

과학 (Science)

1 DNA 염기서열의 순서 바꾸기

프로그램 p08-01 ... DNA 염기서열의 순서 바꾸기

관련 학습 ... 딕셔너리(dictionary) 자료형

2 자유 낙하와 포물선 운동 궤적 그리기

2.1 자유 낙하 운동 궤적 그리기

프로그램 p08-02-1 ... 물체를 자유 낙하시키기

관련 학습 ... 모듈 임포트(import)

2.2 자유 낙하 운동 궤적 그리기

프로그램 p08-02-1 ... 물체의 포물선 운동

관련 학습 ... math 수학 모듈, 라디안(radian)

Thinking!

잠깐! Coding

Coding! Programming



1. 문자열을 입력받아 문자열의 길이를 구하고, 문자열의 첫 번째 문자, 두 번째 문자, 마지막 문자를 출력해보자.

문자열 : Python Programming

문자열 길이 : 18

첫 번째 문자 : P

두 번째 문자 : y

마지막 문자 : g




문자열은 인덱스(index)에 의해 개별 문자를 추출하며, 문자열의 인덱스는 0부터 문자열길이-1 까지 부여된다. 문자열의 길이는 `len()` 함수로 계산한다. 첫 번째 문자는 `str[0]`, 두 번째 문자는 `str[1]`, 마지막 문자는 `str[len(str)-1]`에 해당한다. 이때 마지막 문자의 인덱스가 `len(str)`이 아닌 `len(str)-1`임에 주의해야 한다.

2. 문자열을 입력받아 for 문을 이용하여 개별 문자로 출력해보자. 그리고 for 문을 이용하여 입력받은 문자열의 역순으로 개별 문자를 출력해보자.

문자열 : Python Programming


개별 문자 출력 : Python Programming

역순 개별 문자 출력 : gnimmargorP nohtyP

 문자열은 인덱스(index)에 의해 개별 문자를 추출하며, 문자열의 인덱스는 0부터 문자열길이-1까지 부여된다. 문자열의 길이는 len() 함수로 계산하며, for 문에서 range(len(str))와 range(len(str)-1, -1, -1)을 반복한다.

3. 0~100 사이의 점수를 입력받아 입력한 점수가 0~100인 경우 점수에 대한 A, B, C, D, F 등급을 출력하고, 범위에 해당하지 않으면 “입력 가능한 점수 범위는 0~100입니다.”를 출력해보자. 점수에 대한 등급 판정은 if-elif-else 문을 이용하여 점수가 90~100일 때 “A”, 80~89일 때 “B”, 70~79일 때 “C”, 60~69일 때 “D”, 0~59일 때 “F”로 출력한다.

```
점수 : -5
입력 가능한 점수 범위는 0~100입니다.
점수 : 85
85 : B
점수 : 50
50 : F
```

 입력한 점수 범위가 0~100인지 구분하기 위하여 `score >= 0 and score <= 100` 조건을 사용하고, if-elif-else 문을 이용하여 점수에 대한 등급을 판정한다.


4. 'Basic Coding 3번 문제'를 참고하여, if-elif-else 문 대신에 딕셔너리를 이용하여 점수에 대한 등급을 출력해보자.

점수 : 85

85 : B

점수 : 55

55 : F

 딕셔너리는 `deg = { 10:'A', 9:'A', 8:'B', 7:'C', 6:'D', 5:'F', 4:'F', 3:'F', 2:'F', 1:'F', 0:'F' }`와 같이 선언하며, 점수 `score`를 10으로 정수 나누기(`score // 10`)를 하여 결과 값이 딕셔너리의 키와 일치하는 값을 구하여 등급을 출력한다.

5. 딕셔너리를 이용하여 제품:값의 형태로 items = { "라면":650, "우유":1100, "콜라":1200, "캔커피":500, "과자":700 }을 선언해보자. while 문을 이용하여 무한 반복하면서 제품을 입력받아 제품에 대한 값들의 합계를 출력해보자. 아무 입력도 하지 않은 채로 Enter 키를 눌러 빈 문자열이 입력되면 무한 반복을 멈추고 전체 합계를 출력한다.

제품명 : 우유

[우유:1100] > 1100

제품명 : 콜라

[콜라:1200] > 2300

제품명 : 라면

[라면:650] > 2950

제품명 : 라면

[라면:650] > 3600

제품명 : 과자

[과자:700] > 4300

제품명 : 딸기

딸기 는 미등록 제품입니다.

