

---

# 모듈의 이해

9주차\_01

한 동 대 학 교  
김 경 미 교수

# 모듈(Modules)

---

- **표준라이브러리(standard library)의 일부분**
  - 파이썬 프로그램에서 사용되기 위한 명령문을 포함하고 있는 파일
  - 프로그래밍에 앞서 구현하려는 기능이 파이썬 라이브러리 모듈에 있는지 여부 확인 필요
- **Turtle, random 모듈을 이미 사용해 봄**
- **random, datetime, math, string, turtle, tkinter, file 등 200여개 모듈 존재**

# 내장형 모듈

```
# eval()
```

```
def dump(expression) :  
    result = eval(expression)  
    print(expression, "=>", result, type(result))
```

```
dump("1")  
dump("1.0")  
dump("'string'")  
dump("1.0 + 2.0")  
dump("'*' * 10")  
dump("len('world')")
```

```
>>>  
===== RESTART: E:/1_Works/2017Work/KI  
=====  
1 => 1 <class 'int'>  
1.0 => 1.0 <class 'float'>  
'string' => string <class 'str'>  
1.0 + 2.0 => 3.0 <class 'float'>  
'*' * 10 => ********** <class 'str'>  
len('world') => 5 <class 'int'>  
>>> |
```

# 모듈 String

```
import string

text = "All that I need you"
print("capword", "=>", string.capwords(text))

print("upper", "=>", text.upper())
print("lower", "=>", text.lower())
print("split", "=>", text.split(" "))
print("replace", "=>", text.replace("you", "him"))
print("find", "=>", text.find("All"))
print("count", "=>", text.count("e"))
```

```
>>>
===== RESTART: E:/1_Works/2017Work/KMooC강의
=====
capword => All That I Need You
upper => ALL THAT I NEED YOU
lower => all that i need you
split => ['All', 'that', 'I', 'need', 'you']
replace => All that I need him
find => 0
count => 2
>>>
```

# 연습문제 1

---

- **스트링 변환하기**

- 사용자에게 30글자 이상의 영어문장을 입력 받는다
- 알파벳이 몇 개로 구성되었는지 출력한다
- 대문자, 소문자 각각 몇 개로 구성되었는지 출력한다

# 연습문제 1 코드

---

```
input_str = input("30글자이상 영어문장 입력: ")
alpha = 0
upper = 0
lower = 0

for letter in input_str:
    alpha = alpha + letter.isalpha()
    upper = upper + letter.isupper()
    lower = lower + letter.islower()

print("알파벳 개수: ", alpha, ", 대문자 개수: ", upper, ", 소문자 개수: ", lower)
```

```
>>>
===== RESTART: E:/1_Works/2017Work/KMooC강의/Exercise Code/8주차
=====
30글자이상 영어문장 입력: As we have already used the term module
알파벳 개수: 32 , 대문자 개수: 1 , 소문자 개수: 31
>>>
```

# math Module(1)

<http://docs.python.org/3/library/math.html>

Methods with Description	
<code>math.ceil(x)</code> ( $N \geq x$ )를 만족하는 가장 큰 정수 $N$ 을 반환	<code>math.fsum(iterable)</code> 합계
<code>math.copysign(x, y)</code> $y$ 의 부호만 $x$ 에 복사해서 반환	<code>math.isfinite(x)</code> 유한수인 경우 참
<code>math.fabs(x)</code> 절대값	<code>math.isinf(x)</code> 무한수인 경우 참
<code>math.factorial(x)</code>	<code>math.fmod(x, y)</code> 실수의 mod 함수
<code>math.floor(x)</code> ( $N \leq x$ )를 만족하는 가장 큰 정수 $N$ 을 반환	<code>math.modf(x)</code> 입력받은 $x$ 를 순수 소수부분과 정수 부분으로 분리해 tuple로 반환, 분리된 부분은 모두 부호가 할당

