

1. Sage 킷 레퍼런스 (기초 수학)

Peter Jipsen, version 1.1, 이상구, 이재화

latest version at wiki.sagemath.org/quickref
GNU Free Document License, extend for your own use

목표: 수학기호를 Sage 명령어로 구현
한글화: 성균관대학교 수학과

노트북 (그리고 명령 line)

입력실행: <shift+enter>

명령어 일부를 입력하고 <tab>을 누르면 해당하는 전체 명령어가 나타난다.

명령어?를 입력하고 <tab>을 누르면 해당 명령어 문서를 보여준다.

명령어??를 입력하고 <tab>을 누르면 소스를 보여준다.

x.<tab> 객체 x 에 대한 모든 방법을 보여준다.

(자세한 것은 dir(x))

search_doc('문자열 또는 정규표현식') 문서와 연결된 링크를 보여준다.

search_src('문자열 또는 정규표현식') 소스와 연결된 링크를 보여준다.

lprint() LaTeX 출력모드로 출력한다.

version() Sage 버전을 출력한다.

셀 입력: 셀 사이의 파란색 선을 클릭한다.

셀 삭제: 내용을 삭제하고 <backspace>를 누른다.

숫자 유형

정수: Z=ZZ e.g. -2 -1 0 1 10^100

유리수: Q=QQ e.g. 1/2 1/1000 314/100-42

소수: R ≈ RR e.g. .5 0.001 3.14 -42.

복소수: C ≈ CC e.g. 1+i 2.5-3*i

기본 상수와 함수

상수: pi=pi e=e i=i ∞=OO (영문자 o)

근사값: pi.n(digits=18)=3.14159265358979324

함수: sin cos tan sec csc cot sinh cosh tanh sech csch coth log ln exp

ab=a*b a/b a^b sqrt(x)

n√x=x^(1/n) |x|=abs(x) log_b(x)=log(x,b)

변수 정의: e.g. t, u, v, y, z = var('t u v y z')

함수 정의: e.g. f(x)=x^2 f(x)=x^2

or f=lambda x: x^2 or def f(x): return x^2

연산

factor(...) expand(...) (...).simplify_...

방정식: f(x)==g(x)

_ 은 이전의 출력결과이다.

_+a _-a *_a /a 방정식에 조작을 가한다.

f(x)=g(x) 풀이: solve(f(x)==g(x),x)

solve([f(x,y)==0, g(x,y)==0], x,y)

find_root(f(x), a, b) f(x) ≈ 0이 되는 x ∈ [a,b]를 찾는다.

∑_{i=k}^n f(i)=sum([f(i) for i in [k..n]])

∏_{i=k}^n f(i)=prod([f(i) for i in [k..n]])

<미적분(Calculus)>

lim_{x→a} f(x)=limit(f(x), x=a) 예 limit(x^5, x=5)

lim_{x→a^-} f(x)=limit(f(x), x=a, dir='minus')

lim_{x→a^+} f(x)=limit(f(x), x=a, dir='plus')

d/dx (f(x))=diff(f(x),x)

∂/∂x (f(x,y))=diff(f(x,y),x)

diff=differentiate=derivative

∫ f(x)dx=integral(f(x),x)

integral=integrate

∫_a^b f(x)dx=integral(f(x),x,a,b)

a에서 n차의 Taylor 다항식:

taylor(f(x),x,a,n)

2d 그래픽

`line([(x1,y1),..., (xn,yn)], options)`

`polygon([(x1,y1),..., (xn,yn)], options)`

`circle((x,y), r, options)` 예: `circle((3,3),2)`

`text("txt",(x,y), options)`

`options` 의 사용은 `plot.options` 에서의 `options` 과 같다. e.g. `thickness=pixel`, `rgbcolor=(r,g,b)`, `hue=h` 여기서 $0 \leq r, b, g, h \leq 1$ 이다.

`figsize=[w,h]` option을 사용하여 그림의 가로, 세로비율을 조절할 수 있다.

`plot(f(x), xmin, xmax, options)`

`parametric plot((f(t),g(t)), tmin, tmax, options)`

`polar plot(f(t), tmin, tmax, options)`

여러 그래프 동시에 보여주기:

`circle((1,1),1)+line([(0,0),(2,2)])`

`animate(개체 리스트, options).show(delay=20)`

3d 그래픽

`line3d([(x1,y1,z1),..., (xn,yn,zn)], options)`

`sphere((x,y,z), r, options)`

`tetrahedron((x,y,z), size, options)`

`cube((x,y,z), size, options)`

`octahedron((x,y,z), size, options)`

`dodecahedron((x,y,z), size, options)`

`icosahedron((x,y,z), size, options)`

`options` e.g. `aspect_ratio=[1,1,1]` `color='red'`

`opacity`

`plot3d(f(x,y), [xb,xe], [yb,ye], options)`

option `plot_points=[m,n]`을 덧붙이거나

`plot3d_adaptive`를 사용한다.