제품명 :

총 금액 : 4300



무한 반복은 while True: 문으로 작성하고, 딕셔너리의 제품 키에 대한 값은 items[it]로 구한다. 제품을 입력한 후의 결과 출력 형식은 print("[%s:%d] > %d"%(it, items[it], s)) 문으로 작성한다.

6. time 모듈을 import한 후, for 문을 이용하여 1부터 5까지의 숫자를 출력해보자. 각 숫자를 출력한 후 sleep() 함수를 이용하여 프로그램을 1초간 멈추게 해보자.

```
1 2 3 4 5
```



sleep() 함수는 time 모듈 내의 함수이며, 프로그램의 실행을 1초간 멈추려면 time.sleep(1)로 작성한다.

7. math 모듈을 import한 후, 실숫값을 입력받아 ceil() 함수, floor() 함수, trunc() 함수 연산의 결과를 출력해보자.

함수	기능	반환값/결과값
ceil(x)	$N \geq x$ 를 만족하는 가장 작은 정수 N을 반환(올림 연산) <code>math.ceil(3.14)</code>	int 4
floor(x)	$N \leq x$ 를 만족하는 가장 큰 정수 N을 반환(내림 연산) <code>math.floor(3.14)</code>	int 3
trunc(x)	x의 정수 부분만을 반환(버림 연산) <code>math.trunc(3.14)</code>	int 3

실수 : 3.14

3.14 : 4

3.14 : 3

3.14 : 3



ceil(), floor(), trunc() 함수는 math 모듈을 import 해야 사용할 수 있다.

8. math 모듈 내의 함수를 사용할 때 `math.` 부분을 사용하지 않도록 math 모듈을 import한 후, 정숫값을 입력받아 `sqrt()` 함수 연산의 결과를 출력해보자.

