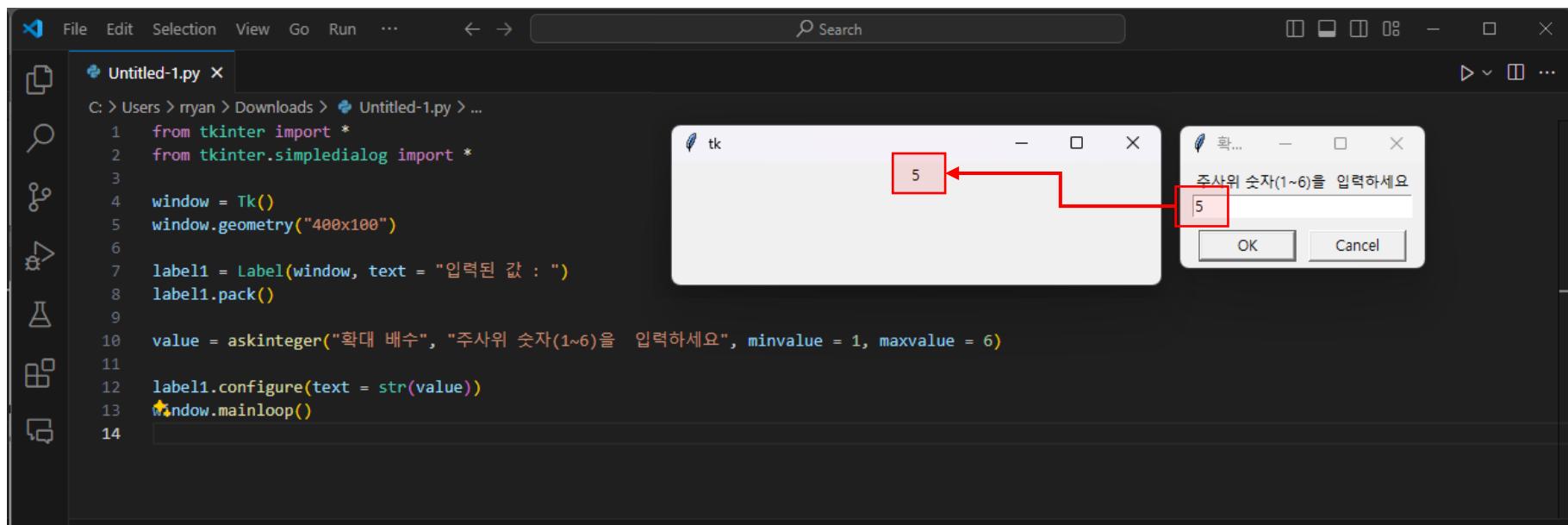
```
C: > Users > rryan > Downloads > Untitled-1.py > ...
1  from tkinter import *
2  from tkinter import messagebox
3
4  def func_open() :
5      messagebox.showinfo("메뉴 선택", "열기 메뉴를 선택함")
6
7  def func_exit() :
8      window.quit()
9      window.destroy()
10
11 window = Tk()
12
13 mainWindow = Menu(window)
14 window.config(menu=mainMenu)
15
16 fileMenu = Menu(mainMenu)
17 mainWindow.add_cascade(label = "파일", menu = fileMenu)
18 fileMenu.add_command(label = "열기", command = func_open)
19
20 fileMenu.add_separator()
21 fileMenu.add_command(label = "종료", command = func_exit)
22
23 window.mainloop()
24
```

To the right of the code, two windows are displayed. The first window is a standard Tkinter application window titled 'tk' with a menu bar. A red box highlights the '파일' (File) menu item, which has a cascading menu with '열기' (Open) and '종료' (Exit) options. An arrow points from this menu to a second window, a message box titled '메뉴 선택' (Menu Selection). The message box contains an information icon and the text '열기 메뉴를 선택함' (The Open menu was selected), with a '확인' (Confirm) button at the bottom.

Selection 05. 메뉴와 대화상자

2. 대화상자의 생성과 사용

- 기본적인 메시지를 표시하는 `messagebox.showinfo()` 외에도 `tkinter.simpledialog` 모듈을 임포트 한 후 `askinteger()` 및 `askstring()` 등을 사용하여 처리가 가능
- 정수를 입력 받는 `askinteger`함수는 파라미터를 (“제목”, “내용”, 옵션) 형태로 사용하며, 실수를 입력 받으려면 `askfloat()` 함수를, 문자열을 입력 받으려면 `askstring` 등을 사용



Chapter 10. 요약

- ✓ 위젯(Widget)은 윈도창에서 나타내는 문자, 버튼, 체크박스, 라디오버튼 등을 의미
- ✓ Tkinter는 파이썬에서 GUI 관련 모듈을 제공하는 표준 윈도 라이브러리
- ✓ 레이블은 문자를 표현할 수 있는 위젯으로, 'Label(부모윈도, 옵션)' 형식으로 사용
- ✓ 버튼은 마우스로 클릭하면 눌리는 효과와 함께 지정한 작업이 실행되도록 하는 위젯으로, 'Button(부모윈도, 옵션)' 형식으로 사용하며 command 옵션을 눌렀을 때 특정 작업을 처리
- ✓ 체크버튼은 켜고 끄는데 사용하는 위젯으로, 'Checkbutton(부모윈도, 옵션)' 형식으로 사용
- ✓ 라디오버튼은 여러 개 중에서 하나를 선택하는 위젯으로, 'Radiobutton(부모윈도, 옵션)' 형식으로 사용
- ✓ pack() 함수의 옵션 중에서 정렬하는 방법은 side=LEFT, RIGHT, TOP, BOTTOM
- ✓ 또한, 위젯 사이에 여백을 주는 방법은 padx=픽셀값, pady=픽셀값으로 하고, 위젯 내부에 여백을 주는 방법은 ipadx=픽셀값, ipady=픽셀값으로 구현

