‘AI+X 인재양성 확대를 위한 교수자 연수

미니 프로젝트 구현 해보기

한국폴리텍 대구캠퍼스 강현우

제안 드리는 내용

- 목표 : 과정이 끝났을 때 프로젝트가 완성

- 가르치고 싶은 기능들을 구성
  - Ex) 배열, 루프, 랜덤, 조건문 등

- 기능을 배울 때 마다 프로젝트를 조금씩 업데이트

프로젝트 예시

선 긋기: 2중 For 루프

그림 배치 : Random

정답 : 조건문

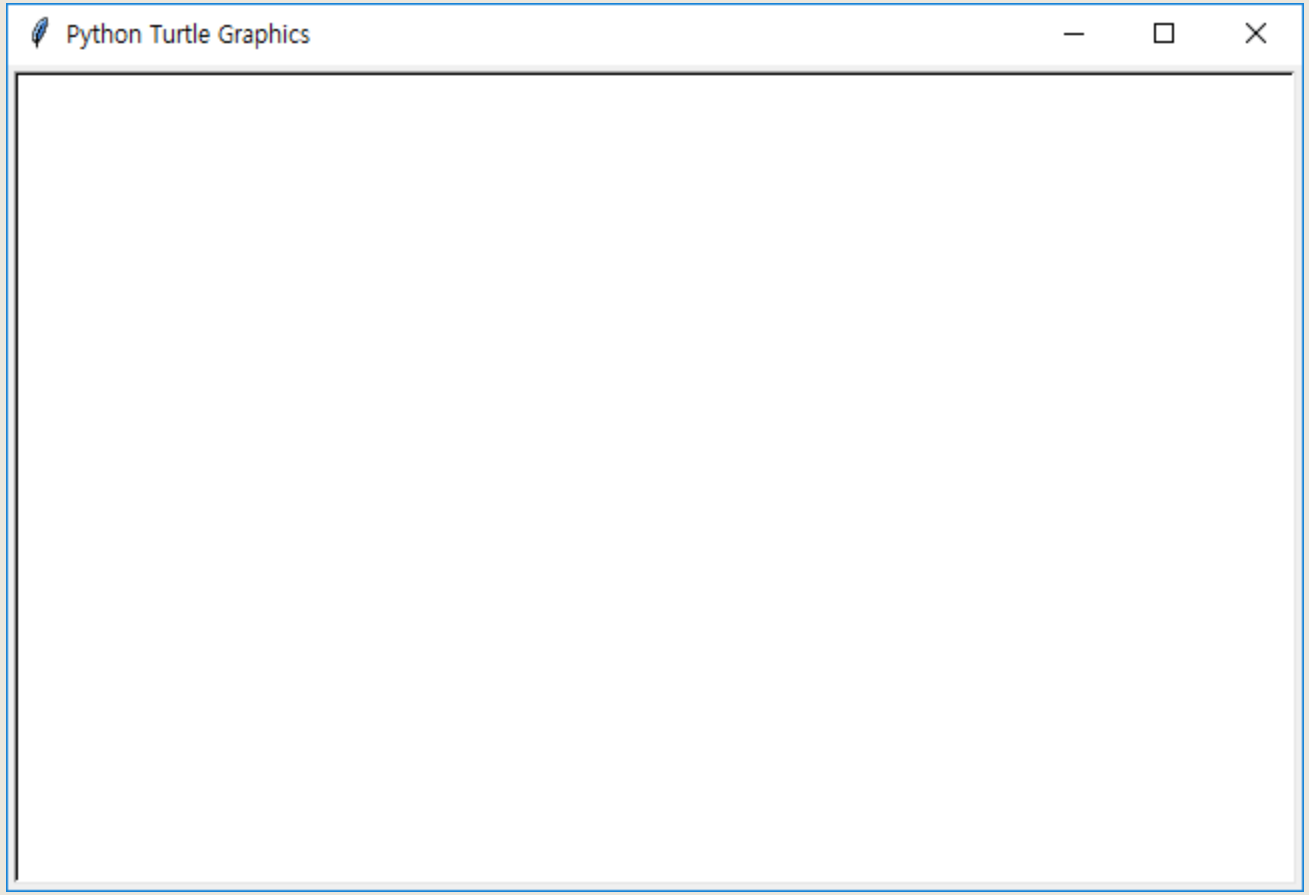
마우스 이벤트 처리
화면 전환

2중 배열

GUI 사용해보기

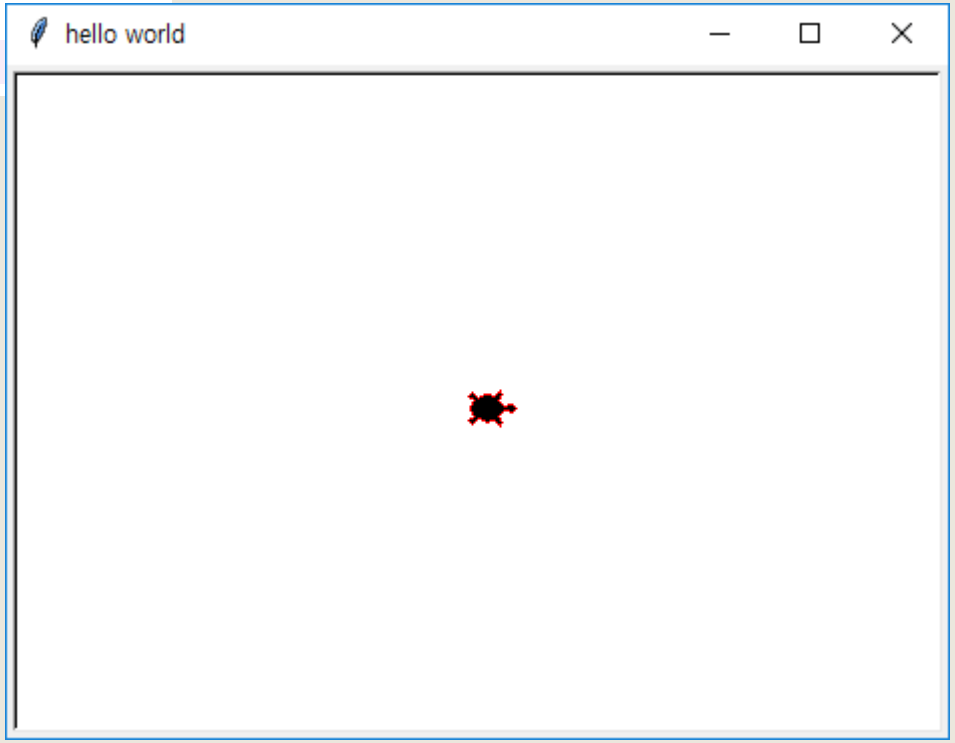
- Python turtle

```
 8 import turtle
 9
10 turtle.done()
11
```