# math Module(2)

<http://docs.python.org/3/library/math.html>

---

- 승, 로그 함수

Methods with Description	
<code>math.exp(x)</code>	<code>math.log2(x)</code> Return the base-2 logarithm of x
<code>math.log(x[, base])</code>	<code>math.log10(x)</code> Return the base-10 logarithm of x
<code>math.pow(x, y)</code>	<code>math.sqrt(x)</code>



# math Module(3)

<http://docs.python.org/3/library/math.html>

---

- 삼각함수, 각도 변환

Methods with Description	
<code>math.asin(x)</code>	<code>math.sin(x)</code>
<code>math.atan(x)</code>	<code>math.tan(x)</code>
<code>math.atan2(y, x)</code>	<code>math.cos(x)</code>
<code>math.degrees(x)</code>	<code>math.radians(x)</code>

# math Module(4)

<http://docs.python.org/3/library/math.html>

---

- **Hyperbolic functions(쌍곡선 함수) 과 constants**

Methods with Description	
<code>math.acosh(x)</code>	<code>math.sinh(x)</code>
<code>math.asinh(x)</code>	<code>math.tanh(x)</code>
<code>math.atanh(x)</code>	<code>math.pi</code> constant $\pi = 3.141592\dots$
<code>math.cosh(x)</code>	<code>math.e</code> constant $e = 2.718281\dots$

# math Module 예제

---

```
>>> import math

>>> math.fmod(5.0, 2.0)      # 실수형의 나머지 연산
1.0

>>> math.factorial(5)      # 정수의 factorial
120

>>> math.log(100,10)      # log값
2.0

>>> math.pow(2,3)
8.0

>>> math.sqrt(16)
4.0

>>> math.e                  # 자연로그 값
2.718281828459045
```

# cmath module

---

- Math module 동일한 기능을 복소수 대상으로 지원
- 참조
  - <https://docs.python.org/3/library/cmath.html#module-cmath>

# 연습문제 2

---

- **Math module을 활용하여**
  - 이차방정식의 근의 공식을 만든다
  - 이차방정식의 a, b, c값을 입력 받아 실근을 출력한다
  - a, b, c값에 따라 함수를 선택하여 근을 계산한다
  - 근의 공식 =  $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

# 연습문제 2 코드, 실근만 처리

```
import math
```

```
def roots_formula(a, b, c):
```

```
    d = b**2 - 4*a*c
```

```
# 판별식 계산
```

```
    if d > 0:
```

```
        root1 = (-b + math.sqrt(d)) / (2*a)
```

```
        root2 = (-b - math.sqrt(d)) / (2*a)
```

```
        return root1, root2
```

```
    elif d == 0:
```

```
        root1 = (-b / (2*a))
```

```
        return root1
```

```
    else:
```

```
        return False
```

```
#허근일 경우 false로 처리
```

```
r = roots_formula(1,5,4)
```

```
print( r )
```

```
>>>
```

```
===== RESTART :
```

```
=====
```

```
(-1.0, -4.0)
```

```
>>> |
```

# 연습문제 2 코드, 허근 포함

```
import math, cmath
```

```
def deter(a, b, c):
```

```
    return math.pow(b, 2) - 4*a*c
```

```
def roots_formula(a, b, c):
```

```
    if deter(a,b,c) >= 0:
```

```
        root01 = (-b + math.sqrt(deter(a,b,c)))/(2*a)
```

```
        root02 = (-b - math.sqrt(deter(a,b,c)))/(2*a)
```

```
    else:
```

```
        root01_real = -b/(2*a)
```

```
        root01_imag = (math.sqrt(math.fabs(deter(a,b,c))))/(2*a)
```

```
        root02_real = -b/(2*a)
```

```
        root02_imag = (math.sqrt(math.fabs(deter(a,b,c))))/(2*a)
```

```
        root01 = root01_real + root01_imag * 1j
```

```
        root02 = root02_real - root02_imag * 1j
```

```
    return [root01, root02]
```

```
print(roots_formula(1,3,5))
```

```
>>>
```

```
===== RESTART: E:/1_Works/2017Work/KMooC강의/Exerci
```

```
=====
```

```
[(-1.5+1.6583123951777j), (-1.5-1.6583123951777j)]
```

```
>>> |
```

