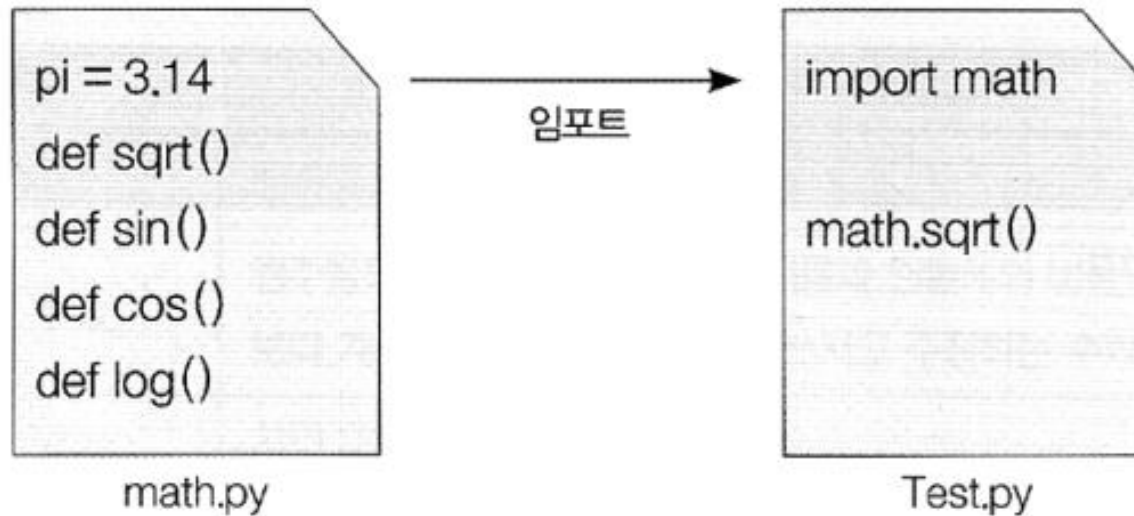

표준모듈

수학

❖ 임포트(import)

- 다른 파일에 정의된 변수, 함수, 객체 등을 사용하기 전에 이를 알리는 것
- 표준 모듈
 - 파이썬에서 제공하는 모듈

```
import 모듈 [as alias]  
from 모듈 import 함수명
```



수학

❖ 임포트(import)

```
import math
```

```
print(math.sqrt(2))
```

```
1.4142135623730951
```

```
from math import sqrt
```

```
print(sqrt(2))
```

```
1.4142135623730951
```

```
import math as m
```

```
print(m.sqrt(2))
```

```
1.4142135623730951
```

```
from math import sqrt as sq
```

```
print(sq(2))
```

```
1.4142135623730951
```

수학

❖ math 모듈

○ 상수

- `pi` : 원주율 상수
- `tau` : 원주율의 2배되는 상수
- `e` : 자연 대수 상수
- `inf` : 무한대 값
- `nan` : 숫자가 아닌 값을 의미

수학

❖ math 모듈

○ 함수

- `sqrt(x)`
- `pow(x,y)`
- `hypot(x,y)`
- `factorial(x)`
- `sin(x), cos(x), tan(x)`
- `degrees(x)`
- `radians(x)`
- `ceil(x)`
- `floor(x)`
- `fabs(x)`
- `trunc(x)`
- `log(x, base)`
- `log10(x)`
- `gcd(a, b)`

수학

❖ math 모듈

```
import math

print(math.sin(math.radians(45)))
print(math.sqrt(2))
print(math.factorial(5))
```

0.7071067811865475

1.4142135623730951

120

❖ 통계 모듈, statistics

○ 함수

- `mean()` : 평균
- `harmonic_mean()` : 조화평균
- `median()` : 중앙값, 짝수인 경우 보간값 계산
- `median_low()` : 중앙값을 구함, 집합 내의 낮은 값 선택
- `median_high()` : 중앙값을 구함, 집합 내의 높은 값 선택
- `mdeian_grouped()` : 그룹 연속 중앙값
- `mode()`: 최빈값
- `pstdev()` : 모표준편차
- `stddev()` : 표준편차
- `variance()` : 분산

수학

❖ 통계 모듈, statistics

```
import statistics as st

score = [30, 40, 60, 70, 80, 90]
print(st.mean(score))
print(st.harmonic_mean(score))
print(st.median(score))
print(st.median_low(score))
print(st.median_high(score))
```

61.666666666666664

53.14586994727592

65.0

60

70

시간

❖ 시간조사, time 모듈

- 1970년 1월 1일 자정을 기준으로 경과한 시간을 초 단위로 표현
--> 에폭(Epoch) 시간 또는 유닉스 시간

```
import time  
  
print(time.time())  
  
1579487898.483571
```

시간

❖ 시간조사, time 모듈

```
import time
```

```
t = time.time()  
print(time.ctime(t))
```