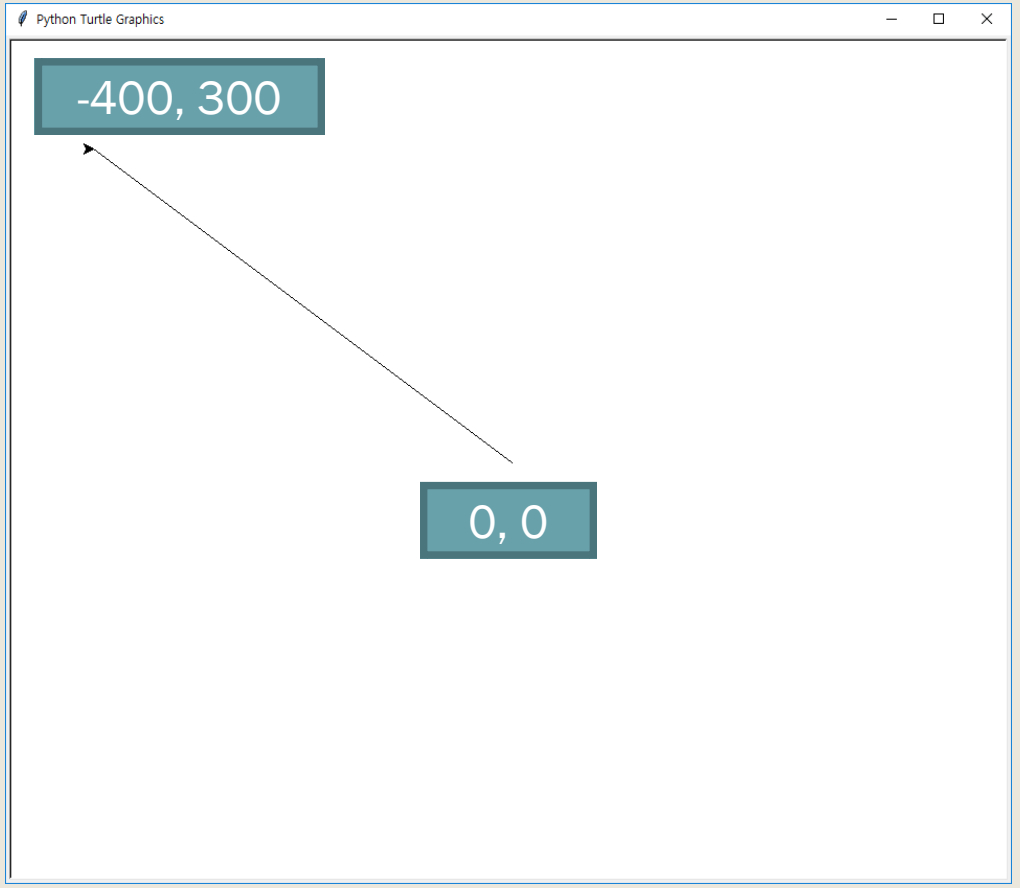
Hello world GUI

```
import turtle

turtle.title('hello world')
turtle.shape('turtle')
turtle.color('red', 'black') #첫 번째 인자 외곽선, 두 번째 인자 내부

turtle.done()
```



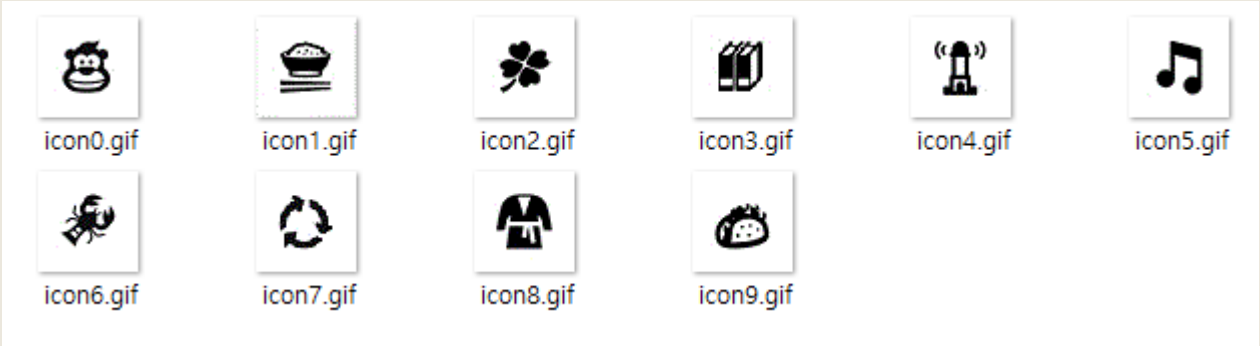
Position

- import turtle as t
- print(t.pos())
- #t.penup()
- t.goto(-400, 300)
- print(t.pos())
- t.showturtle()
- t.done()



Icon 만들기

- 인터넷에서 사용하고자 하는 icon을 다운
- Icon의 이름을 규칙을 가지도록 변경
- 크기를 동일하게 설정
- 확장자는 반드시 gif 파일로 설정

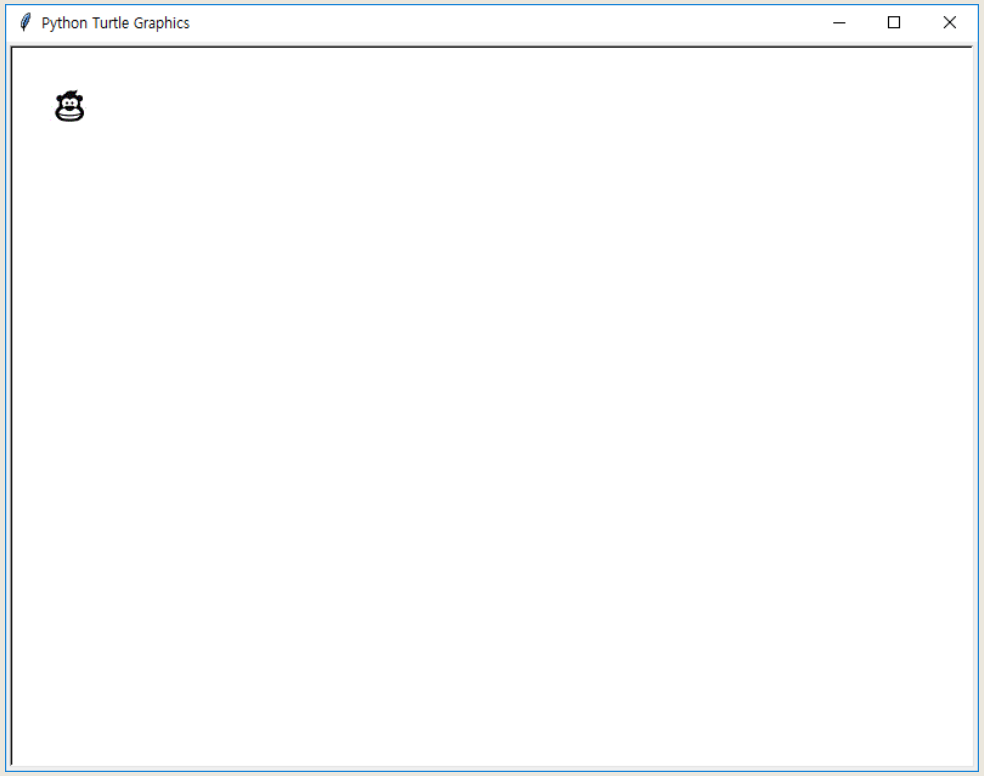


무료 Icon 사이트

- Icons8
  - https://icons8.com/icons
- Freevector
  - https://www.freevector.com/vector/icons
- 아이콘을 받아 오는 재미도 있음
- 동일한 크기로 받아올 수 있도록 해야함

