# **13차 연습문**항

## **파이썬 기본문항: 모듈**

| 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\_강의명.pdf  ex) DR-11111\_홍길동\_파이썬\_연습문항\_1차시\_환경및기본.pdf  ex) DR-11111\_홍길동\_파이썬\_추가문항\_1차시\_환경및기본.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. |
| --- |

### **1.** **파이썬에서 모듈의 크기에 대한 설명 중 올바른 것은 무엇입니까?**

(1) 모듈은 항상 작은 크기로 제한되어야 한다.

(2) 모듈의 크기는 소스 코드 파일의 크기와 동일하다.

(3) 모듈의 크기는 포함된 기능의 수에 따라 결정된다.

(4) 모듈의 크기는 특정하지 않으며, 기능의 논리적 그룹에 따라 다를 수 있다.

| 4번 |
| --- |

### **2.** **모듈을 사용하는 이유가 아닌 것은 모두 고르시오.**

(1) 코드 재사용성을 높이기 위해

(2) 코드의 가독성을 개선하기 위해

(3) 관련 객체들의 결합성을 증가시키기 위해

(4) 대규모 프로젝트의 관리를 용이하게 하기 위해

(5) 코드의 실행 속도를 빠르게 하기 위해

(6) 특정 기능이나 로직을 독립적으로 테스트하기 위해

| 3, 5번 |
| --- |

### **3.** **다음 중 파이썬의 time 모듈에서 "epoch"에 대한 설명으로 옳은 것은 무엇입니까?**

(1) Epoch는 모든 컴퓨터 시스템에서 2000년 1월 1일 00:00:00을 기준으로 하는 시간의 시작점을 나타낸다.

(2) Epoch는 모든 컴퓨터 시스템에서 동일하게 1970년 1월 1일 00:00:00 UTC를 기준으로 하는 시간의 시작점을 나타낸다.

(3) Epoch는 컴퓨터 시스템에 따라 다를 수 있지만, 대부분의 유닉스 기반 시스템 에서는 1970년 1월 1일 00:00:00 UTC를 기준으로 한다.

(4) Epoch는 시스템의 현재 시간을 기준으로 설정된다.

| 2번 |
| --- |

### **4.** **glob 모듈을 이용하면 유닉스나 윈도우의 커맨드 라인에서 사용하는 와일드카드를 사용할 수 있다. 다음을 와일드카드를 사용해 작성하세요.**

(1) "bc.txt"로 끝나는 3글자 파일명 ex) abc.txt, qbc.txt

(2) 확장자 ".txt"를 가진 모든 파일

(3) 모든 파일

| (1)    (2)    (3) |
| --- |

### **5.** **datetime 모듈을 사용하여 다음을 출력하는 코드를 작성하세요.**

(1) 오늘의 년도, 월, 일

(2) 현재 일자를 형식에 맞추어 출력 (예: 2022년 08월 18일 요일: Thursday)

(3) 현재 시각을 형식에 맞추어 출력 (예: 12시 35분 25초)

| Example Code |
| --- |
| import datetime    # (1)  # Write your code here    # (2)  # Write your code here    # (3)  # Write your code here |

|  |
| --- |

### **6.** **정해진 기준(1970년 1월 1일 00:00:00 UTC)이후 현재까지 몇 초가 흘렀는지 출력하는 코드를 작성하세요.**

| Example Code |
| --- |
| import time    # Write your code here |

|  |
| --- |

### **7.** **time\_utils.py라는 모듈을 만드세요. 이 모듈에는 시간과 관련된 여러 유틸리티 함수를 포함합니다.**

- current\_time(): 현재 시간을 반환합니다.(형식은 “2024-07-01 01:23:07”과 같아야 합니다.)

- sleep\_for(seconds): 주어진 초만큼 일시 정지합니다.

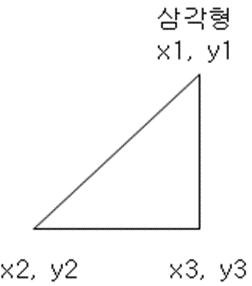
- time\_elapsed(start\_time): 주어진 시작 시간부터 경과된 시간을 반환합니다.

| Example Code (time\_utils.py) |
| --- |
| import time  def current\_time():  lt= time.strftime('%Y년 %m월 %d일 %H:%M:%S')  return lt    def sleep\_for(seconds):  time.sleep(seconds)  def time\_elapsed(start\_time):  now=time.time()  t=now-start\_time  return t  if \_\_name\_\_ == '\_\_main\_\_':  s, t=current\_time()  print(s)  sleep\_for(1)  print('1')  print(time\_elapsed(t)) |

| Example Code (test\_time.py) |
| --- |
| import time\_utils  import time    print("Current time:", time\_utils.current\_time()) # a  print("Sleeping for 2 seconds...")  start\_time = time.time()  time\_utils.sleep\_for(2) # b  print("Current time:", time\_utils.current\_time()) # c  print("Elapsed time after sleeping:", time\_utils.time\_elapsed(start\_time), "seconds") |
| **output**    Current time: 2024-07-23 08:50:07  Sleeping for 2 seconds...  Current time: 2024-07-23 08:50:09  Elapsed time after sleeping: 2.007751226425171 seconds |

|  |
| --- |

### **8.** **여섯 개 좌표를 입력 받는 삼각형 클래스를 만들고 모듈화 시킨다. (모듈명은 Triangle.py) 이 삼각형 모듈을 사용해서 면적을 구하는 프로그램을 작성한다. (프로그램명: TestTriangle.py) 삼각형의 좌표값은 아래 그림과 같이 직각 삼각형으로 입력된다고 가정한다.**



| Example Code(Triangle.py) |
| --- |
| class Triangle:  def \_\_init\_\_(self):  pass  def calcArea(self):  w=((self.x3-self.x2)\*\*2+(self.y3-self.y2)\*\*2)\*\*(1/2)  h=((self.x3-self.x1)\*\*2+(self.y1-self.y1)\*\*2)\*\*(1/2)  print(f'삼각형의 넓이는 {w\*h/2:.2f}입니다.')    def getCoordsInfo(self):  self.x1 = int(input("x1 좌표값을 정수로 입력하세요: "))  self.y1 = int(input("y1 좌표값을 정수로 입력하세요: "))  self.x2 = int(input("x2 좌표값을 정수로 입력하세요: "))  self.y2 = int(input("y2 좌표값을 정수로 입력하세요: "))  self.x3 = int(input("x3 좌표값을 정수로 입력하세요: "))  self.y3 = int(input("y3 좌표값을 정수로 입력하세요: "))  if \_\_name\_\_ == '\_\_main\_\_':  t=Triangle()  t.getCoordsInfo()  t.calcArea() |

| Example Code(Triangle.py) |
| --- |
| import Triangle  t = Triangle.Triangle()  t.getCoordsInfo()  t.calcArea() |

|  |
| --- |