

공학적 문제해결 기법

14주차 1차시

캔버스에 그림 그리기





14주차 1차시 : 캔버스에 그림 그리기

I. 개요

캔버스에 좌표를 이용한 도형 그리기

II. 학습 개요

1) 학습 목표

더 나은 그래픽을 위해 tkinter 모듈을 사용하여 캔버스를 생성하고, 클릭할 수 있는 버튼을 만들어 동작시켜 본다. 캔버스에 그림을 그리기 위해 좌표를 이용하여 정사각형은 물론 여러 가지 직사각형을 그려봄으로써 기존의 turtle과 차이점을 안다.

2) 학습 목차(세부 목차)

- tkinter 사용하여 클릭할 수 있는 버튼 만들기
- 그림을 그리기 위한 캔버스 생성
- 여러 가지 도형 그리기



14주차 1차시 : 캔버스에 그림 그리기

III. 학습 1 - tkinter 사용하여 클릭할 수 있는 버튼 만들기

tkinter 모듈

- ✓ 파이썬은 그래픽 생성하기 위해 다양한 방법을 제공
- ✓ tkinter는 단순한 그림을 그릴 수 있을 뿐만 아니라
- ✓ 간단한 워드프로세서와 같이 완벽한 애플리케이션을 만들기 위해 사용된다
- ✓ tkinter를 이용해 버튼이 있는 기본적인 애플리케이션을 만들어 본다



```
>>> from tkinter import *
```

```
>>> tk = Tk()
```

```
>>> btn = Button(tk, text="click me")
```

```
>>> btn.pack()
```

- ✓ from 모듈명 import *를 이용하면 그 모듈명을 사용하지 않아도 모듈에 있는 것을 사용



14주차 1차시 : 캔버스에 그림 그리기

III. 학습 1 - tkinter 사용하여 클릭할 수 있는 버튼 만들기

tkinter 모듈

- ✓ 앞서 익힌 `import turtle`을 사용하면 그 안에 있는 것들을 사용하기 위해서 모듈명을 포함시켜야함



```
import turtle  
t = turtle.Pen()
```

- ✓ `import *`를 사용하면 `turtle.Pen`이라고 호출하지 않아도 되고,
- ✓ 여러 가지 클래스들과 함수를 가진 모듈을 사용할 경우에는 타이핑을 줄여줄 수 있기 때문에 유용
- ✓ 위 코드를 다시 쓰면



```
from turtle import *  
t = Pen()
```



14주차 1차시 : 캔버스에 그림 그리기

III. 학습 1 - tkinter 사용하여 클릭할 수 있는 버튼 만들기

tkinter 모듈

```
>>> from tkinter import *  
>>> tk = Tk()  
>>> btn = Button(tk,  
text="click me")  
>>> btn.pack()
```



- ✓ 다른 색상으로 실습을 쉽게 할 수 있도록 색으로 채워진 원을 그리는 코드는?
- ✓ tk 객체는 버튼이나 입력 상자 또는 그림을 그릴 수 있는 캔버스와 같은 다른 것들을 추가할 수 있는 기본적인 창을 생성
- ✓ tkinter 모듈이 제공하는 메인 클래스
- ✓ Tk 클래스의 객체를 생성하지 않으면 어떠한 그래픽이나 애니메이션을 만들 수 없다.
- ✓ btn.pack() : 버튼을 화면에 표시하라고 지시



14주차 1차시 : 캔버스에 그림 그리기

III. 학습 1 - tkinter 사용하여 클릭할 수 있는 버튼 만들기

tkinter 모듈

- ✓ didkme버튼을 하루 종일 클릭한다고 해도 약간의 코드를 수정하기 전까지 아무런 일도 일어나지 않는다.
- ✓ 코드를 수정하기 위해 이제 화면에 열려 있는 창을 닫는다.
- ✓ 먼저 텍스트를 출력하는 함수를 만들어 본다.