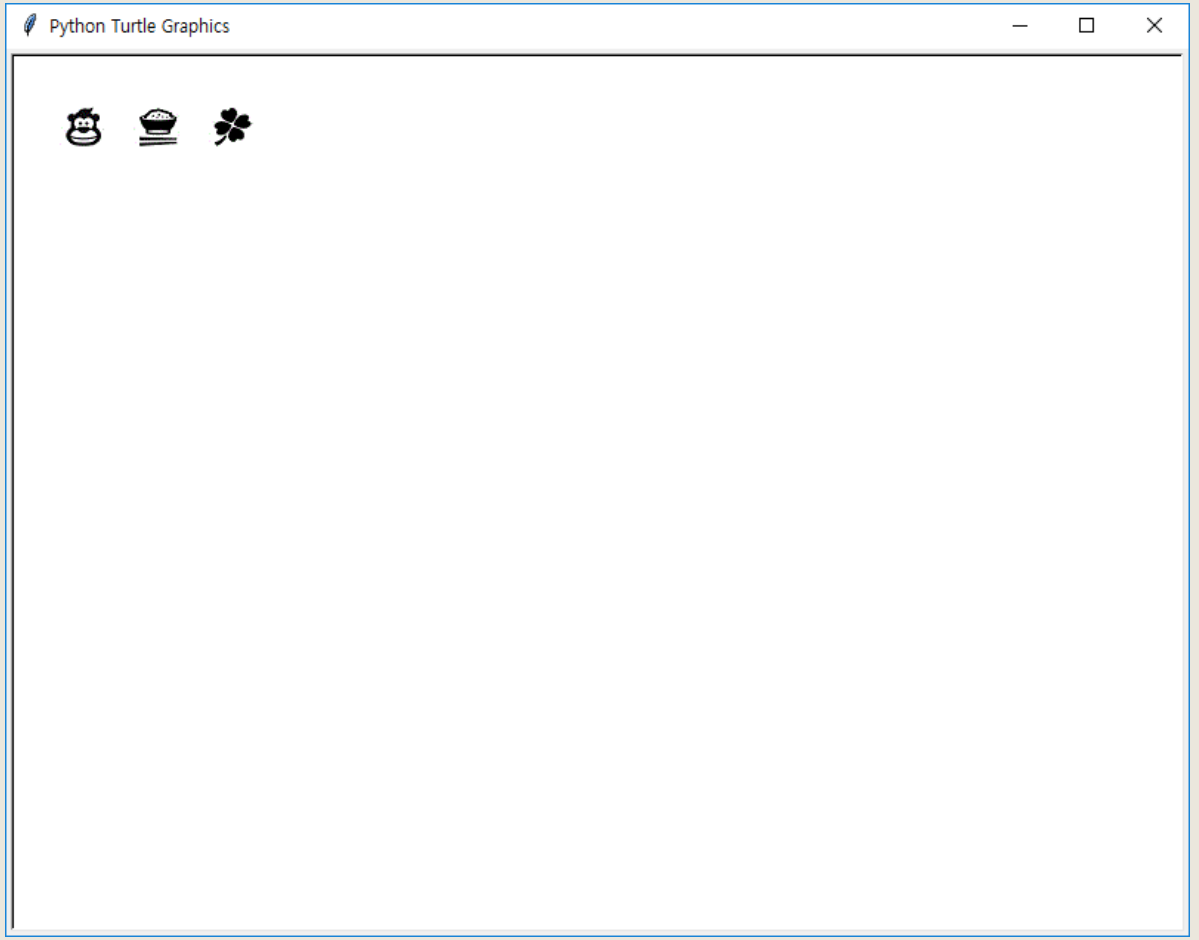
Turtle 모양 변경

```
 8 import turtle as t
 9
10 t.setup(width = 800, height = 600, startx = 0, starty = 0)
11 t.register_shape("icon/icon0.gif")
12 t.shape("icon/icon0.gif")
13
14
15 t.home()
16 t.penup()
17 t.goto(-350, 250)
18
19 print(t.shape())
20
21 t.showturtle()
22
23 t.done()
```



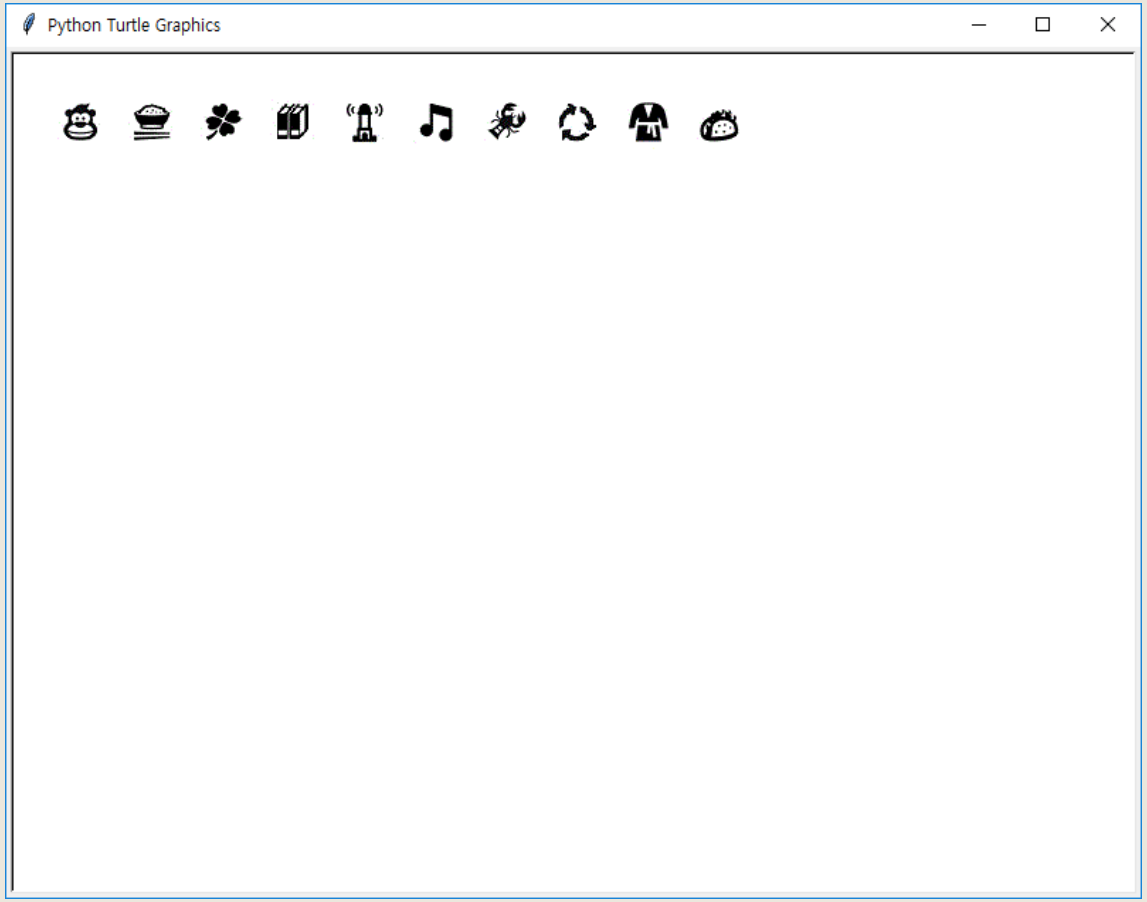
Stamp 이용하기

```
8  import turtle as t
9
10 t.setup(width = 800, height = 600, startx = 0, starty = 0)
11 t.register_shape("icon/icon0.gif")
12 t.register_shape("icon/icon1.gif")
13 t.register_shape("icon/icon2.gif")
14 t.penup()
15 t.home()
16
17 #첫 번째 아이콘
18 t.shape("icon/icon0.gif")
19 t.goto(-350, 250)
20 t.stamp()
21
22 #이동후 두 번째 아이콘
23 t.shape("icon/icon1.gif")
24 t.forward(50)
25 t.stamp()
26
27 #이동후 세 번째 아이콘
28 t.shape("icon/icon2.gif")
29 t.forward(50)
30 t.stamp()
31
32 #t.showturtle()
```



