**Sage Quick Reference:**
**Elementary Number Theory**

William Stein (modified by nu)

Sage Version 3.4

http://wiki.sagemath.org/quickref



Everywhere $m, n, a, b, etc.$ are elements of `ZZ`
`ZZ` $= \mathbb{Z} =$ all integers

## Integers

$$\ldots, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, \ldots$$

$n$ divided by $m$ has *remainder* `n % m`
`gcd(n,m)`, `gcd(list)`
extended gcd $g = sa + tb = \gcd(a, b)$: `g,s,t=xgcd(a,b)`
`lcm(n,m)`, `lcm(list)`
binomial coefficient $\binom{m}{n}$ = `binomial(m,n)`
digits in a given base: `n.digits(base)`
number of digits: `n.ndigits(base)`
(*base* is optional and defaults to 10)
divides $n \mid m$: `n.divides(m)` if $nk = m$ some $k$
divisors – all $d$ with $d \mid n$: `n.divisors()`
factorial – $n!$ = `n.factorial()`

## Prime Numbers

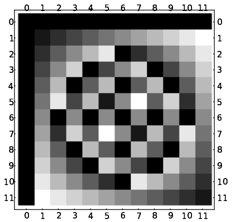
$$2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, \ldots$$

factorization: `factor(n)`
primality testing: `is_prime(n)`, `is_pseudoprime(n)`
prime power testing: `is_prime_power(n)`
$\pi(x) = \#\{p : p \le x \text{ is prime}\}$ = `prime_pi(x)`
set of prime numbers: `Primes()`
$\{p : m \le p < n \text{ and } p \text{ prime}\}$ =`prime_range(m,n)`
prime powers: `prime_powers(m,n)`
first $n$ primes: `primes_first_n(n)`
next and previous primes: `next_prime(n)`, `previous_prime(n)`, `next_probable_prime(n)`
prime powers: `next_prime_power(n)`, `pevious_prime_power(n)`
Lucas-Lehmer test for primality of $2^p - 1$

```
def is_prime_lucas_lehmer(p):
    s = Mod(4, 2^p - 1)
    for i in range(3, p+1): s = s^2 - 2
    return s == 0
```

## Modular Arithmetic and Congruences

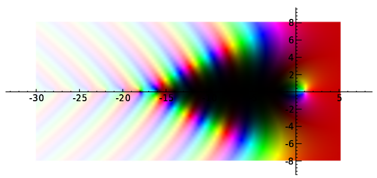
`k=12; m = matrix(ZZ, k, [(i*j)%k for i in [0..k-1] for j in [0..k-1]]); m.plot(cmap='gray')`



Euler's $\phi(n)$ function: `euler_phi(n)`
Kronecker symbol $\left(\frac{a}{b}\right)$ = `kronecker_symbol(a,b)`
Quadratic residues: `quadratic_residues(n)`
Quadratic non-residues: `quadratic_residues(n)`
ring $\mathbb{Z}/n\mathbb{Z}$ = `Zmod(n) = IntegerModRing(n)`
$a$ modulo $n$ as element of $\mathbb{Z}/n\mathbb{Z}$: `Mod(a, n)`
primitive root modulo $n$ = `primitive_root(n)`
inverse of $n \pmod{m}$: `n.inverse_mod(m)`
power $a^n \pmod{m}$: `power_mod(a, n, m)`
Chinese remainder theorem: `x = crt(a,b,m,n)`
finds $x$ with $x \equiv a \pmod{m}$ and $x \equiv b \pmod{n}$
discrete log: `log(Mod(6,7), Mod(3,7))`
order of $a \pmod{n}$ = `Mod(a,n).multiplicative_order()`
square root of $a \pmod{n}$ = `Mod(a,n).sqrt()`

## Special Functions

`complex_plot(zeta, (-30,5), (-8,8))`



$$\zeta(s) = \prod_p \frac{1}{1-p^{-s}} = \sum \frac{1}{n^s} = \texttt{zeta(s)}$$

$$\mathrm{Li}(x) = \int_2^x \frac{1}{\log(t)} dt = \texttt{Li(x)}$$

$$\Gamma(s) = \int_0^\infty t^{s-1} e^{-t} dt = \texttt{gamma(s)}$$

## Continued Fractions

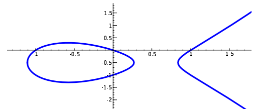
`continued_fraction(pi)`

$$\pi = 3 + \cfrac{1}{7 + \cfrac{1}{15 + \cfrac{1}{1 + \cfrac{1}{292 + \cdots}}}}$$

continued fraction: `c=continued_fraction(x, bits)`
convergents: `c.convergents()`
convergent numerator $p_n$ = `c.pn(n)`
convergent denominator $q_n$ = `c.qn(n)`
value: `c.value()`

## Elliptic Curves

`EllipticCurve([0,0,1,-1,0]).plot(plot_points=300,thickness=3)`



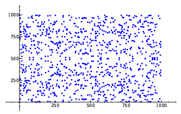
`E = EllipticCurve(`$[a_1, a_2, a_3, a_4, a_6]$`)`

$$y^2 + a_1 xy + a_3 y = x^3 + a_2 x^2 + a_4 x + a_6$$

conductor $N$ of $E$ =`E.conductor()`
discriminant $\Delta$ of $E$ =`E.discriminant()`
rank of $E$ = `E.rank()`
free generators for $E(\mathbb{Q})$ = `E.gens()`
$j$-invariant = `E.j_invariant()`
$N_p = \#\{\text{solutions to } E \text{ modulo } p\}$ = `E.Np(prime)`
$a_p = p + 1 - N_p$ =`E.ap(prime)`
$L(E, s) = \sum \frac{a_n}{n^s}$ = `E.lseries()`
$\mathrm{ord}_{s=1} L(E, s)$ = `E.analytic_rank()`

## Elliptic Curves Modulo $p$

`EllipticCurve(GF(997), [0,0,1,-1,0]).plot()`



`E = EllipticCurve(GF(p),` $[a_1, a_2, a_3, a_4, a_6]$`)`
$\#E(\mathbb{F}_p)$ = `E.cardinality()`
generators for $E(\mathbb{F}_p)$ = `E.gens()`
$E(\mathbb{F}_p)$ =`E.points()`