

Sage는 python 프로그래밍 언어로 작성된 수학전용 S/W이다.

온라인으로 사용하기 위해서는 <https://cloud.sagemath.com> 접속하여 사용한다(크롬사용).

1. 변수 대입, 비교판단

```
sage: a = 5
sage: a
5
sage: 2 == 2
True
sage: 2 == 3
False
sage: 2 < 3
True
sage: a == 5
True
G
```

2. 수학 계산

```
sage: 2**3 # means exponent
8
sage: 2 ^ 3 # ^ is a synonym for
8
sage: 10 % 3 # for integer arguments, % means mod
1
sage: 10 / 4
5/2
sage: 10 // 4 # for integer arguments, // returns the integer quotient
2
sage: sqrt(3.4) # sqrt returns the square root
1.84390889145858
sage: exp(2)
e^2
```

3. 형선언

```
sage: x = 1 # x is an integer
sage: type(x)
```

```

<type `sage.rings.integer.Integer'>
sage: x = 1/2 # now x is a rational number
sage: type(x)
<type `sage.rings.rational.Rational'>
sage: x = `hello ' # now x is a string
sage: type(x)
<type `str'>

```

4. 여러식 표기(: 세미콜론) 및 줄연결방법(\ 백슬래시)

```

sage : a = 5; b = 3; c = a + b;
sage : 2 + \
      ....: 3

```

5. python 프로그래밍

1) 함수 정의

```

sage: def even(n):
      return n%2 == 0 # blocks of code are indented
sage: even (2)
True
sage: even(3)
False

```

2) 반복 정의(iterating)

```

sage : for i in range(3): # is like for(i=0; i<3; i++)
: : : : : print i
0
1
2

```

```

sage: for i in range(2,5): # is like for(i=2;i<5;i++)
: : : : : print i
2
3
4

```

```

sage : for i in range(1,6,2): # is like for(i=1;i<6;i+=2)
: : : : : print i
1
3
5

```

3) 조건문(if): 조건문이 참일 때, 아래의 실행문을 실행하고, 거짓일 때 무시

3 if 조건문:
실행문

주어진 x,y,z값 중에서 최소값을 구하려면 다음과 같이 표현한다.

```
def least(x,y,z)
if x<y and x<z:
    print x, ' is least'
elif y<z:
    print y, ' is least'
else:
    print z, ' is least'
```

```
least(5,3,9)
```

```
3 is least
```

6. 데이터 구조(list)

sage의 기본 데이터구조는 리스트형식이다. 0 첨자부터 시작되는 배열구조이다.

```
sage : v = range(2,10)
```

```
[2, 3, 4, 5, 6, 7, 8, 9]
```

```
sage: v[0] # 2
```

```
sage: v[3] # 5
```

```
sage : len(v)
```

```
8
```

```
sage : v.append(10)
```

```
sage : del v[1]
```

```
sage : v
```

```
[2, 4, 5, 6, 7, 8, 9, 10]
```

```
sage : v.insert(1,3)
```

```
[2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
sage : v.remove(9)
```

```
[2, 3, 4, 5, 6, 7, 8, 10]
```

```
sage : v.reverse()
```

```
[10, 8, 7, 6, 5, 4, 3, 2]
```

```
sage : v.sort()
```

```
[2, 3, 4, 5, 6, 7, 8, 10]
```

7. 기본 링(Ring)

전주대 토목환경공학과 지반연구실

링 표현은 덧셈 과 곱셈이 작동되는 수학적인 구조이다.

ZZ: 정수(integer) QQ: 유리수(rational number) RR: 실수(real) CC: 복소수(complex)

```
sage : QQ
Rational Field
```

8. 이론수 함수

```
sage : a = 15; b = 35; c = 28; n = 19
sage : gcd(a, b) # return the greatest common divisor of the integers
a ar  $\left\{ 3x \mid x \in \{0, 1, \dots, 9\} \text{ and } x \text{ is odd} \right\}$ 
5
sage : mod(a,n) # return the value of a modulo n
15
sage : factor(a) # return the prime factors of a
3 * 5
sage : divisors(c) # return the divisors of c
[1, 2, 4, 7, 14, 28]
sage : prime divisors(c) # return only the prime divisors of c
[2, 7]
```

9. 행렬(matrix)

```
sage : A = matrix([[1,2,3], [3,2,1], [1,1,1]])
sage : w = vector([1,1,-4])
sage : w * A
(0, 0, 0)
sage : A * w
(-9, 1, -2)
```

10. 집합(set)

다음과 같은 집합표현을 sage로표시하면

```
sage: [3*x for x in range(0,10) if x%2 == 1]
[3, 9, 15, 21, 27]
```

11. 선형대수학

QQ(유리수)형태로 3x3 행렬M를 생성하고자 하려면, 우선 행렬공간을 다음과 같이 만든다.

```
sage:M = MatrixSpace(QQ,3)
```