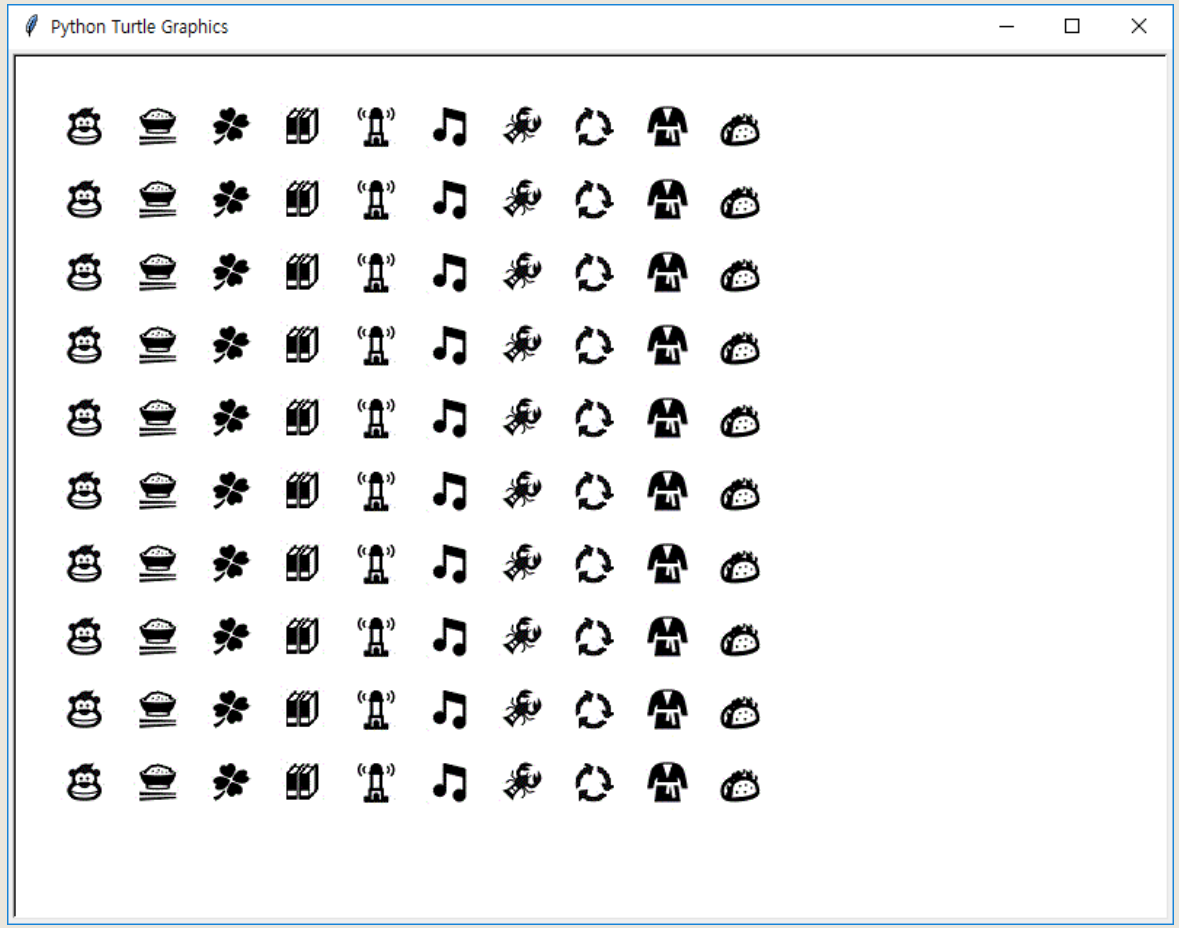
Stamp 반복 작업

```
 8 import turtle as t
 9
10 t.setup(width = 800, height = 600, startx = 0, starty = 0)
11
12 num_of_icon = 10
13
14 #아이콘 등록
15 for i in range (num_of_icon):
16     t.register_shape("icon/icon" + str(i) + ".gif")
17
18 t.penup()
19 t.home()
20
21 #최초 위치
22 t.goto(-350, 250)
23
24 #아이콘 모양 변경하며 10회 stamp
25 for i in range (num_of_icon):
26     t.shape("icon/icon" + str(i) + ".gif")
27     t.stamp()
28     t.forward(50)
29
30 t.done()
```



2중 for 루프

```
8  import turtle as t
9
10 t.setup(width = 800, height = 600, startx = 0, starty = 0)
11
12 num_of_icon = 10
13
14 #아이콘 등록
15 for i in range (num_of_icon):
16     t.register_shape("icon/icon" + str(i) + ".gif")
17
18 t.penup()
19 t.hideturtle()  #커서를 보이지 않게 함
20 t.home()
21
22 #최초 위치
23 start_x_pos = -350
24 start_y_pos = 250
25
26 #10줄 반복
27 for line in range (10):
28     #줄이 바뀔 때 마다 y의 좌표가 50씩 감소함
29     t.goto(start_x_pos, start_y_pos - (50 * line))
30     #모양 변경하며 10회 출력
31     for i in range (num_of_icon):
32         t.shape("icon/icon" + str(i) + ".gif")
33         t.stamp()
34         t.forward(50)
35
36 t.done()
```



Random

- random()
  - 0부터 1 사이의 float 숫자를 리턴
- randint (min, max)
  - min과 max 사이의 임의의 정수를 리턴
  - 주의 : min 과 max 가 포함된다

```
27 #10줄 반복
28 for line in range (10):
29     #줄이 바뀔 때 마다 y의 좌표가 50씩 감소함
30     t.goto(start_x_pos, start_y_pos - (50 * line))
31     #모양 변경하며 10회 출력
32     for i in range (10):
33         icon_num = random.randint(0, num_of_icon - 1)
34         t.shape("icon/icon" + str(icon_num) + ".gif")
35         t.stamp()
36         t.forward(50)
```

Event Handler

- 마우스 클릭, 키보드 버튼 등 입력에 대한 처리
- t.onkey (start_game, "space")
- t.listen()
  - Space 키가 눌리면 start_game 함수가 실행

