

7장. 모듈(Module)



모듈(Module)

- 모듈(module)

- 변수나 함수 또는 클래스를 모아 놓은 **소스파일**로써, 이를 사용하는 다른 파일에서는 첫 부분에 **[import 모듈이름]**으로 선언한다.

모듈	설명
math	수학 계산과 관련된 모듈
datetime	날짜 및 시간과 관련된 모듈
time	시간과 관련된 모듈
random	난수를 발생시키는 모듈
os	운영 체제(OS) 자원 제어 관련 모듈

math 모듈

◎ math 모듈

〈 모듈 불러오기 〉

```
import math
```

기능	함수의 사용
올림	<code>math.ceil(2.54)</code> # 3
내림	<code>math.floor(2.54)</code> # 2
제곱근	<code>math.sqrt(16)</code> # 4.0
원주율	<code>math.pi</code> # 3.1415

math 모듈

◎ math 모듈

```
import math

# 올림
print(math.ceil(2.54))

# 반올림 - math 모듈이 아님
print(round(2.54))

# 내림(버림)
print(math.floor(2.54))

# 제곱근 - 실수로 반환
print(math.sqrt(2))
print(math.sqrt(25))
```

```
3
3
2
1.4142135623730951
5.0
3.141592653589793
원의 넓이: 50.27
```

math 모듈

◎ math 모듈

```
# 원주율
print(math.pi)

# 원의 넓이 = math.pi * 반지름 * 반지름
radius = 4
area = math.pi * radius * radius
print(f"원의 넓이: {area:.2f}")
```

datetime 모듈

◎ datetime 모듈

```
import datetime
```

```
now = datetime.datetime.today()
```

```
#now = datetime.date.today() #날짜만 출력
```

```
print(now)
```

```
print(f'{now.year}년')
```

```
print(f'{now.month}월')
```

```
print(f'{now.day}일')
```

```
print(f'{now.hour}시')
```

```
print(f'{now.minute}분')
```

```
print(f'{now.second}초')
```

```
# 날짜 표시
```

```
print(f'{now.year}년 {now.month}월 {now.day}일')
```

```
# 시간 표시
```

```
print(f'{now.hour}시 {now.minute}분 {now.second}초')
```

```
2025-05-16 15:41:39.437440
```

```
2025년
```

```
5월
```

```
16일
```

```
15시
```

```
41분
```

```
39초
```

```
2025년 5월 16일
```

```
15시 41분 39초
```

datetime 모듈

◎ 지나온 날짜 계산하기

datetime.date(2025, 3, 1) – 특정 날짜 설정

```
지금까지 몇 일?  
개강 이후 : 27일 지났습니다.
```

```
print("지금까지 몇 일?")  
#특정한 날짜 설정  
the_day = datetime.date(2025, 4, 19)  
# print(the_day)  
today = datetime.date.today()  
  
# 지나온 날 =  
passedtime = today - the_day  
print(f'개강 이후 : {passedtime.days}일 지났습니다.')
```

calendar 모듈

◎ calendar 모듈

- `calendar.prcal(2025)` : 2025년의 달력을 표시
- `calendar.prmonth(2025, 5)` : 2025년 5월의 달력을 표시
- `calendar.weekday(2025, 5, 18)` : 날짜에 해당하는 요일 정보

```
import calendar

calendar.prcal(2025) #전체 출력

calendar.prmonth(2025, 5) #5월 출력

idx = calendar.weekday(2025, 5, 18) # 요일
print(idx)

days = ['월', '화', '수', '목', '금', '토', '일']
print(days[idx])
```


calendar 모듈

◎ calendar 모듈

2025																				
January							February							March						
Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su
		1	2	3	4	5						1	2						1	2
6	7	8	9	10	11	12	3	4	5	6	7	8	9	3	4	5	6	7	8	9
13	14	15	16	17	18	19	10	11	12	13	14	15	16	10	11	12	13	14	15	16
20	21	22	23	24	25	26	17	18	19	20	21	22	23	17	18	19	20	21	22	23
27	28	29	30	31			24	25	26	27	28			24	25	26	27	28	29	30
														31						
April							May							June						
Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su
	1	2	3	4	5	6				1	2	3	4							1
7	8	9	10	11	12	13	5	6	7	8	9	10	11	2	3	4	5	6	7	8
14	15	16	17	18	19	20	12	13	14	15	16	17	18	9	10	11	12	13	14	15
21	22	23	24	25	26	27	19	20	21	22	23	24	25	16	17	18	19	20	21	22
28	29	30					26	27	28	29	30	31		23	24	25	26	27	28	29
														30						

time 모듈

◎ time 모듈

- time.time() 현재 시간을 실수 형태로 돌려주는 함수
- time.localtime() 연도, 월, 일, 시, 분, 초.. 형태
- time.sleep(2) 일정한 시간 간격을 두고 루프를 실행할 수 있다.

```
print(time.time())      # 현재시간을 초로 반환
print(time.localtime()) # 년, 월, 일, 시, 분, 초로 반환
print(time.ctime())     # 날짜와 시간 표기 형태

# 년과 일로 환산(1970. 1.1 이후)
year = round(time.time()/(365*24*60*60))
day = round(time.time()/(24*60*60))
print(year)
print(day)
```