`parametric_plot3d((f(t), g(t), h(t)), [tb,te],`

`options)`

`parametric_plot3d((f(u,v), g(u,v), h(u,v),`

`[ub,ue], [vb,ve], options)`

+를 사용하여 개체들을 결합한다.

<이산 수학>

$\lfloor x \rfloor = \text{floor}(x)$ $\lceil x \rceil = \text{ceil}(x)$

n 을 k 로 나눈 나머지는 $n\%k$ $k|n$ iff $n\%k==0$

$n! = \text{factorial}(n)$ $\binom{x}{m} = \text{binomial}(x,m)$

$\phi = \text{golden_ratio}$ $\phi(n) = \text{euler_phi}(n)$

문자열: e.g. `s='Hello'="Hello"=""+"He"+"llo"`

`s[0]='H' s[-1]='o' s[1:3]='el' s[3:]='lo'`

리스트: e.g. `[1, 'Hello', x] = []+[1,`

`'Hello']+[x]`

순서쌍: e.g. `(1, 'Hello', x)` (변경할 수 없음)

집합: e.g. `{1, 2, 1, a} = Set([1, 2, 1, 'a'])`

`(={1, 2, a})`

List comprehension \approx 집합 표시 방법, e.g.

`{f(x):x∈X,x>0} = Set([f(x) for x in X if x>0])`

<선형대수학(Linear algebra)>

$\begin{pmatrix} 1 \\ 2 \end{pmatrix} = \text{vector}([1, 2])$

$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = \text{matrix}([[1, 2], [3, 4]])$

$\begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix} = \text{det}(\text{matrix}([[1, 2], [3, 4]]))$

$Av = A*v$ $A^{-1} = A^{-1}$ $A^t = A.\text{transpose}()$

방법: `nrows()` `ncols()` `nullity()` `rank()`

`trace()...`

Sage 모듈과 패키지

`from module_name import *` (many preloaded)

e.g. `calculus coding combinat crypto functions`

`games geometry graphs groups logic matrix`

`numerical plot probability rings sets stats`

`sage.module_name.all.<tab>` 전환된 명령어를

보여준다.

표준패키지: Maxima GP/PARI GAP Singular R

Shell...

옵션패키지: Biopython Fricas(Axiom) Gnuplot

Kash...

`%package_name` 을 입력하고 패키지 명령어 문법을 사용한다.

`time command` 시간 정보를 보여준다.

2. Sage 킷 레퍼런스 (미적분)

William Stein, 이상구, 이재화

Sage Version 3.4

<http://wiki.sagemath.org/quickref>

GNU Free Document License, extend for your own use

한글화: 성균관대학교 수학과

내장된 상수와 함수

상수: $\pi=pi$ $e=e$ $i=I=i$

$\infty=oo=infinity$ $NaN=NaN$ $\log(2)=log2$

$\phi=golden_ratio$ $\gamma=euler_gamma$

$0.915 \approx catalan$ $2.685 \approx khinchin$

$0.660 \approx twinprime$ $0.261 \approx merten$ $1.902 \approx brun$

근사값: $pi.n(digits=18)=3.14159265358979324$

내장함수: \sin \cos \tan \sec \csc \cot \sinh
 \cosh \tanh sech csch coth \log \ln $\exp \dots$

문자식 표현 정의

변수 생성: $t, u, v, y, z = \operatorname{var}('t u v y z')$

$\operatorname{var}("t u theta")$ 또는 $\operatorname{var}("t,u,theta")$

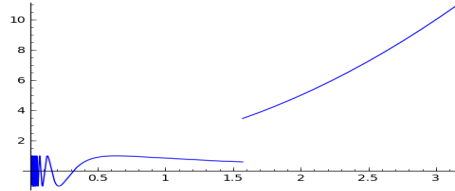
곱셈은 $*$ 를 사용하고, 지수는 $^$ 를 사용한다.