- t.onkey (show_ans, "a")
- t.listen()
  - A 키가 눌리면 show_and 함수가 실행

- Reset 함수는 화면 클리어

```
31 #10x10 이미지를 그리는 함수
32 def start_game():
33     t.reset()
34     t.penup()
35     t.hideturtle()  #커서를 보이지 않게 함
36     #10줄 반복
37     for line in range (10):
38         #줄이 바뀔 때 마다 y의 좌표가 50씩 감소함
39         posX = start_x_pos
40         posY = start_y_pos - line * 50
41 
42         #모양 변경하며 10회 출력
43         for i in range (10):
44             rand_stamp(posX, posY)
45             posX += 50
46 
47 def show_ans():
48     t.reset()
49 
50 
51 t.onkey (start_game, "space")
52 t.listen()
53 
54 t.onkey (show_ans, "a")
55 t.listen()
```

정답 만들기

- 2자리 숫자를 선정 (ex. 24)
- 2자리 숫자의 십의 자리와 일의 자리를 각각 더함
  - 2 + 4
  - 24 // 10 = 2 (몫)
  - 24 % 10 = 4 (나머지)
- 원래 선정된 숫자에서 더한 수를 뺌
  - 24 – (2 + 4) = 18
- 계산된 숫자는 반드시 9의 배수
- 9의 배수 자리에 정답 아이콘을 배치

정답 배치

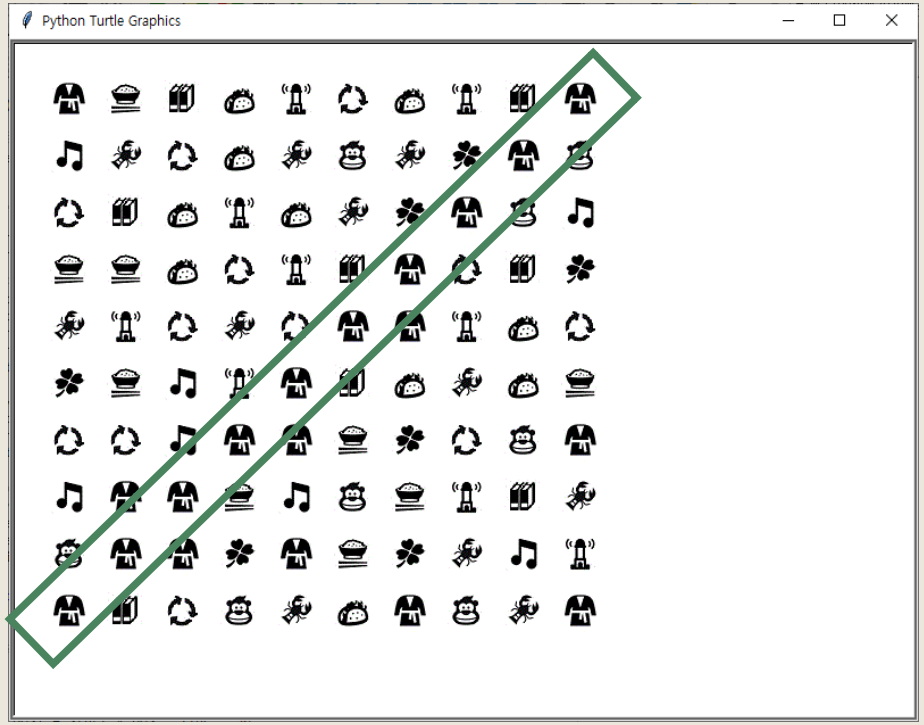
```python
#posX, posY 에 임의의 스탬프를 찍는 함수
def rand_stamp (posX, posY):
    t.goto(posX, posY)
    icon_num = random.randint(0, num_of_icon - 1)
    t.shape("icon/icon" + str(icon_num) + ".gif")
    t.stamp()

def draw_ans (posX, posY):
    t.goto(posX, posY)
    t.shape("icon/icon" + str(ans_index) + ".gif")
    t.stamp()

#10x10 이미지를 그리는 함수
def start_game():
    t.reset()
    t.penup()
    t.hideturtle()  #커서를 보이지 않게 함
    #매 판마다 정답은 달라져야 함
    global ans_index
    ans_index = random.randint(0, num_of_icon - 1)

    #10줄 반복
    for line in range (10):
        #줄이 바뀔 때 마다 y의 좌표가 50씩 감소함
        posX = start_x_pos
        posY = start_y_pos - line * 50

        #모양 변경하며 10회 출력
        for i in range (10):
            total_index = line * 10 + i
            if total_index % 9 == 0:
                draw_ans(posX, posY)
            else:
                rand_stamp(posX, posY)
            posX += 50
```

```python
def show_ans():
    t.reset()


t.onkey (start_game, "space")
t.listen()

t.onkey (show_ans, "a")
t.listen()
```