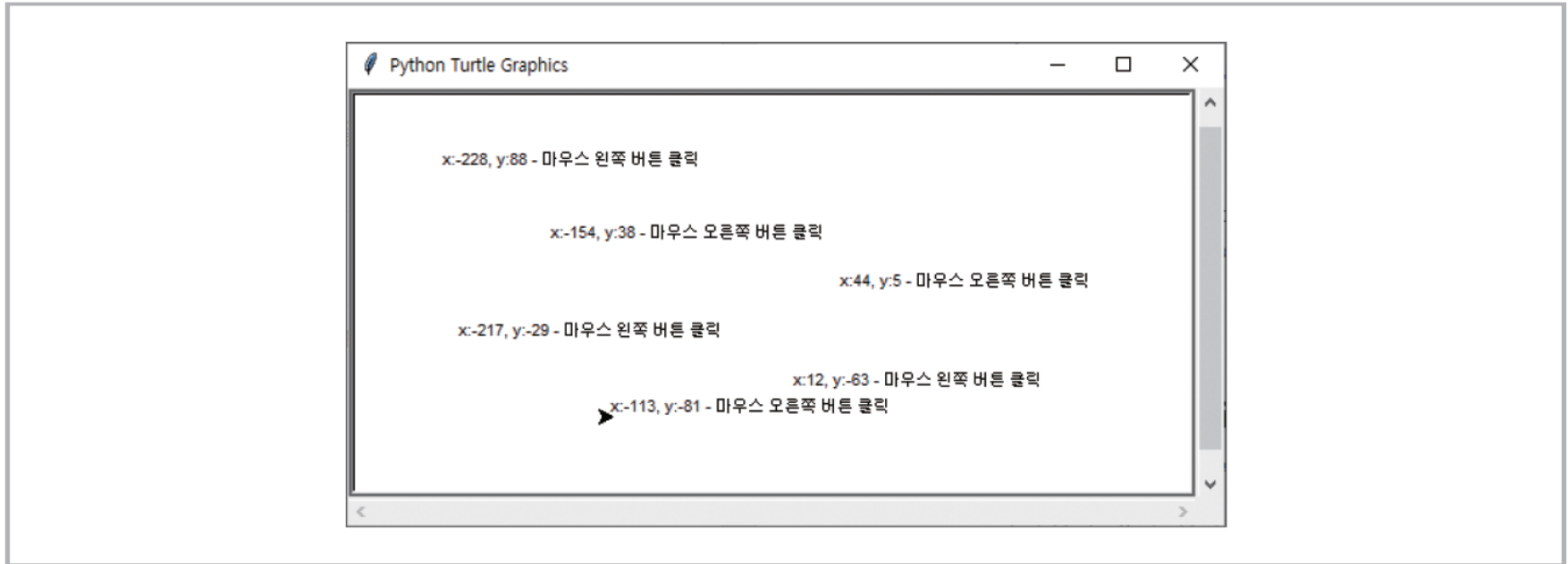
정수 : 3


3.0 : 1.7320508075688772



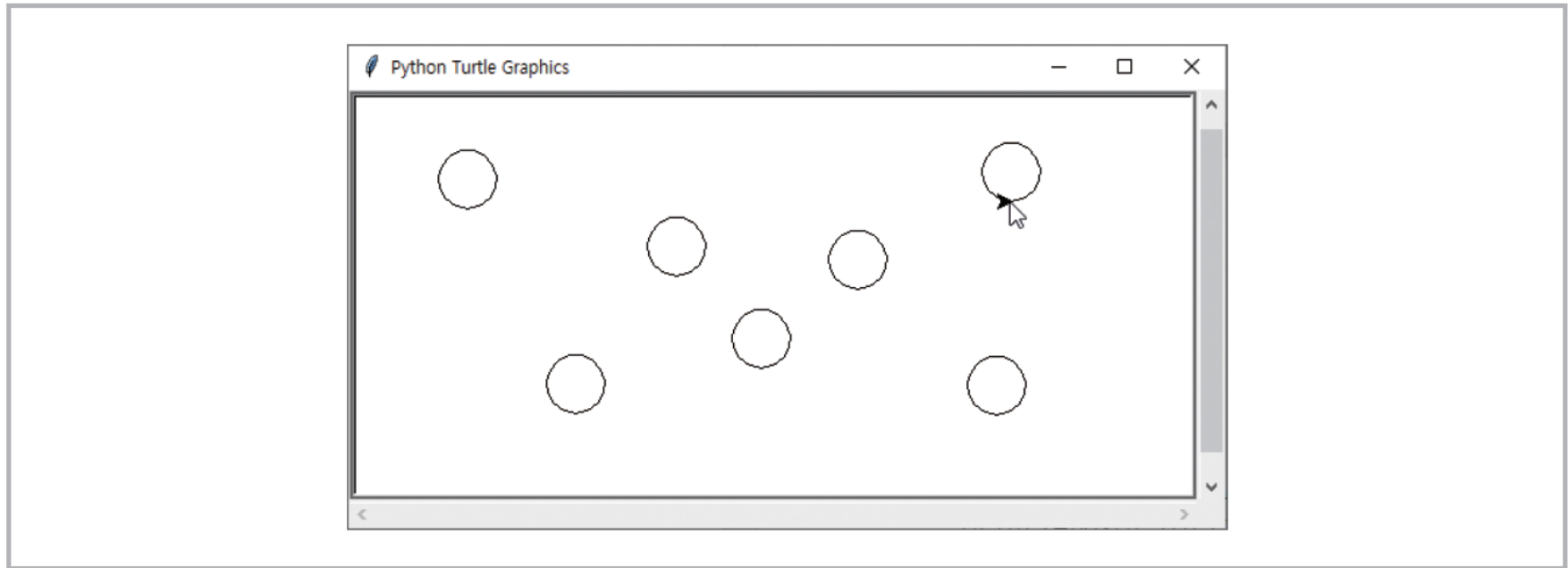
`sqrt()` 함수는 math 모듈을 import 해야 사용할 수 있으며, `from math import sqrt`로 import하면 `math.` 부분을 생략할 수 있다.


9. 터틀 스크린에서 마우스 왼쪽 버튼, 마우스 오른쪽 버튼을 클릭하면 해당 위치에 'x:50, y:100 - 마우스 왼쪽 버튼 클릭', 'x:50, y:100 - 마우스 오른쪽 버튼 클릭' 형식으로 출력해보자.



-  마우스 왼쪽 버튼을 클릭하면 `write_xyleft()` 함수가 동작하도록 콜백 함수를 등록하고, 마우스 오른쪽 버튼을 클릭하면 `write_xyright()` 함수가 동작하도록 콜백 함수를 등록한다.