5행렬M은 다음과 같이 표시한다.

```
sage: A=M([2, 2, 3, 4, 5, 6, 7, 8, 9])
sage: A
[2  2  3]
[4  5  6]
[7  8  9]
```

역행렬, 트레이스, 전치행렬, ref은 다음과 같이 구한다.

```
sage: B=A.inverse(); B
[  1  -2   1]
[ -2   1   0]
[  1  2/3 -2/3]
sage: A.trace()
16
sage:A.determinant()
-3
sage:A.transpose()
[2  4  7]
[2  5  8]
[3  6  9]
sage:A.echelon_form()
[1  0  0]
[0  1  0]
[0  0  1]
```

선형연립방정식 $Ax=b$ 를 풀기위한 명령어는 다음과 같다.

```
sage: A=matrix(QQ,3,[1,2,4,2,3,1,0,1,2])
[1  2  4]
[2  3  1]
[0  1  2]
sage: b=matrix(QQ,3,2,[1,7,5,2,1,3]) # 두 경우에 대한 열벡터를 표시
[1  7]
[5  2]
[1  3]
sage: x=A.solve_right(b) # 해구하기 방법1
sage: x=A\b               # 해구하기 방법2
sage: x=A._backslash_(b) # 해구하기 방법3
[ -1   1]
[13/5 -3/5]
```

$$\begin{bmatrix} -4/5 & 9/5 \end{bmatrix}$$

두 경우에 대한 해를 동시에 구함

6

```
def _backslash_(self,B):
    return self.solve_right(B)
```

또한, 다음과 같은 방법으로도 해를 구할 수 있다.

```
sage: A=matrix(QQ,3,[1,2,4,2,3,1,0,1,2])
```

$$\begin{bmatrix} 1 & 2 & 4 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 3 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 & 2 \end{bmatrix}$$

```
sage: b=vector(QQ,[1,5,1])
```

$$\begin{pmatrix} 1, & 5, & 1 \end{pmatrix}$$

```
sage: x=A.augment(b).rref()
```

$$\begin{bmatrix} 1 & 0 & 0 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 & 0 & 13/5 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 1 & -4/5 \end{bmatrix}$$

행렬에 대한 고유값과 특성다항식을 구하는 명령어는 다음과 같다.

```
sage: A=matrix(QQ,2,2,[1,-3,-3,1])
```

$$\begin{bmatrix} 1 & -3 \end{bmatrix}$$

$$\begin{bmatrix} -3 & 1 \end{bmatrix}$$

```
sage: f=A.charpoly();f
```

$$x^2 - 2x - 8$$

```
sage: factor(f)
```

$$(x - 4) * (x + 2)$$

```
sage: A.eigenvalues()
```

$$\begin{bmatrix} 4, & -2 \end{bmatrix}$$

```
sage: A.eigenvectors_right()
```

$$[(4, [(1, -1)], 1), (-2, [(1, 1)], 1)] \quad \# \text{ [(고유값, [고유벡터], 중복도)]}$$



```
message = "Hello World"
```

```
print message
```

```
sin?
```

```
f(x) = 1 - sin(x)^2
```

```
print f
```

```
print maxima(f)
```

```
f.simplify_trig()
```

```
f(x=pi/2)
```

```
f(x=pi/3)
```

```
integrate(f, x).simplify_trig()
```

```
print maxima(integrate(f, x).simplify_trig())
```

```
f.differentiate(2).substitute({x: 3/pi})
```

```
print maxima(f.differentiate(2).substitute({x: 3/pi}))
```

```

v=[2,3,range(4,7), QQ[x]]
v or v[2]
M=matrix([[1,42], [3,4]])
M[0,1]
f(x)=x^3-x;plot(f,-2,2)
g(x,y)= y^2-x^3+2; g(3,5)
plot(x^3-x, -2, 2)
var('y')
p1 = implicit_plot(y^2==x^3-2,(x,-3,3), (y,-3,3),color= `red`)
p2 = implicit_plot(y^2==x^3+2,(x,-3,3), (y,-3,3),color= `blue`)
show(p1+p2)

```

- 방정식해 구하기

```

sage: x = var('x')
sage: solve(x^2 + 3*x + 2, x)
x = -2, x = -1

```

-미분방정식 해구하기

```

sage: t = var('t') # define a variable t
sage: x = function('x',t) # define x to be a function of that variable
sage: DE = diff(x, t) + x - 1
sage: desolve(DE, [x,t])
(c + e^t)*e^(-t)

