# 15차 연습문항

## **파이썬 심화문항: 그래픽스**

|  |
| --- |
| 수강생 공지 사항   * 제출은 프로그래머스를 통해 해주시기 바랍니다([링크](https://campus.programmers.co.kr/app/courses/24550/curriculum)). * 파일명은 아래와 같은 형식으로 제출해주세요   + 교육생번호\_이름\_교과목\_문항\_N차시.pdf   ex) DR-11111\_홍길동\_파이썬\_연습문항\_1차시.pdf  ex) DR-11111\_홍길동\_파이썬\_추가문항\_1차시.pdf   * 답은 "write your answer"에 적어주세요. 다만 코딩 문제의 경우 output까지 답에 포함시켜 주세요.   ex)     * 답은 캡쳐를 하셔도 되고 텍스트로 넣으셔도 됩니다. * 마감 기한은 문제가 나간 주 **일요일 23:59까지**입니다. |

### 다음 중 터틀 그래픽에 대한 설명으로 틀린 것은 무엇인가요?

A. turtle.forward(distance) 함수는 터틀이 현재 방향으로 지정한 거리(distance)만큼 이동하게 한다.

B. turtle.right(angle) 함수는 터틀이 오른쪽으로 지정한 각도만큼 회전하게 한다.

C. turtle.penup() 함수는 터틀이 이동할 때 선을 그리지 않게 한다.

D. turtle.start() 함수는 터틀 그래픽 창을 시작하게 한다.

**답**

|  |
| --- |
| D. turtle.start()라는 함수는 존재하지 않으며, 터틀 그래픽 창을 시작하고 유지하는 함수는 turtle.done()입니다. |

### 다음 중 터틀 그래픽으로 가능하지 않은 작업은 무엇인가요?.

A. 직선을 그리기

B. 원을 그리기

C. 다양한 색상을 사용하여 도형을 그리기

D. 음향 효과를 추가하기

E. 다각형을 그리기

**답**

|  |
| --- |
| D. 음향 효과를 추가하기 |

### 터틀 그래픽으로 정사각형을 그리는 프로그램을 작성하시오.

**답**

|  |
| --- |
| import turtle  # 화면 설정  screen = turtle.Screen()  screen.title("Turtle Graphics Example")  # 터틀 객체 생성  t = turtle.Turtle()    # 터틀의 속도 설정  t.speed(1) # 1은 기본 속도이며 값을 올리면 그리는 속도가 빨라짐    # 선 그리기  t.forward(100)   # 앞으로 100픽셀 이동  t.right(90)    # 오른쪽으로 90도 회전  t.forward(100)   # 앞으로 100픽셀 이동  t.right(90)    # 오른쪽으로 90도 회전  t.forward(100)   # 앞으로 100픽셀 이동  t.right(90)    # 오른쪽으로 90도 회전  t.forward(100)   # 앞으로 100픽셀 이동  # 터틀 그래픽 창을 닫지 않도록 대기  turtle.done() |

### 문제3번의 코드를 반복문으로 변환해 보세요.

**답**

|  |
| --- |
| import turtle as t    t.shape('turtle') # 터틀의 모양을 거북이 모양으로 설정합니다.  t.speed(1) # shape등 몇 개의 함수는 별도로 터틀객체를 생성하지 않아도 됨    for i in range(4):     t.forward(100)     t.right(90)    t.done() |

### 6각형을 그리는 코드를 작성해 보시오.

**답**

|  |
| --- |
| import turtle as t    t.shape('turtle')  for i in range(6):     t.forward(100)     t.right(360 / 6) |

### 원을 그리고, 원의 내부는 빨간색으로 채우시오. 그리고 원의 외곽선은 파랑색으로 그리는 프로그램을 작성하시오.

**답**

|  |
| --- |
| import turtle as t    # 터틀 객체 생성  turtle = t.Turtle()    # 원 그리기  turtle.color("blue")  # 외곽선 색상을 파란색으로 설정  turtle.fillcolor("red")  # 내부 채우기 색상을 빨간색으로 설정  turtle.begin\_fill()  # 채우기 시작  turtle.circle(100)  # 반지름이 100인 원 그리기  turtle.end\_fill()  # 채우기 종료    # 화면 유지  t.done() |

### 반지름이 120 이고 원주 각도가 90인 부채꼴 형상의 도형을 그려 보시오.(피자의 1/4조각 같은 모양.)

**답**

|  |
| --- |
| import turtle as t    # Create a turtle object  turtle = t.Turtle()  # Set radius and angle  radius = 120  angle = 90    # Draw the fan shape (pizza slice)  turtle.penup()  turtle.goto(0, -radius)  # Move to the starting point at the bottom center  turtle.pendown()  turtle.begin\_fill()  # Start filling the shape    turtle.setheading(0)  # Face right  turtle.forward(radius)  # Move forward along the radius  turtle.left(90)  # Turn left 90 degrees    turtle.circle(radius, angle)  # Draw the curved edge of the pizza slice    turtle.goto(0, -radius)  # Go back to the starting point  turtle.end\_fill()  # End filling the shape    # Keep the window open  t.done() |