# 연습문제 3

---

- **Math module을 활용하여**
  - 사용자에게 각도(degree)를 입력 받는다
  - 입력 값을 radian 값으로 바꾼 후
  - 다음의 삼각함수 값을 출력 하시오
    - $\sin()$ ,  $\cos()$ ,  $\tan()$



# 연습문제 3 코드

---

```
import math

angle = float(input("각도[degree]: "))
rad = math.radians(angle)           # radian 값으로 바꾸기

result_sin = math.sin(rad)
result_cos = math.cos(rad)
result_tan = math.tan(rad)

print("degree:", angle, ", radian:", rad)
print("sin:", result_sin, ", cos:", result_cos, ", tan:", result_tan)
```

```
>>>
===== RESTART: E:/1_Works/2017Work/KMooC강의/Exercise Code/8주차_합
=====
각도[degree]: 90
degree: 90.0 , radian: 1.5707963267948966
sin: 1.0 , cos: 6.123233995736766e-17 , tan: 1.633123935319537e+16
>>>
```

# Datetime module(1)

<https://docs.python.org/3/library/datetime.html?highlight=datetime#module-datetime>

---

- 날짜와 시간 관련 기능을 제공
- **class datetime.date**
  - Gregorian calendar 기준
  - Year, month, day
- **class datetime.time**
  - Hour, minute, second, microsecond
- **class datetime.datetime**
  - Class date, time을 합친 것

# 예제, date object

---

- 날짜를 불러오고, 날짜 간의 마이너스가 가능하다

```
import time
from datetime import date
```

```
today = date.today()
print(today)
```

```
Bday = date(today.year+1, 2, 24)
print(Bday)
```

```
due = abs(Bday - today)
print(due)
```

```
>>>
===== RESTART: E:/1_Works/2017Work/
=====
2017-06-30
2018-02-24
239 days, 0:00:00
>>> |
```

# 예제, datetime object

---

**.combine**

**.now**

**.utcnow, 협정 세계시(Coordinated Universal Time, UTC)**

```
from datetime import date, time, datetime
```

```
d=date(1990, 7,5)
```

```
t=time(12,30)
```

```
print(datetime.combine(d, t))
```

```
print(datetime.now())
```

```
print(datetime.utcnow())
```

```
>>>
```

```
===== RESTART: E:/1_Works/2017\
```

```
====
```

```
1990-07-05 12:30:00
```

```
2017-06-30 19:02:00.045292
```

```
2017-06-30 10:02:00.059293
```

```
>>> |
```

## 연습문제 4

---

- **부모님 생일이 며칠 남았는지 계산하려고 한다**
  - 사용자에게 생일을 입력 받는다
  - 오늘부터 생일까지 며칠 남았는지 계산하여 출력한다

# 연습문제 4 코드

```
from datetime import date

birthday = input("생일을 입력하세요 [ex)3월15일 = 3 15] : ")
birthday = birthday.split()
today = date.today()
birthday = date(today.year, int(birthday[0]), int(birthday[1]))

due = birthday - today
if due.days < 0:
    next_birthday = date(today.year + 1, birthday.month, birthday.day)
    due = next_birthday - today

print("생일까지 남은 날짜는: ", due.days)
```

```
>>>
===== RESTART: E:/1_Works/2017Work/KMooC강의
====
생일을 입력하세요 [ex)3월15일 = 3 15] : 5 25
생일까지 남은 날짜는: 329
>>> |
```

# 요약

---

- 모듈이 무엇인지 이해한다
- `string` 모듈 활용한다
- `math` 모듈 활용한다
- `datetime` 모듈 활용한다

---

# 감사합니다

9주차\_01 모듈의 이해