10. 터틀 스크린에서 마우스 왼쪽 버튼을 클릭하면 해당 위치에 반지름이 20인 원을 그려보자. 단, 원을 그리는 동작 이외의 움직임은 동작에서는 선을 그리지 않는다. 마우스 오른쪽 버튼을 클릭하면 터틀 스크린 내의 모든 내용을 지워보자.



 마우스 왼쪽 버튼을 클릭하면 `write_xy()` 함수가 동작하도록 콜백 함수를 등록하고, 마우스 오른쪽 버튼을 클릭하면 `screen_clear()` 함수가 동작하도록 콜백 함수를 등록한다. 터틀 스크린 내의 모든 내용을 지우려면 `clear()` 함수를 사용한다. 선을 그릴 때는 `pendown()` 함수를 사용하고, 선을 그리지 않아야 할 때는 `penup()` 함수를 사용한다.

1. 딕셔너리를 이용하여 1분기 8개, 2분기 6개, 3분기 10개, 4분기 13개의 판매 실적을 등록한 후, 해당 분기별 판매 실적을 '#' 문자를 이용하여 다음과 같이 출력해보자.

```
1 분기 : ##### ( 8 )  
2 분기 : ##### ( 6 )  
3 분기 : ##### ( 10 )  
4 분기 : ##### ( 13 )
```

 sales = {1:8, 2:6, 3:10, 4:13}로 선언하고, for 문을 이용하여 딕셔너리의 모든 키에 해당하는 값을 print(i, "분기 :", '#' * sales[i])와 같이 출력한다.


2. 딕셔너리를 이용하여 다음의 제품 재고 현황을 등록한 후, 재고수를 입력받아 해당 재고수보다 작은 재고수인 제품을 모두 출력해보자.

제품	공책	연필	지우개	복사지
재고수	325	427	125	510

파악 재고수 기준 : 400

공책 : 325

지우개 : 125

 items = {'공책':325, '연필':427, '지우개':125, '복사지':510}로 선언하고, for 문을 이용하여 딕셔너리의 모든 키에 해당하는 값의 크기가 입력된 재고수보다 작으면 해당 제품명과 재고수를 출력한다.

3. 딕셔너리를 이용하여 비어 있는 영한사전에 단어를 등록해보자. while 문을 이용하여 무한 반복하면서 영어 단어와 한글 단어를 입력받아 등록하고, 아무 입력도 하지 않은 채로 Enter 키를 눌러 영어 단어와 한글 단어에 빈 문자열이 입력되면 무한 반복을 멈추고 딕셔너리에 등록된 모든 키와 값들을 출력한다.

영어 단어 : Python

한글 단어 : 파이썬

영어 단어 : string

한글 단어 : 문자열

영어 단어 : iteration

한글 단어 : 반복

영어 단어 : selection

한글 단어 : 선택

영어 단어 :

한글 단어 :

```
{'Python': '파이썬', 'string': '문자열', 'iteration': '반복', 'selection': '선택'}
```



engkor_dict = dict() 문장을 통해 비어 있는 딕셔너리를 선언하며, 입력한 영어 단어와 한글 단어의 등록은 engkor_dict[eng] = kor 문장을 통해 이루어진다.

4. ‘Enhancement Coding 3번 문제’를 참고하여, 영어 단어를 입력할 경우 딕셔너리 engkor_dict에 등록된 한글 단어를 표시하도록 수정해보자. 만약 딕셔너리가 비어 있으면 “사전이 비어 있습니다.”를 출력하고, 비어 있지 않으면 검색을 하여 단어를 표시한다. 만약 영어 단어가 등록되어 있지 않으면 “단어가 등록되어 있지 않습니다.”를 출력하고, 딕셔너리가 비어 있거나 비어 있지 않더라도 영어 단어가 없으면 “단어를 추가합니다.”를 출력하고 한글 단어를 추가로 입력받아 딕셔너리에 등록한다. while 문에 의한 무한 반복의 종료는 영어 단어가 빈 문자열로 입력되었으면 종료하고 딕셔너리에 등록된 모든 키와 값들을 출력한다.

영어 단어 : Python

사전이 비어 있습니다.

단어를 추가합니다.


한글 단어 : 파이썬

영어 단어 : string

string 단어가 등록되어 있지 않습니다.