이벤트 등록

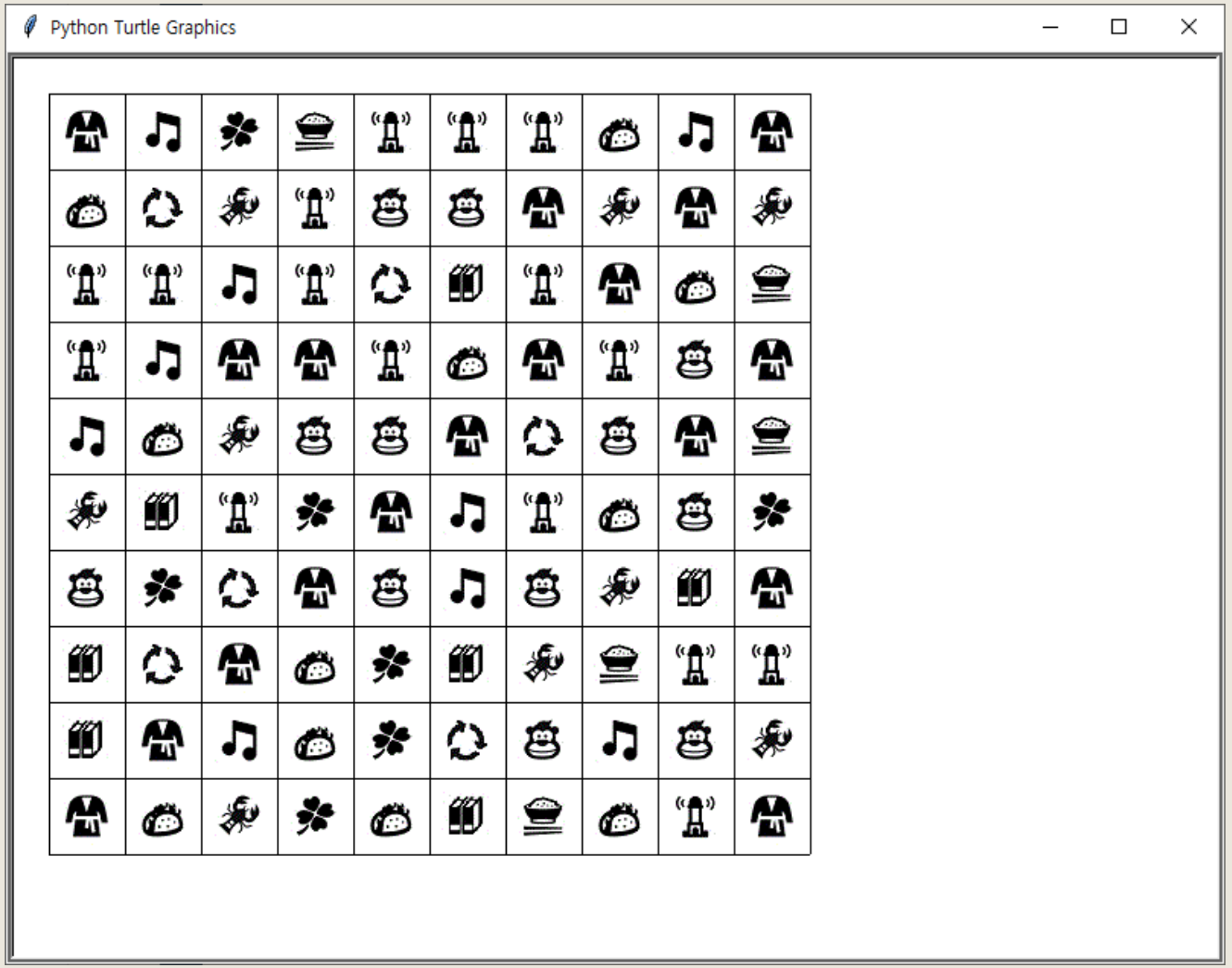
함수 분리

정답 배치

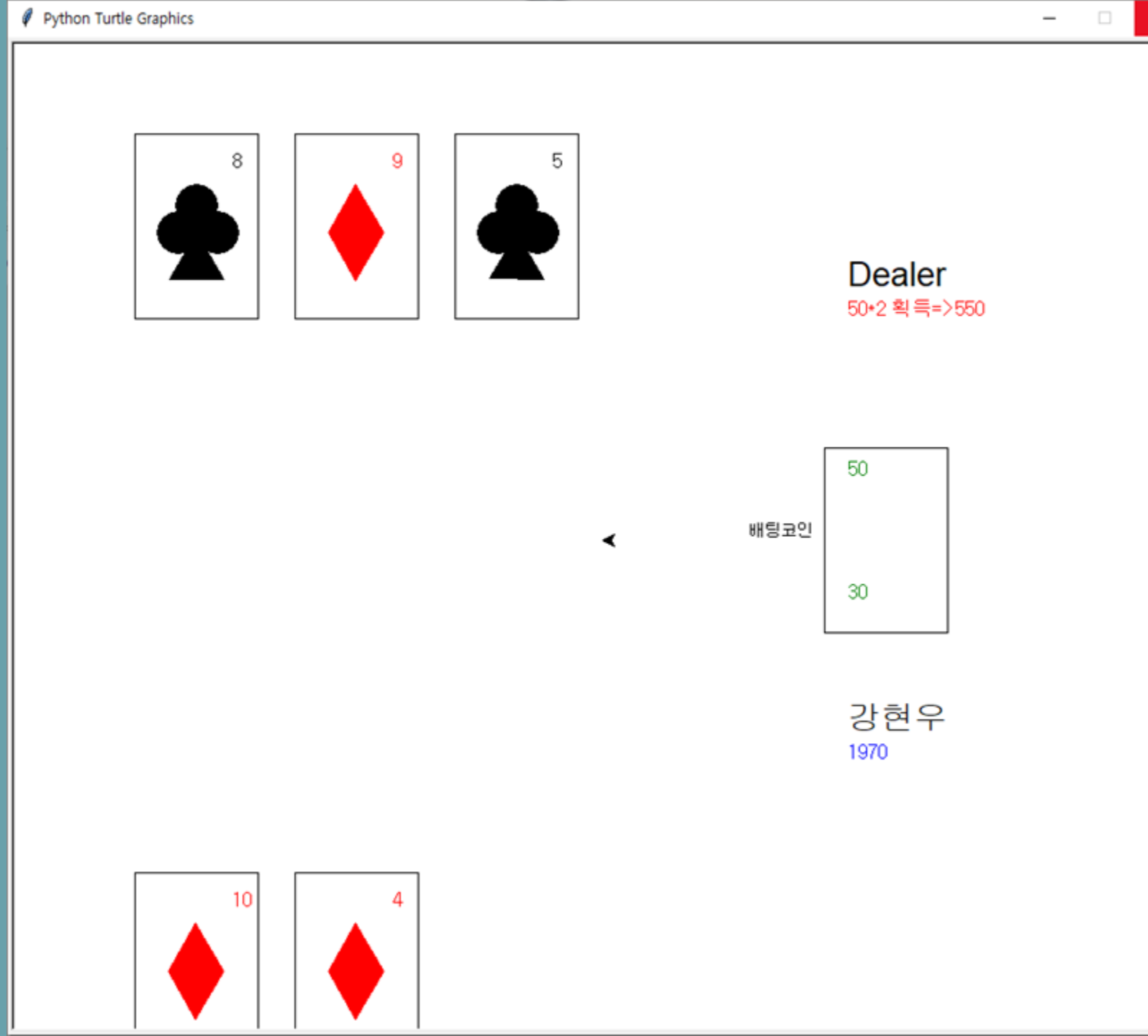


Line 그리기

```
def draw_line (startX, startY, endX, endY):
    t.goto(startX, startY)
    t.pendown()
    t.goto(endX, endY)
    t.penup()

def draw_board():
    for x in range (-375, -375 + 50*11, 50):
        draw_line(x, 275, x, -225)
    for y in range (-225, -225 + 50*11, 50):
        draw_line(-375, y, -375 + 50*10, y)
```