### 이 코드는 각 반복에서 터틀이 91도씩 왼쪽으로 회전하면서 선을 그리는 예제입니다. colors[i % 4]를 사용하여 색상을 순환하면서 그립니다. 프로그램을 실행해서 어떤 도형이 그려지는지 확인해 보세요.

|  |
| --- |
| Example Code |
| import turtle  turtle.speed(0) #숫자가 클수록 빠르게 그립니다. 0는 가장 빠른 속도입니다.  turtle.bgcolor("black") # 배경색  colors = ["red", "yellow", "blue", "green"]  for i in range(100):      turtle.color(colors[i % 4]) # colors 인덱스를 만들어 색상을 순환시킵니다.      turtle.forward(i \* 4) # 현재 반복 횟수 i에 4를 곱한 만큼 앞으로 전진합니다.  #이로 인해 반복할 때마다 그리는 선의 길이가 점점 길어집니다.      turtle.left(91)  turtle.done() |

**답**

|  |
| --- |
|  |

### 키보드의 방향키(화살표)를 누르면 거북이가 화살표 방향으로 이동하면서 선을 그리는 프로그램을 작성하라.

- 위, 아래 화살표를 누르면 전진, 후진하면서 선을 그린다..

- 오른쪽, 왼쪽 화살표를 누르면 우방향, 좌방향으로 회전하면서 선을 그린다

<참조함수>

- turtle.pendown() 현재 위치에 도장을 찍어 선을 그림

- turtle.right(5) 오른쪽으로 5도 회전 : turtle.left(5) 왼쪽으로 5도 회전

- t.Screen().onkeypress(move\_forward, "Up") 키보드 이벤트 처리 / 위쪽 화살표 키

- t.Screen().onkeypress(move\_backward, "Down") 아래쪽 화살표 키

- t.Screen().onkeypress(move\_right, "Right") 오른쪽 화살표 키

- t.Screen().onkeypress(move\_left, "Left") 왼쪽 화살표 키

**답**

|  |
| --- |
| import turtle as t    # 터틀 객체 생성  turtle = t.Turtle()    # 이동 함수 정의  def move\_forward():      turtle.forward(5)  # 앞으로 5픽셀 이동      turtle.pendown()  # 현재 위치에 도장을 찍어 선을 그림    def move\_backward():      turtle.backward(5)  # 뒤로 5픽셀 이동      turtle.pendown()  # 현재 위치에 도장을 찍어 선을 그림    def move\_right():      turtle.right(5)  # 오른쪽으로 5도 회전      turtle.forward(5)  # 앞으로 5픽셀 이동      turtle.pendown()  # 현재 위치에 도장을 찍어 선을 그림    def move\_left():      turtle.left(5)  # 왼쪽으로 5도 회전      turtle.forward(5)  # 앞으로 5픽셀 이동      turtle.pendown()  # 현재 위치에 도장을 찍어 선을 그림    # 초기 설정  turtle.speed(0)  # 최고 속도로 설정  turtle.shape('turtle')  # 터틀 모양으로 설정    # 키보드 이벤트 처리  t.Screen().onkeypress(move\_forward, "Up")  # 위쪽 화살표 키  t.Screen().onkeypress(move\_backward, "Down")  # 아래쪽 화살표 키  t.Screen().onkeypress(move\_right, "Right")  # 오른쪽 화살표 키  t.Screen().onkeypress(move\_left, "Left")  # 왼쪽 화살표 키    # 이벤트 리스너 활성화  t.Screen().listen()    # 터틀 그래픽 창을 닫을 때 종료하기 위해 대기  t.done() |

### 산포그래프(Scatter plot)는 데이터 포인트를 점으로 표시하여 두 변수간의 관계를 시각화하는 그래프입니다. 주로 두 변수 간의 상관관계, 군집성, 이상치 등을 시각적으로 분석할 때 사용됩니다. 주로 matplotlib 와 seaborn 라이브러리를 사용하여 산포그래프를 그려라.

아래의 두 변수 x, y에 대한 산포그래프를 matplotlib를 사용하여 그려보세요.

- matplotlib가 설치되지 않았으면 command prompt에서 pip install matplotlib를 실행하여 설치하세요.

x = [1, 2, 3, 4, 5]

y = [2, 3, 5, 7, 11]

**답**

|  |
| --- |
| import matplotlib.pyplot as plt    # 데이터 준비  x = [1, 2, 3, 4, 5]  y = [2, 3, 5, 7, 11]    # 그래프 그리기  plt.plot(x, y, marker='o', linestyle='-', color='b', label='Data 1')    # 그래프에 제목과 레이블 추가  plt.title('Simple Line Plot')  plt.xlabel('X-axis')  plt.ylabel('Y-axis')    # 범례 추가  plt.legend()    # 그래프 표시  plt.show() |