단어를 추가합니다.

```
한글 단어 : 문자열
영어 단어 : iteration
iteration 단어가 등록되어 있지 않습니다.
단어를 추가합니다.
한글 단어 : 반복
영어 단어 : selection
selection 단어가 등록되어 있지 않습니다.
단어를 추가합니다.
한글 단어 : 선택
영어 단어 : string
string : 문자열
영어 단어 : iteration
iteration : 반복
영어 단어 :
{'Python': '파이썬', 'string': '문자열', 'iteration': '반복', 'selection': '선택'}
```

 `len(engkor_dict) > 0`인 경우 딕셔너리가 비어 있지 않은 것이며, 딕셔너리에 영어 단어가 존재하는지 여부는 `eng in engkor_dict`로 판별할 수 있다.

5. 현재 우리나라 화폐는 5만원, 1만원, 5천원, 1천원, 5백원, 1백원, 5십원, 1십원, 5원, 1원의 10종류로 되어 있다. 임의의 금액을 입력받아 10종류의 화폐를 사용하여 화폐의 개수(종류)가 가장 적게 구성되도록 해보자. // 연산자와 % 연산자를 사용하여 계산해보자.

```
금액 : 87846
50000 : 1
10000 : 3
5000 : 1
1000 : 2
500 : 1
100 : 3
50 : 0
10 : 4
5 : 1
1 : 1
총 9 종류 17 개 필요
```

 화폐 단위는 moneys = {1:50000, 2:10000, 3:5000, 4:1000, 5:500, 6:100, 7:50, 8:10, 9:5, 10:1}로 선언한다.

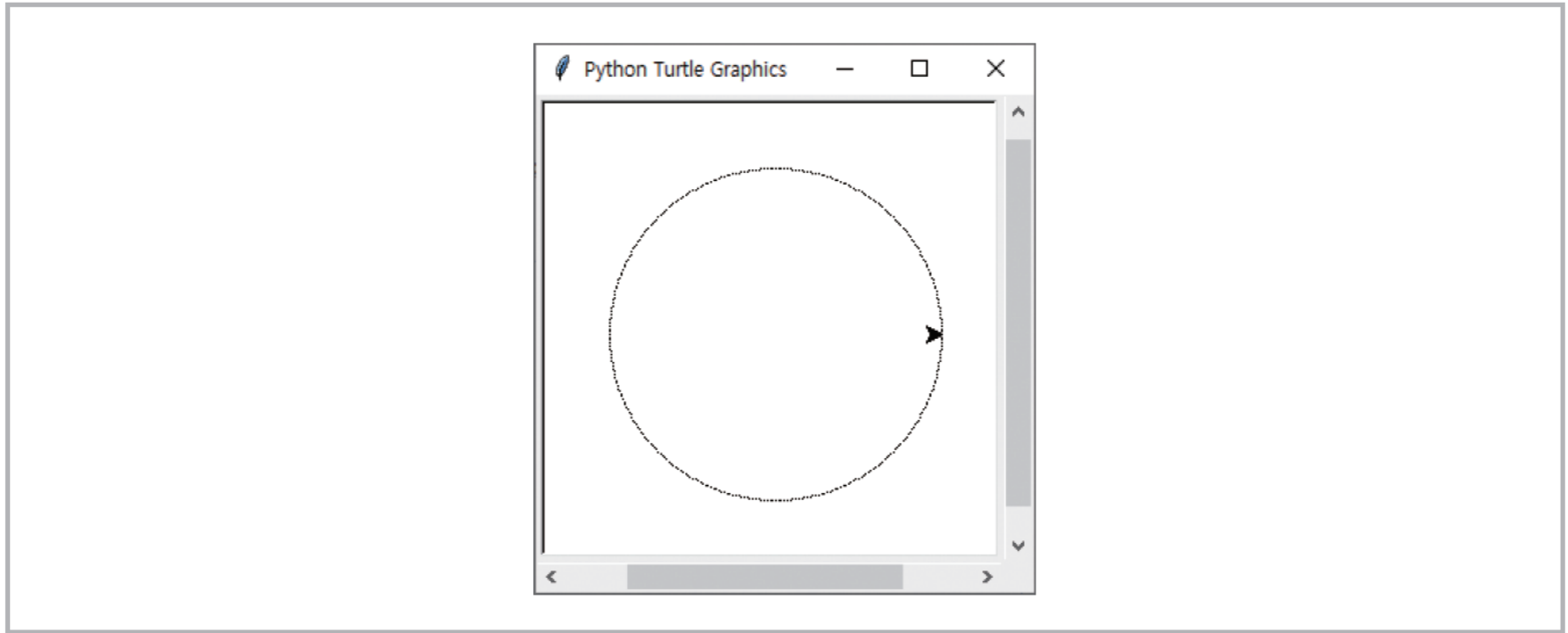
6. 'Enhancement Coding 5번 문제'를 참고하여, math 모듈의 trunc() 함수를 사용하여 같은 결과가 나오도록 프로그램을 변경해보자.


```
금액 : 87846
50000 : 1
10000 : 3
5000 : 1
1000 : 2
500 : 1
100 : 3
50 : 0
10 : 4
5 : 1
1 : 1
총 9 종류 17 개 필요
```