Mon Jan 20 11:38:44 2020

```
import time
```

```
t = time.time()  
print(time.localtime(t))
```

```
time.struct_time(tm_year=2020, tm_mon=1, tm_mday=20, tm_hour=11, tm_min=38,  
tm_sec=56, tm_wday=0, tm_yday=20, tm_isdst=0)
```

시간

❖ 시간조사, time 모듈

```
import time
```

```
now = time.localtime()  
print("%d년 %d월 %d일" % (now.tm_year, now.tm_mon, now.tm_mday))  
print("%d:%d:%d" % (now.tm_hour, now.tm_min, now.tm_sec))
```

2020년 1월 20일
11:40:44

```
import datetime
```

```
now = datetime.datetime.now()  
print("%d년 %d월 %d일" % (now.year, now.month, now.day))  
print("%d:%d:%d" % (now.hour, now.minute, now.second))
```

2020년 1월 20일
11:42:2

시간

❖ 실행 시간 측정

```
import time

start = time.time()
for a in range(1000):
    print(a)
end = time.time()

print(end - start)
```

```
0
1
:
999
0.1033775806427002
```

시간

❖ 실행 멈춤

```
import time

print("안녕하세요")
time.sleep(1)
print("밤에 성시경이 두 명 있으면 뭘까요?")
time.sleep(5)
print('야간투시경입니다.')
```

안녕하세요
밤에 성시경이 두 명 있으면 뭘까요?
야간투시경입니다.

시간

❖ 달력

```
import calendar as cal

print(cal.calendar(2018))
print(cal.month(2019, 1))
```

2018

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

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

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

:

시간

❖ 달력

```
import calendar as cal

dates = ["월", "화", "수", "목", "금", "토", "일"]

day = cal.weekday(2020, 8, 15)
print("광복절은 %s요일입니다." % dates[day])
```

광복절은 토요일입니다.

난수

❖ random 모듈

- .random()
 - 0 ~ 1 사이의 난수 리턴(1은 미포함)

```
import random

for i in range(5):
    print(random.random())
```

```
0.9085194756407313
0.5157698060289099
0.6511156516629886
0.6844494104248139
0.07817243576575794
```


난수

❖ random 모듈

- `.randint(begin, end)`
 - `begin ~ end` 사이의 정수 난수를 리턴 (`end`도 포함)
- `.randrange(begin, end)`
 - `begin ~ end` 사이의 정수 난수를 리턴 (`end`는 포함되지 않음)
- `.uniform(begin, end)`
 - `begin ~ end` 사이의 실수 난수를 리턴 (`end` 미포함)

```
import random

for i in range(5):
    print(random.randint(1, 10))
```

```
3
3
10
4
5
```

난수

❖ random 모듈

```
import random

for i in range(5):
    print(random.randrange(1, 10))
```

2
3
4
4
6

```
import random

for i in range(5):
    print(random.uniform(1, 10))
```

8.810192699010965
3.4528419150136185
9.062162080248966
9.674934682673284
9.667378503574168

난수

❖ random 모듈

- .choice(시퀀스)
 - 시퀀스에서 랜덤하게 요소 선택하여 리턴

```
import random

food = ["짜장면", "짬뽕", "탕수육", "군만두"]
print(random.choice(food))
```

탕수육

```
i = random.randrange(len(food))
print( food[i])
```

짜장면

난수

❖ random 모듈

- .shuffle(시퀀스)
 - 시퀀스의 내용을 랜덤하게 섞음

```
import random

food = ["짜장면", "짬뽕", "탕수육", "군만두"]
print(food)
random.shuffle(food)
print(food)
```

```
['짜장면', '짬뽕', '탕수육', '군만두']
['군만두', '탕수육', '짜장면', '짬뽕']
```

난수

❖ random 모듈

- .sample(시퀀스, count)
 - 시퀀스에서 랜덤하게 count개의 요소 리턴

```
import random
```

```
food = ["짜장면", "짬뽕", "탕수육", "군만두"]  
print(random.sample(food, 2))
```

```
['짬뽕', '짜장면']
```

```
import random
```

```
nums = random.sample(range(1, 46), 6)  
nums.sort()  
print(nums)
```

```
[5, 6, 17, 22, 28, 36]
```

sys 모듈

❖ 시스템 정보

```
import sys

print("버전: ", sys.version)
print("플랫폼: ", sys.platform)
print("바이트 순서: ", sys.byteorder)
print("모듈 경로: ", sys.path)
sys.exit(0)
```

```
버전:  3.7.3 (default, Mar 27 2019, 22:11:17)
[GCC 7.3.0]
플랫폼:  linux
바이트 순서:  little
모듈 경로:  [ ... ]
```

sys 모듈

❖ 명령형 인수

```
import sys
```

```
print(sys.argv)  
# [파일경로, 인자1, 인자2, ...]
```

버전: 3.7.3 (default, Mar 27 2019, 22:11:17)

[GCC 7.3.0]

플랫폼: linux

바이트 순서: little

모듈 경로: [...]