time 모듈

◎ 수행시간 측정하기

```
# 수행 시간 측정
start = time.time() # 시작 시간

# 1초 간격으로 1 ~ 10 출력
for i in range(1, 11):
    print(i)
    time.sleep(1) # 1 - 1초

end = time.time() # 종료 시간
print(f'수행시간 : {end-start:.3f}초')
```

```
1
2
3
4
5
6
7
8
9
10
수행시간 : 10.010초
```

random 모듈

● random 모듈

- random.random() : 0.0에서 1.0사이의 실수 값 중에서 난수 값 발생
- random.randint(1, 10) : 1과 10사이의 정수중에서 난수 값 발생
- random.choice(a) : 리스트에서 무작위로 하나를 선택

```
import random
# 0.0 ~ 1.0 사이의 난수 값
print(random.random())

# 1 ~ 10 사이의 무작위 수
print(random.randint(1, 10))

# 주사위 눈(1 ~ 6)
print(random.randint(1, 6))

# 리스트에서 무작위 선택
fruits = ["딸기", "수박", "참외", "사과"]
print(random.choice(fruits))
```

```
0.25227595799752256
9
1
참외
```

random 모듈

- random 모듈

```
# 동전 던지기
coin = random.randint(0, 1)
print(coin)
if coin == 0:
    print("앞면")
else:
    print("뒷면")

# 주사위 10번 던지기
for i in range(10):
    dice = random.randint(1, 6)
    print(dice)
```

숫자 추측 게임

- 숫자를 추측해서 맞히는 게임

- 게임 방법

- 게임이 시작되면 컴퓨터가 난수(1 ~ 30 사이의 수)를 생성한다.
 - 사용자의 추측값이 정답과 같으면 '정답!', 추측값이 정답보다 크면 "너무 커요!", 추측값이 정답보다 작으면 '너무 작아요!' 출력
 - 사용자의 추측값이 1 ~ 30의 범위를 벗어나면 "범위를 초과했어요. 다시 입력하세요" 출력함

```
맞혀보세요(입력: 1 ~ 30): 40  
범위를 초과했어요. 다시 입력하세요  
맞혀보세요(입력: 1 ~ 30): 20  
너무 작아요  
맞혀보세요(입력: 1 ~ 30): 26  
너무 작아요  
맞혀보세요(입력: 1 ~ 30): 28  
정답!
```

숫자 추측 게임

- 숫자를 추측해서 맞추는 게임

```
com = random.randint(1, 30) #컴퓨터의 난수
# print(com)

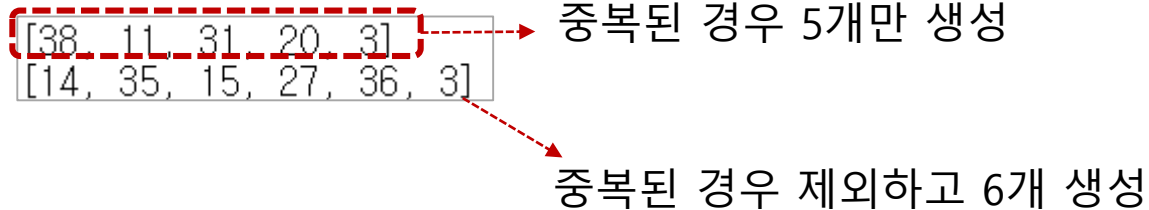
while True:
    x = input("맞혀 보세요(입력: 1 ~ 30): ")
    guess = int(x) # 사용자가 추측한 수

    if guess < 0 or guess > 30:
        print("범위를 초과했어요. 다시 입력하세요")
    elif guess == com:
        print("정답!")
        break
    elif guess > com:
        print("너무 커요")
    else:
        print("너무 작아요")
```

로또(lotto) 복권

- 로또(lotto) 복권 추첨 프로그램

로또 번호를 중복되지 않도록 생성하는 프로그램 만들기



로또(lotto) 복권

- 로또(lotto) 복권 추첨 프로그램

```
import random

lotto = []

for i in range(6):
    n = random.randint(1, 45)
    if n not in lotto:
        lotto.append(n)
    ...
    중복될 경우 5개만 저장됨
    [44, 7, 10, 8, 31]
    ...
print(lotto)
```

```
while len(lotto) < 6:
    n = random.randint(1, 45)
    if n not in lotto:
        lotto.append(n)
# 6개가 될때까지 저장됨
print(lotto)
```

영어 타자 연습 게임

● 게임 방법

- 게임이 시작되면 영어 단어가 화면에 표시된다.
- 사용자는 최대한 빠르고 정확하게 입력해야 한다.
- 바르게 입력했으면 다음 문제로 넘어가고 "통과"를 출력한다.
- 오타가 있으면 같은 단어가 한 번 더 나온다.
- 타자 게임 시간을 측정한다.