➡

```
>>> def hello():  
    print('hello there')
```

- ✓ 이 함수를 사용하도록 예제를 수정

➡

```
>>> from tkinter import *  
>>> tk = Tk()  
>>> btn = Button(tk, text="dick me", command=hello)  
>>> btn.pack()
```




14주차 1차시 : 캔버스에 그림 그리기

III. 학습 1 - tkinter 사용하여 클릭할 수 있는 버튼 만들기

tkinter 모듈

- ✓ 매개변수 `command`를 추가했고, 이것은 버튼이 클릭될 때 `hello` 함수를 사용
- ✓ 코드를 실행하면 버튼을 클릭할 때마다 셸에 "hello there"라고 쓰인 것을 보게 됨
- ✓ 다섯 번 클릭한 결과


```
7% clickable_button2.py - C:\Windows and Settings\Administrator\바탕 화면...
File Edit Format Run Options Windows Help

def hello():
    print('hello there')

from tkinter import *
tk = Tk()
btn = Button(tk, text="click me", command=hello)
btn.pack()

7% *Python 3.3.3 Shell*
File Edit Shell Debug Options Windows Help

Python 3.3.3 (v3.3.3:c3896275c0f6, Nov 18 2013, 21:18:40) [AMD64] on win32
Type "copyright", "credits" or "license()" for more information
>>> ===== RESTART =====
>>>
>>> hello there
hello there
hello there
hello there
hello there
```





14주차 1차시 : 캔버스에 그림 그리기

III. 학습 2 - 그림을 그리기 위한 캔버스 생성

캔버스 생성

☒ 캔버스를 생성하려면 파이썬에게 캔버스의 폭과 높이의 픽셀을 전달해야함

☒ 이 코드는 버튼 코드와 비슷



```
>>> from tkinter import *
```

```
>>> tk = Tk()
```

```
>>> canvas = Canvas(tk, width=500, height=500)
```

```
>>> canvas.pack()
```

☒ 500픽셀의 폭과 500픽셀의 높이로 캔버스의 크기를 보여 줌

☒ tk = Tk() 이 코드를 실행하면 다음과 같이 작은 창이 나타나고, canvas.pack()를 실행했을 때 여러분이 지정한 크기의 캔버스가 나타남