```

- 미적분 구하기

```

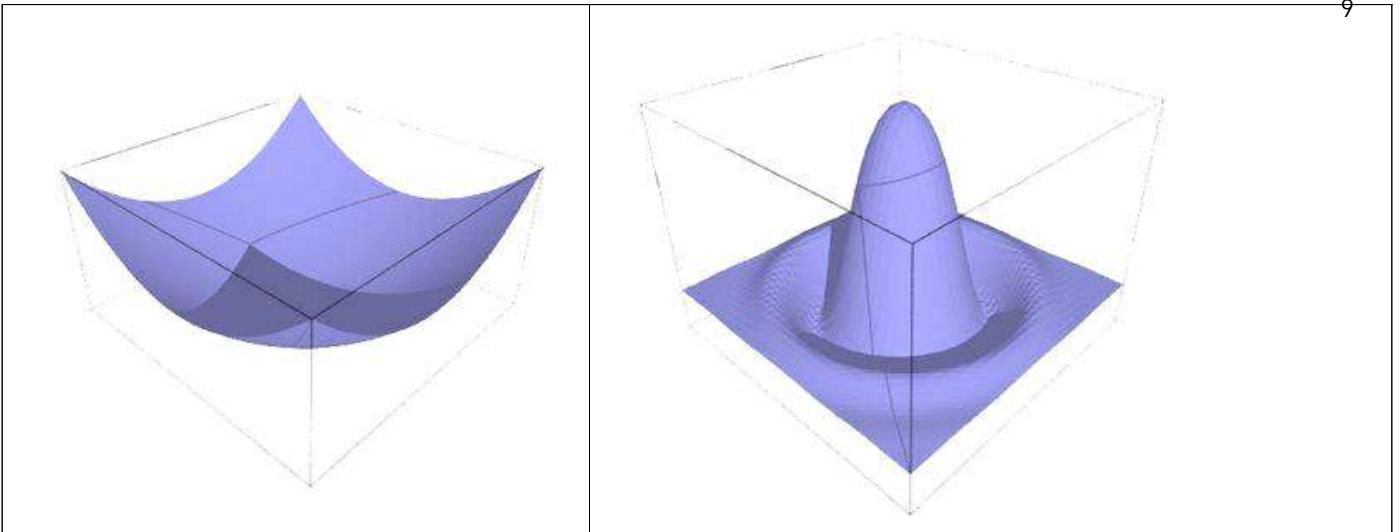
sage: x = var('x')
sage: diff(sin(x), x)
cos(x)
sage: integral(x*sin(x^2), x)
-1/2*cos(x^2)

```

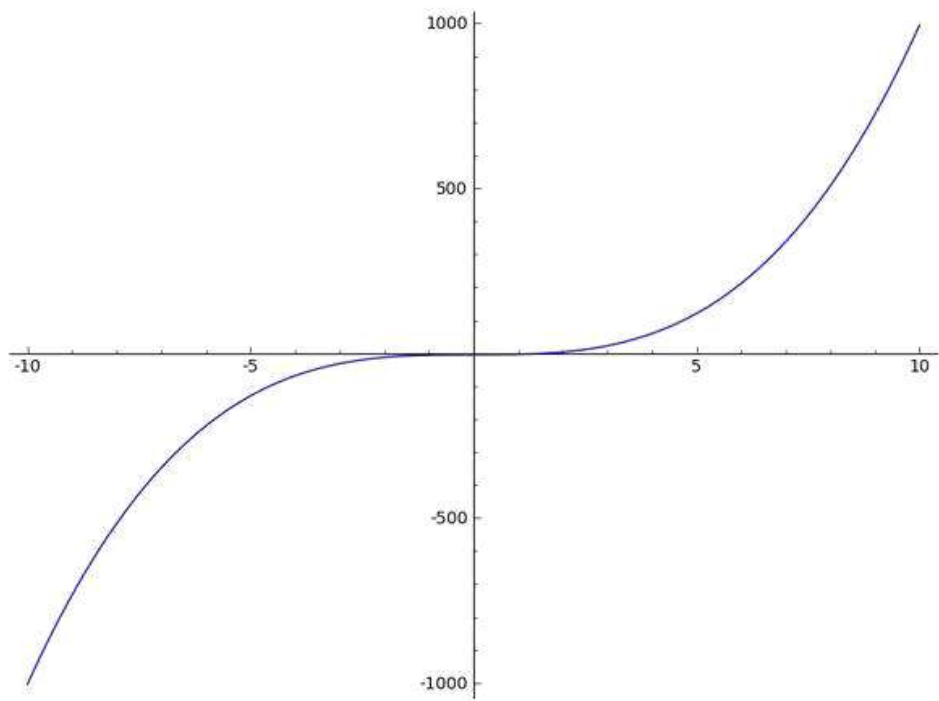
- 함수 그리기

sage: circle((0,0), 1, rgbcolor=(1,1,0)) % 중심이 원점, 반경1, 노란색 원

var('x y'); plot3d(x^2 + y^2, (x,-2,2), (y,-2,2))	var('x y');plot3d(cos((x^2+y^2)^2)*e^-(x^2+y^2), (x,-2,2),(y,-2,2)) show(aspect_ratio=(1,1,2.5))
---	---



```
f(x) = (x^3) - 5
plot (f(x), (x, -10, 10))
```



sage: @interact

```
def interact_fun(a=5, y=(0,20,1)): print a + y
```