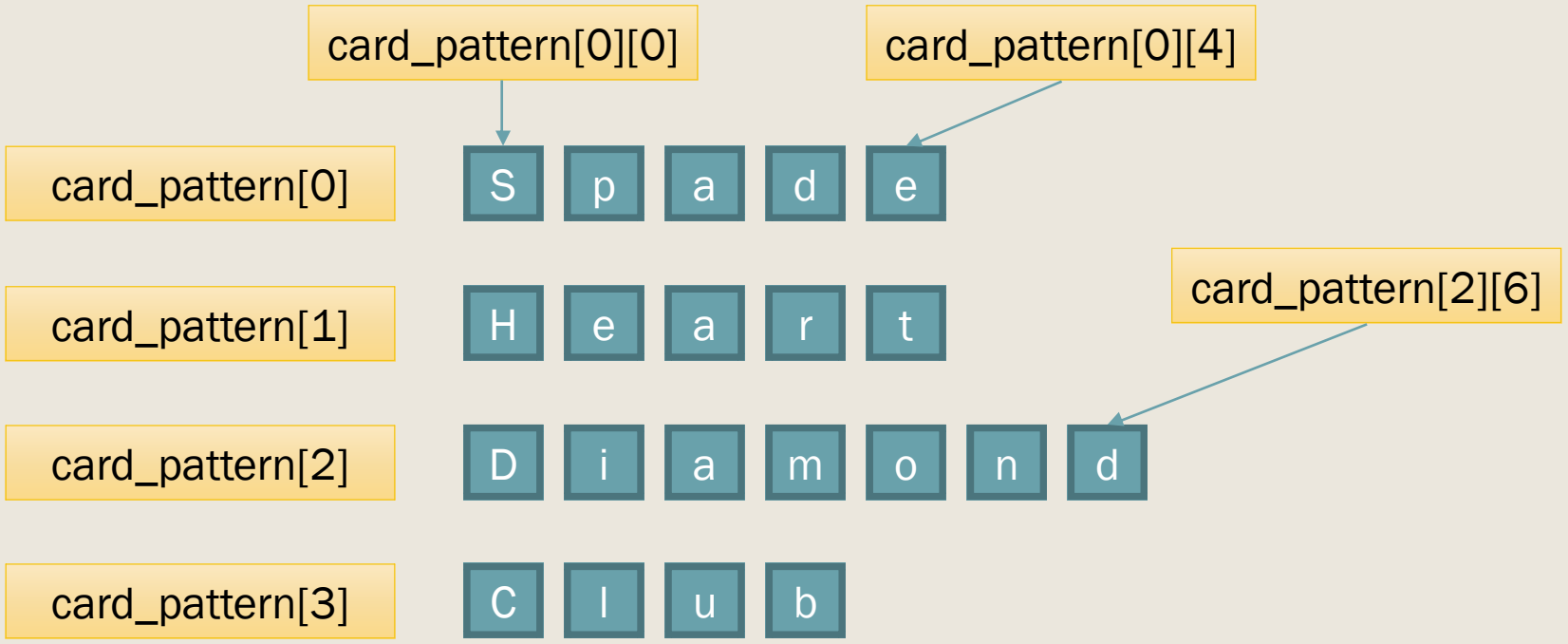


BLACK JACK


Python Turtle Graphics
8
9
5
Dealer
50*2 획득=>550
50
배팅코인
30
강현우
1970
10
4

List

- card_pattern = ["Spade", "Heart", "Diamond", "Club"]
- card_num = ['A', 2, 3, 4, 5, 6, 7, 8, 9, 10, 'J', 'Q', 'K']
- print (card_pattern[2][6])



3중 List?!

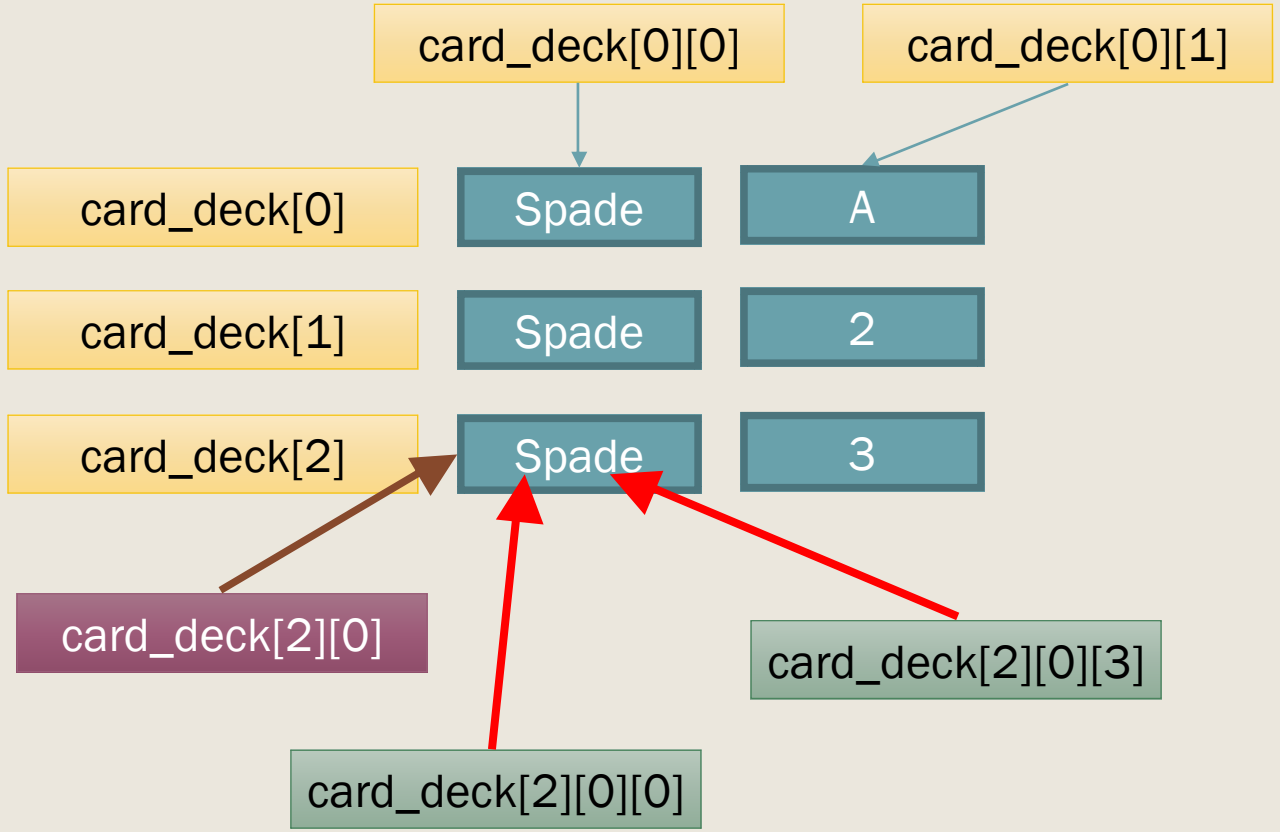
- card_deck = []
- for pattern in card_pattern:

  for num in card_num:

  card = [pattern, num]

  card_deck.append(card)
- print(card_deck[2][0][3])

- 3중으로 접근할 일은 없다!



shuffle

- import random
- print(card_deck)
- random.shuffle(card_deck)
- print(card_deck)

```
[['Spade', 'A'], ['Spade', 2], ['Spade', 3], ['Spade', 4], ['Spade', 5], ['Spade', 6], ['Spade', 7], ['Spade', 8], ['Spade',
9], ['Spade', 10], ['Spade', 'J'], ['Spade', 'Q'], ['Spade', 'K'], ['Heart', 'A'], ['Heart', 2], ['Heart', 3], ['Heart', 4],
['Heart', 5], ['Heart', 6], ['Heart', 7], ['Heart', 8], ['Heart', 9], ['Heart', 10], ['Heart', 'J'], ['Heart', 'Q'],
['Heart', 'K'], ['Diamond', 'A'], ['Diamond', 2], ['Diamond', 3], ['Diamond', 4], ['Diamond', 5], ['Diamond', 6], ['Diamond',
7], ['Diamond', 8], ['Diamond', 9], ['Diamond', 10], ['Diamond', 'J'], ['Diamond', 'Q'], ['Diamond', 'K'], ['Club', 'A'],
['Club', 2], ['Club', 3], ['Club', 4], ['Club', 5], ['Club', 6], ['Club', 7], ['Club', 8], ['Club', 9], ['Club', 10],
['Club', 'J'], ['Club', 'Q'], ['Club', 'K']]
```

```
[['Diamond', 7], ['Spade', 'K'], ['Heart', 3], ['Club', 7], ['Heart', 'K'], ['Heart', 5], ['Heart', 'Q'], ['Spade', 3],
['Diamond', 4], ['Diamond', 'J'], ['Diamond', 'K'], ['Spade', 8], ['Club', 10], ['Club', 'J'], ['Club', 'A'], ['Heart', 'J'],
['Heart', 7], ['Club', 8], ['Diamond', 6], ['Heart', 2], ['Diamond', 8], ['Spade', 'A'], ['Diamond', 9], ['Spade', 4],
['Spade', 2], ['Diamond', 2], ['Diamond', 'A'], ['Heart', 9], ['Club', 9], ['Club', 3], ['Heart', 8], ['Heart', 4], ['Club',
6], ['Spade', 7], ['Club', 5], ['Club', 'Q'], ['Spade', 9], ['Diamond', 3], ['Club', 2], ['Spade', 'Q'], ['Spade', 5],
['Heart', 'A'], ['Spade', 6], ['Club', 'K'], ['Diamond', 'Q'], ['Heart', 6], ['Spade', 'J'], ['Diamond', 10], ['Club', 4],
['Spade', 10], ['Diamond', 5], ['Heart', 10]]
```