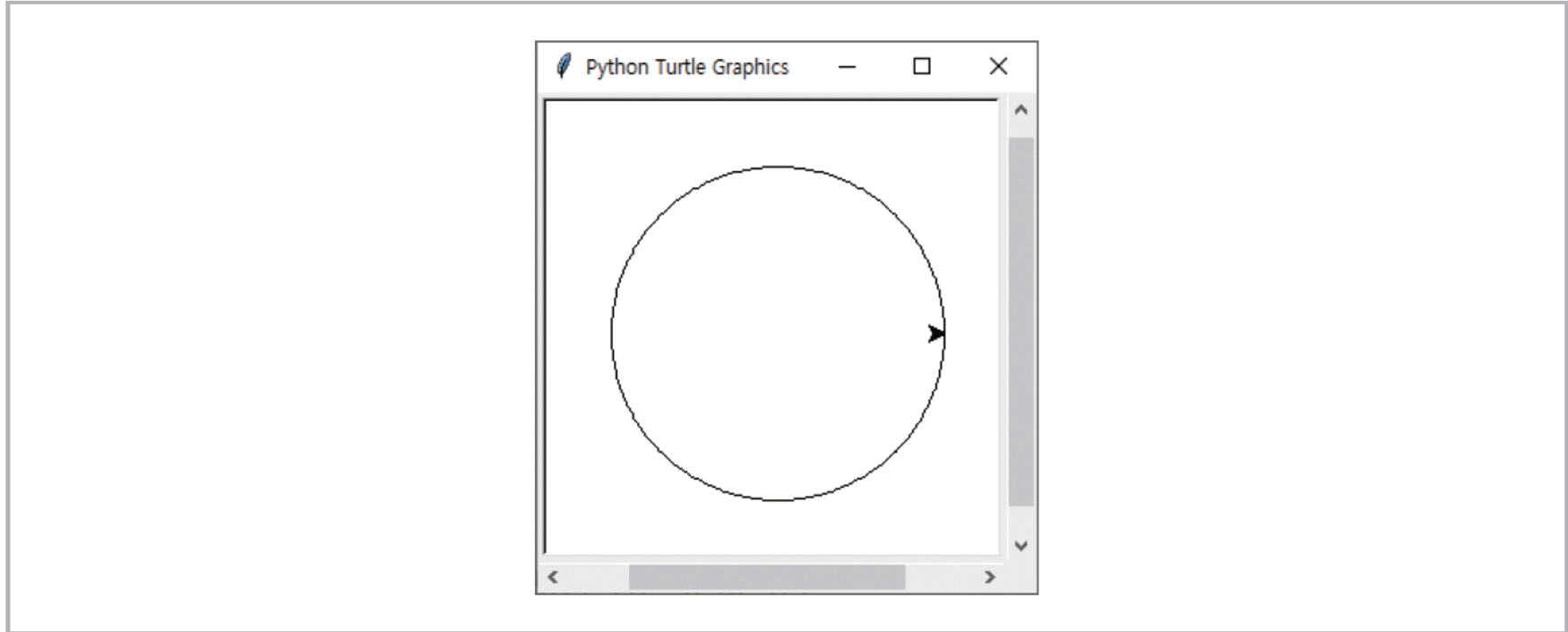
trunc() 함수를 $\text{trunc}(p / \text{moneys}[i])$ 와 $p - \text{trunc}(p / \text{moneys}[i]) * \text{moneys}[i]$ 형태로 사용한다.


7. 삼각함수의 정의($x = r * \cos\theta$, $y = r * \sin\theta$)와 라디안 각도를 이용하여 원을 그려보자.