문제 3
earth
earth
통과!

문제 4
strawberry
str
오타! 다시 도전!

문제 4
potato
potato
통과!

문제 8
flower
flower
통과!

문제 9
sky
sky
통과!

문제 10
earth
earth
통과!
타자 시간: 28.80초

타자 연습 게임

- 영어 타자 연습 프로그램

```
import random
import time

word = ["sky", "earth", "sun", "moon", "flower",
        "tree", "mountain", "strawberry", "garlic", "potato"]
n = 1 #문제 번호

print("[타자 게임] 준비되면 엔터!")
input()

start = time.time() #시작 시간
```

타자 연습 게임

- 영어 타자 연습 프로그램

```
while n < 11:
    print("\n문제", n)
    question = random.choice(word)
    print(question)

    you = input() # 사용자 입력
    if question == you:
        print("통과!")
        n += 1 #다음 문제
    else:
        print("오타! 다시 도전!")

end = time.time() #종료 시간
et = end - start
print(f"타자 시간: {et:.2f}초")
```

os 모듈(Module)

◎ os 모듈

- 환경변수나 디렉터리, 파일 등의 OS 자원을 제어할 수 있게 해주는 모듈이다.

c:\pyworks 디렉터리

```
2025-04-24 오전 07:48 <DIR> .
2025-04-24 오전 08:03 <DIR> basic
2025-04-24 오전 08:03 <DIR> choice
                0개 파일                0 바이트
                3개 디렉터리 30,873,776,128 바이트 남음
```

```
import os

os.chdir('c:/pyworks') # pyworks 디렉터리로 이동
dir = os.popen('dir')   # dir 명령 실행
print(dir.read())       # dir 결과 출력
```

모듈의 사용과 패키지

◆ 모듈 사용방법

- **import** 모듈 이름
- **from** 패키지 이름 **import** 파일(모듈)이름
- **from** 패키지이름.파일이름 **import** 함수, 클래스

```
import datetime

# 현재 날짜와 시간 출력
now = datetime.datetime.now()
print(now)

# 특정한 날짜 설정
the_day = datetime.date(2025, 6, 3)
print(the_day)
```

```
from datetime import datetime, date

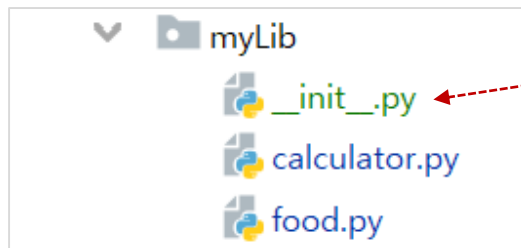
# 현재 날짜와 시간
now = datetime.now()
print(now)

# 오늘 날짜 출력
today = date.today()
print(today)
```

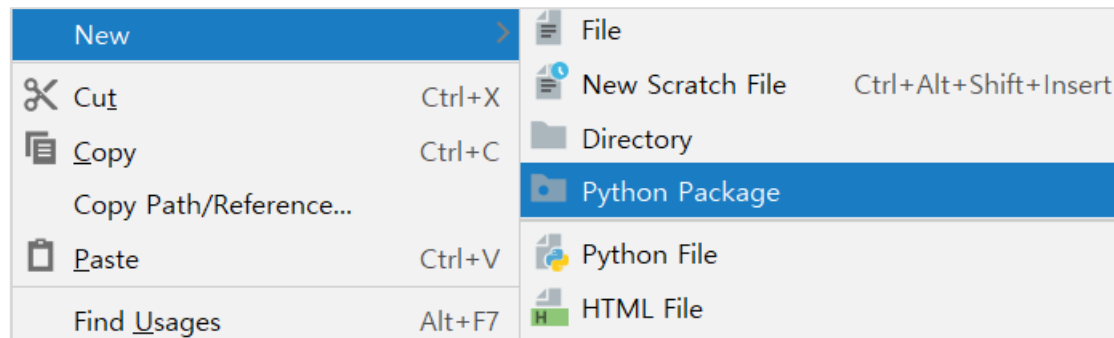
패키지(package)

◆ 패키지(package)

- 모듈(파일)을 모아 놓은 디렉터리를 **패키지**라 한다.
- 파이썬의 패키지로 인식되려면 **`__init__.py`**을 포함해야 한다.



`__init__.py`엔 아무 내용이 없지만, 삭제하면 작동하지 않음



모듈(Module) 만들기

◎ 모듈 만들기

calculator.py

```
def add(x, y):  
    return x + y  
  
def sub(x, y):  
    return x - y  
  
def mul(x, y):  
    return x * y  
  
def div(x, y):  
    return x / y
```

food.py

```
name = "장금이"  
  
def cook():  
    print('요리하다')  
  
def eat():  
    print('먹다')
```