Chapter 10. 요약

- ✓ 위젯을 고정 위치에 배치하려면 pack() 함수가 아닌 place() 함수를 사용하며
'place(x=X좌표, y=Y좌표, width=폭, height=높이)' 형식으로 사용
- ✓ 키보드 및 마우스를 누르는 것을 이벤트(Event)라고 하며, 윈도 프로그램의 mainloop()는 이런 이벤트가 발생하는 것을 기다리는 함수
- ✓ <Button> 이벤트는 모든 마우스 버튼을 클릭할 때 공통으로 발생하며, 처리하고자 하는 값에 따라 <Button-1> (마우스 왼쪽 클릭), <Button-3> (마우스 가운데 클릭) 등을 사용
- ✓ event 매개변수에는 클릭한 위치의 좌표 (x, y), 마우스 버튼의 번호 (num) 등 마우스와 관련된 다양한 정보를 포함
- ✓ 위젯에서 키보드가 눌리면 발생하는 대표적인 키보드 관련 이벤트는 <key>로, 키보드가 눌리면 <key> 이벤트가 발생하고 어떤 키를 눌렀는지 알려면 keycode 값을 확인
- ✓ 숫자나 문자를 입력 받으려면 tkinter.simpledialog 모듈을 임포트한 후 askinteger(), askfloat(), askstring() 함수 등을 사용

감사합니다.

우창우

Dr.woo@chungbuk.ac.kr

팀프로젝트 현황

조 번호	프로젝트 주제	조장	조원
1조	물리 엔진과 충돌 처리 기능, 그리고 캐릭터 디자인을 반영한 슈팅 게임 (슈퍼 마리오)	조형준	고태경, 김다민
2조	다중 미니게임 클라이언트를 위한 통합 아키텍처 개발	김민혁	전영우, 김정민
3조	팀프로젝트 수행능력 및 파이썬 프로그래밍 역량 강화를 위한 방탈출 게임 개발	홍성진	김태영, 정새연
4조	재학생의 완벽한 한 끼 보장을 위한 맛집 서비스 플랫폼 개발	이규민	우태현, 전수혁
5조	날씨에 따른 일정 관리를 위한 캘린더 앱	배정민	박상인, 서범교, 송설희
6조	교내 학생 간의 물품 중고거래를 위한 시스템 SW를 통한 웹사이트 제작	박조현	김건우, 오다영
7조	최근 인기 있는 음악의 키워드를 활용한 시기별 음악을 찾아주는 프로그램 개발	박주현	권정욱, 정현준
8조	보드게임 (like 클루)	김규현	김준후, 조윤정
9조	학생들을 위한 학습도우미 앱	신종환	신승우, 한강민
11조	학교 졸업사정 Q&A 챗봇	한준영	고태영, 이관학, 육광민
12조	학생의 수업의 참여율 감소 문제를 해결하기 위한 수업 참여도 인공지능 기술개발	윤시훈	전준석, 김민경
13조	AI와 데이터베이스를 활용해 복잡성을 최소화한 화장품 추천 웹사이트	김준호	황지연, 이용희
14조	게임개발 이해도 향상을 위한 고도엔진을 이용한 퍼즐게임 개발	배수환	이한결, 신혜원
15조	학생들의 경제적 부담 완화를 위한 택시 동승 서비스 개발	박성범	이태정, 김민석

중간고사 (Midterm) : 반영비율 20%

- 일시 : 2024.05.01 (수), 09:30 ~ 11:30 (2시간)
- 장소 : S4-1동(전자정보3관) 102호 (※ 201호 아님)
- 내용 : **객관식(1~7주차 학습내용) 및 주관식(실행결과, 코드수정보완[블럭])**

〈객관식 예시〉

다음 보기지를 실행하면 다른 값이 출력하는 것을 하나 고르세요.

- ① print("200")
- ② print(200)
- ③ print(100+100)
- ④ print("100+100")

〈주관식 예시〉

다음과 같은 결과가 나오도록 코드의 빈 곳을 채우세요.

실행 결과:

10진수 ==> 12345678

2진수 ==> 0b101111000110000101001110

코드:

```
num = 12345678
```

```
print("10진수 ==> ", [ 1 ])
```

```
print("2진수 ==> ", [ 2 ])
```

실습&과제 : (제출처) Dr.woo@chungbuk.ac.kr, (기한) 4.27(토) 까지

1. 사진 앨범

- <이전>과 <다음> 버튼을 누르면 앨범에 있는 사진 목록을 보여주는 프로그램