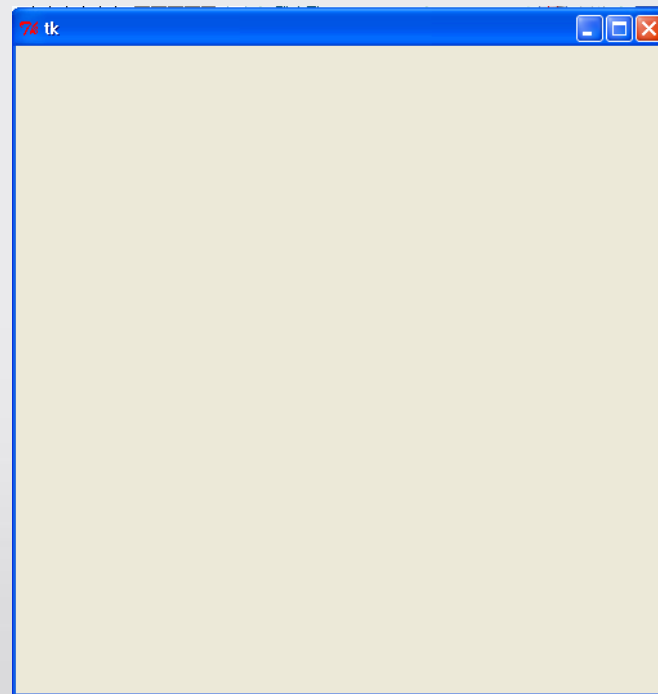


14주차 1차시 : 캔버스에 그림 그리기

III. 학습 2 - 그림을 그리기 위한 캔버스 생성

캔버스 생성

 실행결과





14주차 1차시 : 캔버스에 그림 그리기

III. 학습 2 - 여러 가지 도형 그리기

선 그리기

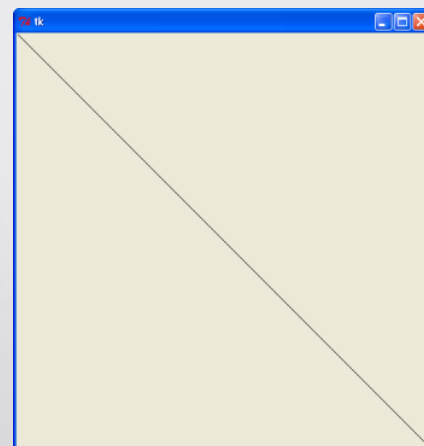
- ✓ 캔버스에 선을 그리기 위해서는 픽셀 좌표를 이용
- ✓ 좌표(coordinate)는 표면에 픽셀 위치를 결정
- ✓ 캔버스가 500픽셀의 폭과 500픽셀의 높이를 가지고 있을 경우
- ✓ 화면의 좌측 상단의 좌표는 (0, 0)이고, 우측 하단의 좌표는 (500, 500)
- ✓ 선을 그리기 위해서 시작 좌표를 (0, 0)으로 그리고 끝 좌표를 (500, 500)



```
>>> from tkinter import *  
>>> tk = Tk()  
>>> canvas = Canvas(tk, width=500, height=500)  
>>> canvas.pack()  
>>> canvas.create_line(0, 0, 500, 500)
```

1

- ✓ create_line 함수는 1을 반환하며, 이것은 식별자임





14주차 1차시 : 캔버스에 그림 그리기

III. 학습 2 - 여러 가지 도형 그리기

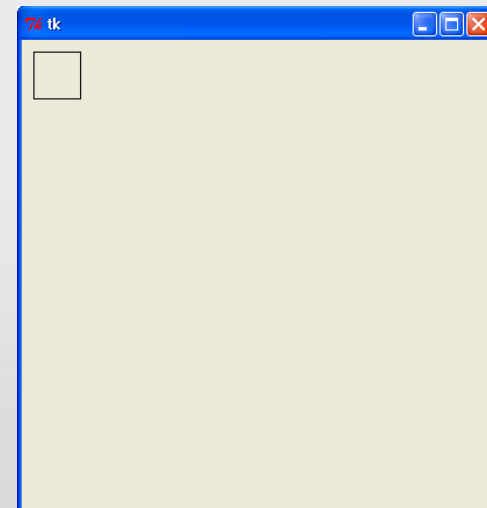
상자 그리기

- ✓ turtle 모듈로 상자를 그릴 때는 앞으로 이동했다가 돌려서 다시 이동하고 하는 식이었다.
- ✓ 즉, 앞으로 얼마나 이동하는가를 방향을 바꾸면서 사각형이나 정사각형을 그렸다.
- ✓ tkinter 모듈을 사용할 경우 각 꼭지점들의 좌표가 필요



```
>>> from tkinter import *  
>>> tk = Tk()  
>>> canvas = Canvas(tk, width=400, height=400)  
>>> canvas.pack()  
>>> canvas.create_rectangle(10, 10, 50, 50)
```

- ✓ 400픽셀의 폭과 400픽셀의 높이를 가진 캔버스를 생성
- ✓ 실행하면 창의 왼쪽 상단에 정사각형이 그려짐





14주차 1차시 : 캔버스에 그림 그리기

III. 학습 2 - 여러 가지 도형 그리기

상자 그리기

- ☒ create_rectangle에 전달하는 매개변수들은 정사각형의 왼쪽상단과 오른쪽 하단 좌표
- ☒ 다음 코드를 입력하고, 결과를 확인해보자

```
>>> from tkinter import *  
>>> tk = Tk()  
>>> canvas = Canvas(tk, width=400, height=400)  
>>> canvas.pack()  
>>> canvas.create_rectangle(10, 10, 300, 50)
```