$2x^5 + \sqrt{2} = 2*x^5 + \operatorname{sqrt}(2)$

조판: $\operatorname{show}(2*theta^5 + \operatorname{sqrt}(2)) \rightarrow 2\theta^5 + \sqrt{2}$

함수그래프 $\operatorname{plot}(\sin(1/x), (0, pi/2))$ 구분 함수:
 $\operatorname{Piecewise}([(0, pi/2), \sin(1/x)], [(pi/2, pi),$

$x^2+1]]$



Symbolic function

문자로 표현된 함수(적분, 미분 가능, etc.):

$f(a, b, theta) = a + b*theta^2$

또한 θ 의 함수에 대한 정식 표현:

$f = \operatorname{function}('f', theta),$

파이썬(Python) 함수

정의:

$\operatorname{def} f(a, b, theta=1):$

$c = a + b*theta^2$

$\operatorname{return} c$

인라인 함수:

$f = \operatorname{lambda} a, b, theta = 1: a + b*theta^2$

단순화 및 전개

아래의 f 는 문자로 표현된 함수이다. (따라서 파이썬 함수가 아니다):

단순화: $f.\operatorname{simplify_exp}(), f.\operatorname{simplify_full}(),$
 $f.\operatorname{simplify_log}(), f.\operatorname{simplify_radical}(),$
 $f.\operatorname{simplify_rational}(), f.\operatorname{simplify_trig}()$

전개: $f.\operatorname{expand}(), f.\operatorname{expand_rational}()$

방정식

관계: $f = g: f == g, f \neq g: f != g,$

$f \leq g: f <= g, f \geq g: f >= g,$

$f < g: f < g, f > g: f > g$

$f = g$ 풀기: $\operatorname{solve}(f == g, x),$

$\operatorname{solve}([f == 0, g == 0], x, y)$

$\operatorname{solve}([x^2+y^2==1, (x-1)^2+y^2==1], x, y)$

해:

$S = \operatorname{solve}(x^2+x+1==0, x,$

$\operatorname{solution_dict}=True)$

해는 $S[0][x]$ $S[1][x]$ 이다.

정확한 해: $(x^3+2*x+1).\operatorname{roots}(x)$

실수해: $(x^3+2*x+1).\operatorname{roots}(x, \operatorname{ring}=RR)$

복소수 해: $(x^3+2*x+1).\operatorname{roots}(x, \operatorname{ring}=CC)$

인수분해

인수분해: $(x^3-y^3).\operatorname{factor}()$

(인수, 지수) 목록:

$(x^3-y^3).\operatorname{factor_list}()$

극한(Limit)

$\lim_{x \rightarrow a} f(x) = \operatorname{limit}(f(x), x=a)$

```

limit(sin(x)/x, x=0)
limf(x)=limit(f(x), x=a, dir='plus')
limit(1/x, x=0, dir='plus')
limf(x)=limit(f(x), x=a, dir='minus')
limit(1/x, x=0, dir='minus')

```

도함수

```

d/dx (f(x))=diff(f(x),x)=f.diff(x)
d/dx (f(x,y))=diff(f(x,y),x)
diff=differentiate=derivative
diff(x*y + sin(x^2) + e^(-x), x)

```

적분

```

int f(x)dx=integral(f(x),x)=f.integrate(x)
integral(x*cos(x^2), x)
int_a^b f(x)dx=integral(f(x),x,a,b)f
integral(x*cos(x^2), x, 0, sqrt(pi))
int_a^b f(x)dx ≈ numerical_integral(f(x),a,b)[0]
numerical_integral(x*cos(x^2),0,1)[0]
assume(...): 적분이 어떤 문제를 물어볼 때 사용
assume(x>0)

```

테일러 및 부분 분수 전개

a 에서 n 차의 Taylor 다항식:

```

taylor(f, x, a, n) ≈ c_0 + c_1(x-a) + ... + c_n(x-a)^n
taylor(sqrt(x+1), x, 0, 5)
부분 분수
(x^2/(x+1)^3).partial_fraction()

```

수치해 및 최적화

```

수치해: f.find_root(a, b, x)
(x^2 - 2).find_root(1,2,x)
최대화: f(x_0) = m 이 최대가 되는 (m, x_0) 찾기
f.find_maximum_on_interval(a, b, x)
최소화: f(x_0) = m 이 최소가 되는 (m, x_0) 찾기
f.find_minimum_on_interval(a, b, x)
최소화: minimize(f, 시작점)
minimize(x^2+x*y^3+(1-z)^2-1, [1, 1, 1])

```

다변수 미적분

```

그래디언트: f.gradient() 또는 f.gradient(변수)
(x^2+y^2).gradient([x, y])
헤시안: f.hessian()
(x^2+y^2).hessian()
야코비 행렬: jacobian(f, 변수)
jacobian(x^2 - 2*x*y, (x,y))

```

무한 급수

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$$

Sage는 아직 실행되지 않지만, Maxima를 이용할 수 있다:

```

s = 'sum (1/n^2,n,1,inf), simpsum'
SR(sage.calculus.calculus.maxima(s)) → π^2/6

```

...

(The end)