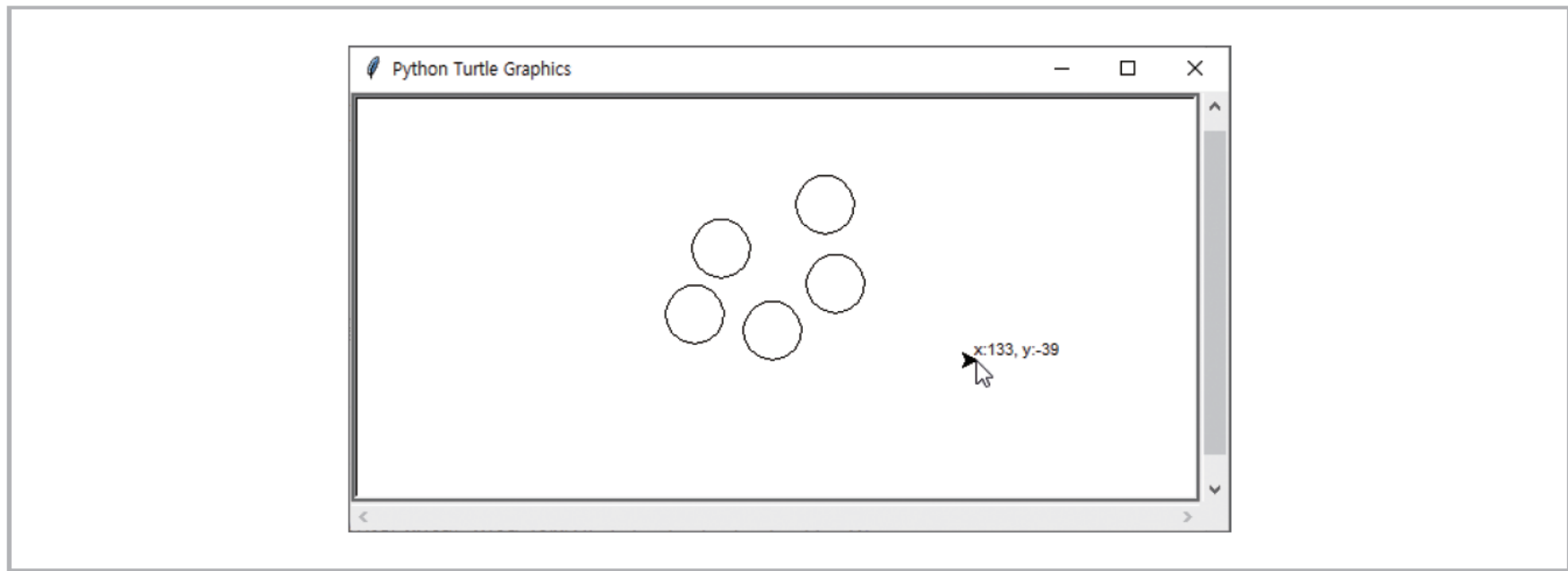
 for a in range(0, 361) 문으로 반복하면서, $rad = \text{math.radians}(a)$ 라디안 값을 구하고 $x = r * \text{math.cos}(rad)$ 와 $y = r * \text{math.sin}(rad)$ 로 x, y 좌표를 구하여 t.dot(2) 함수로 점을 찍어 원을 그린다.


8. 'Enhancement Coding 7번 문제'를 참고하여, 점을 찍는 위 방식은 원의 선이 점으로 되어 있어 완전히 연결된 선의 원이 그려지지 않는다는 문제점이 있다. 선 그리기를 이용하여 선으로 연결된 원을 그리도록 수정해보자.



 for a in range(0, 361) 문으로 반복하면서, $rad = \text{math.radians}(a)$ 라디안 값을 구하고 $x = r * \text{math.cos}(rad)$ 와 $y = r * \text{math.sin}(rad)$ 로 x, y 좌표를 구하여 t.goto(x,y) 함수로 선을 그린다.

9. 'Basic Coding 10번 문제'를 참고하여, 터틀 스크린의 중앙에서 상하좌우 각 100의 범위에서만 원을 그리고, 범위를 벗어난 지점을 클릭하면 해당 좌표를 $x:-110$, $y:-110$ 형식으로 출력하도록 수정해보자.



 마우스 왼쪽 버튼을 클릭한 위치가 중앙에서 x 범위 $-100 \sim 100$ 내이고 y 범위 $-100 \sim 100$ 내이면 원을 그린다. 그리고 범위를 벗어나면 `t.write("x:%d, y:%d"%(x,y))`와 같이 좌표를 출력한다.