List append, pop

- Append
  - 리스트의 맨 마지막에 원소를 추가
- Pop
  - 리스트의 맨 마지막 원소를 리턴

- card = card_deck.pop()
- print(card)
- card = card_deck.pop()
- print(card)

게임 진행 기본 룰

```
print("게임 시작")
print("카드를 한장 받는다")
user_input = None
while (user_input != 'q'):
    user_input = input("카드를 더 받으시겠습니까? (Hit: h / Stay: s): ")

    if (user_input == 'h' or user_input == 'H'):
        print("카드를 한장 받는다")
    elif (user_input == 's' or user_input == 'S'):
        print("점수를 계산한다")
        break
    else:
        print("잘못된 입력입니다.")
```

참가자가 여러 명이라면?

Player Class 설계

- 플레이어는 이름이 있다
- 플레이어는 게임에 사용할 돈을 가지고 있다
- 현재 플레이 중인 (혹은 보유 중인) 카드가 있다
- 현재 플레이 중인 카드를 이용해 계산된 점수가 있다

```
class Player:
    def __init__ (self, user_name, credit = 1000):
        self._name = user_name
        self._credit = credit
        self.holding_card = []
        self.score = 0
```

Class Player

```
class Player:
    def __init__ (self, user_name, credit = 1000):
        self._name = user_name
        self._credit = credit
        self.holding_card = []
        self.score = 0

    def hit(self):
        self.holding_card.append(card_deck.pop())
        self.calc_score()

    def calc_score(self):
        self.score = 0
        for card in self.holding_card:
            if card[1] == 'A' or card[1] == 'J' or card[1] == 'Q' or card[1] == 'K':
                self.score += 10
            else:
                self.score += card[1]

    def show_player_info(self):
        print ("=" * 50)
        print ("[ Player " + self._name + " ]" + " Score : " + str(self.score))
        for card in self.holding_card:
            print (card)
        print ("=" * 50)
```

Base Game

```
48 player1 = Player("강현우")
49 dealer = Player("Dealer")
50
51
52 print("게임을 시작합니다.")
53 print("각 플레이어가 카드를 한장 씩 받습니다")
54 player1.hit()
55 dealer.hit()
56
57 user_input = None
58 while (user_input != 'q'):
59
60     dealer.show_player_info()
61     player1.show_player_info()
62
63     user_input = input("카드를 더 받으시겠습니까? (Hit: h / Stand: s): ")
64
65     if (user_input == 'h' or user_input == 'H'):
66         print("{0} 플레이어가 카드를 한장 받습니다.".format(player1._name))
67         player1.hit()
68
69     elif (user_input == 's' or user_input == 'S'):
70         print("점수를 계산한다")
71         break
72
73     else:
74         print("잘못된 입력입니다.")
```

```
각 플레이어가 카드를 한장 씩 받습니다
================================================
[ Player Dealer ] Score : 10
['Diamond', 'A']
================================================
================================================
[ Player 강현우 ] Score : 6
['Heart', 6]
================================================

카드를 더 받으시겠습니까? (Hit: h / Stand: s): h
강현우 플레이어가 카드를 한장 받습니다.
================================================
[ Player Dealer ] Score : 10
['Diamond', 'A']
================================================
================================================
[ Player 강현우 ] Score : 13
['Heart', 6]
['Heart', 7]
================================================
```

Dealer Class

- 딜러도 플레이어이므로 기본 룰은 같다
  - 다만, 딜러는 Hit와 Stay를 임의로 할 수 없다
  - 플레이어 class를 상속하여 딜러 클래스를 만든다
  - 함수 하나만 추가하면 됨!

- class Dealer (Player):

```
def dealer_rule(self):
        self.calc_score()
        if self.score <= 16:    #딜러는 16이하이면 무조건 카드를 받아야 한다.
                print ("딜러가 카드를 한 장 받습니다.")
                self.hit()
        elif self.score > 16:
                print ("딜러가 Stay 합니다.")
```

Game 초기화

```
player1 = Player("강현우")
dealer = Dealer("Dealer")
print("게임을 시작합니다.")
print("각 플레이어가 카드를 두 장 씩 받습니다")
player1.hit()
dealer.hit()
player1.hit()
dealer.hit()
```

Main 루프

```
user_input = None
while (user_input != 'q'):
    dealer.show_player_info()
    player1.show_player_info()

    user_input = input("카드를 더 받으시겠습니까? (Hit: h / Stay: s): ")

    if (user_input == 'h' or user_input == 'H'):
        print("{0} 플레이어가 카드를 한장 받습니다.".format(player1._name))
        player1.hit()
        if (player1.score > 21):
            print("버스트 :( 패배하였습니다.")
            break

    elif (user_input == 's' or user_input == 'S'):
        dealer.dealer_rule()
        break

    else:
        print("잘못된 입력입니다.")
```

점수 계산

```
print("\n점수를 계산합니다")
dealer.show_player_info()
player1.show_player_info()
if (dealer.score > 21):
    print("딜러 버스트!! Player {0} 승리!".format(player1._name))
elif (player1.score == 21):
    print("블랙잭!! Player {0} 승리!".format(player1._name))
elif (player1.score > dealer.score):
    print("Player {0} 승리".format(player1._name))
elif (player1.score == dealer.score):
    print("무승부 입니다")
else:
    print("Dealer 승리")
```

함수 오버라이드

- 딜러의 show_player_info 함수를 재정의

```
def show_player_info(self):
    print (" " * 50 + "=" * 50)
    print (" " * 50 + "[ Player " + self._name + " ]" + " Score : " + str(self.score))
    for card in self.holding_card:
        print (" " * 49, end = " ")
        print (card)
    print (" " * 50 + "=" * 50)
```

```
점수를 계산합니다
                                                  ==================================================
                                                  [ Player Dealer ] Score : 21
                                                  ['Diamond', 4]
                                                  ['Spade', 7]
                                                  ['Spade', 'J']
                                                  ==================================================
==================================================
[ Player 강현우 ] Score : 17
['Spade', 5]
['Diamond', 2]
['Club', 'J']
==================================================
Dealer 승리
```

참고문헌

- 한국폴리텍대학 모두의 강좌
  - Python 프로그래밍 기초, Python 프로그래밍 고급

- Python 공식 사이트 튜토리얼
  - https://docs.python.org

- Wikidocs – Python 계단 밟기
  - https://wikidocs.net/book/2070

- 모던 파이썬 입문
  - 아나 벨 지음, 길벗