14주차 1차시 : 캔버스에 그림 그리기

III. 학습 2 - 여러 가지 도형 그리기

다양한 크기의 직사각형 그리기

- ☒ 이작업을하기 위해서 random 모듈을 임포트
- ☒ 직사각형의상단 왼쪽과 하단 오른쪽의좌표로 랜덤 수를 이용
- ☒ random 모듈에서 제공하는 randrange 라는 함수를 이용
- ☒ randrange(100)이라고 호출하면 0에서 99 사이에 있는 숫자를 반환
- ☒ File > NewFile 를 선택해 새로운 창을 만들고 다음의 코드를 입력

```
from tkinter import *
import random
tk = Tk()
canvas = Canvas(tk, width=400,height=400)
canvas.pack()

def random_rectangle(width, height):
    x1 = random.randrange(width)
    y1 = random.randrange(height)
    x2 = random.randrange(x1 + random.randrange(width))
    y2 = random.randrange(y1 + random.randrange(height))
    canvas.create_rectangle(x1, y1, x2, y2)
```

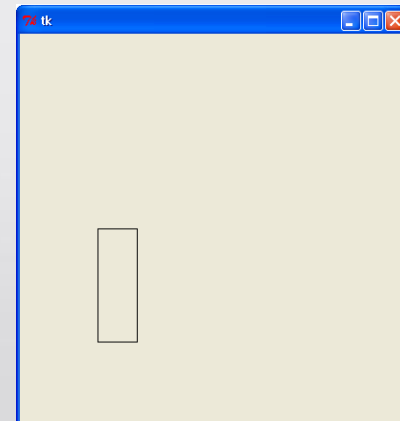


14주차 1차시 : 캔버스에 그림 그리기

III. 학습 2 - 여러 가지 도형 그리기

다양한 크기의 직사각형 그리기

- ✓ random_rectangle을 실행하기 위해 캔버스의 폭과 높이를 전달
- ✓ 다음의 코드를 입력한 함수 밑에 추가
→ random_rectangle(400, 400)
- ✓ 입력한 코드를 randomrect.py라는 이름으로 저장하고 Run>Run Module을 선택
- ✓ 난수를 발생시켰기에 여러분과 실행한 결과가 다를 수 있다.





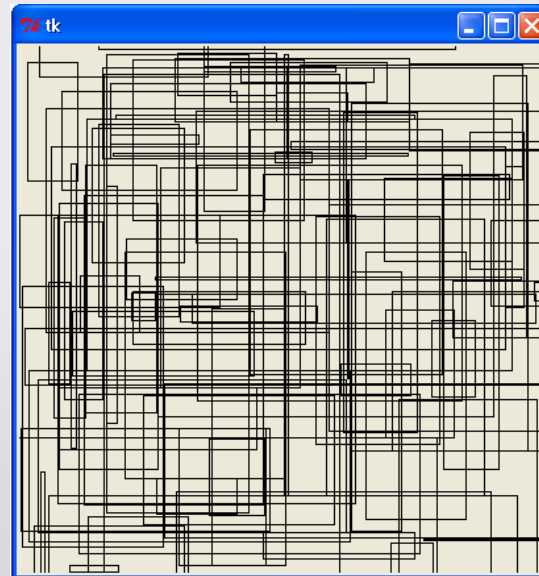
14주차 1차시 : 캔버스에 그림 그리기

III. 학습 2 - 여러 가지 도형 그리기

다양한 크기의 직사각형 그리기

- ✓ random_rectangle을 여러 번 호출하는 루프를 만들어서 화면을 직사각형으로 채워보자.
- ✓ 다음의 코드를 추가로 입력하여 저장한 다음, 다시 실행해 보자

➡ `for x in range(0, 100):`
 `random_rectangle(400, 400)`





『 이 콘텐츠는 2014학년도 학부교육 선도대학 육성사업(ACE)에 의하여 개발한 것임 』

