class **int**(object)

| int(x=0) -> int or long

| int(x, base=10) -> int or long

|

|

| Methods defined here:

|

| \_\_abs\_\_(...)

| x.\_\_abs\_\_() <==> abs(x)

|

| \_\_add\_\_(...)

| x.\_\_add\_\_(y) <==> x+y

|

| \_\_and\_\_(...)

| x.\_\_and\_\_(y) <==> x&y

|

| \_\_cmp\_\_(...)

| x.\_\_cmp\_\_(y) <==> cmp(x,y)

|

| \_\_coerce\_\_(...)

| x.\_\_coerce\_\_(y) <==> coerce(x, y)

|

| \_\_div\_\_(...)

| x.\_\_div\_\_(y) <==> x/y

|

| \_\_divmod\_\_(...)

| x.\_\_divmod\_\_(y) <==> divmod(x, y)

|

| \_\_float\_\_(...)

| x.\_\_float\_\_() <==> float(x)

|

| \_\_floordiv\_\_(...)

| x.\_\_floordiv\_\_(y) <==> x//y

|

| \_\_format\_\_(...)

|

| \_\_getattribute\_\_(...)

| x.\_\_getattribute\_\_('name') <==> x.name

|

| \_\_getnewargs\_\_(...)

|

| \_\_hash\_\_(...)

| x.\_\_hash\_\_() <==> hash(x)

|

| \_\_hex\_\_(...)

| x.\_\_hex\_\_() <==> hex(x)

|

| \_\_index\_\_(...)

| x[y:z] <==> x[y.\_\_index\_\_():z.\_\_index\_\_()]

|

| \_\_int\_\_(...)

| x.\_\_int\_\_() <==> int(x)

|

| \_\_invert\_\_(...)

| x.\_\_invert\_\_() <==> ~x

|

| \_\_long\_\_(...)

| x.\_\_long\_\_() <==> long(x)

|

| \_\_lshift\_\_(...)

| x.\_\_lshift\_\_(y) <==> x<<y

|

| \_\_mod\_\_(...)

| x.\_\_mod\_\_(y) <==> x%y

|

| \_\_mul\_\_(...)

| x.\_\_mul\_\_(y) <==> x\*y

|

| \_\_neg\_\_(...)

| x.\_\_neg\_\_() <==> -x

|

| \_\_nonzero\_\_(...)

| x.\_\_nonzero\_\_() <==> x != 0

|

| \_\_oct\_\_(...)

| x.\_\_oct\_\_() <==> oct(x)

|

| \_\_or\_\_(...)

| x.\_\_or\_\_(y) <==> x|y

|

| \_\_pos\_\_(...)

| x.\_\_pos\_\_() <==> +x

|

| \_\_pow\_\_(...)

| x.\_\_pow\_\_(y[, z]) <==> pow(x, y[, z])

|

| \_\_radd\_\_(...)

| x.\_\_radd\_\_(y) <==> y+x

|

| \_\_rand\_\_(...)

| x.\_\_rand\_\_(y) <==> y&x

|

| \_\_rdiv\_\_(...)

| x.\_\_rdiv\_\_(y) <==> y/x

|

| \_\_rdivmod\_\_(...)

| x.\_\_rdivmod\_\_(y) <==> divmod(y, x)

|

| \_\_repr\_\_(...)

| x.\_\_repr\_\_() <==> repr(x)

|

| \_\_rfloordiv\_\_(...)

| x.\_\_rfloordiv\_\_(y) <==> y//x

|

| \_\_rlshift\_\_(...)

| x.\_\_rlshift\_\_(y) <==> y<<x

|

| \_\_rmod\_\_(...)

| x.\_\_rmod\_\_(y) <==> y%x

|

| \_\_rmul\_\_(...)

| x.\_\_rmul\_\_(y) <==> y\*x

| [[1]](#endnote-1)

| \_\_ror\_\_(...)

| x.\_\_ror\_\_(y) <==> y|x

|

| \_\_rpow\_\_(...)

| y.\_\_rpow\_\_(x[, z]) <==> pow(x, y[, z])

|

| \_\_rrshift\_\_(...)

| x.\_\_rrshift\_\_(y) <==> y>>x

|

| \_\_rshift\_\_(...)

| \_\_ror\_\_(...)

| x.\_\_ror\_\_(y) <==> y|x

|

| \_\_rpow\_\_(...)

| y.\_\_rpow\_\_(x[, z]) <==> pow(x, y[, z])

|

| \_\_rrshift\_\_(...)

| x.\_\_rrshift\_\_(y) <==> y>>x

|

| \_\_rshift\_\_(...)

| x.\_\_rshift\_\_(y) <==> x>>y

|

| \_\_rsub\_\_(...)

| x.\_\_rsub\_\_(y) <==> y-x

|

| \_\_rtruediv\_\_(...)

| x.\_\_rtruediv\_\_(y) <==> y/x

|

| \_\_rxor\_\_(...)

| x.\_\_rxor\_\_(y) <==> y^x

|

| \_\_str\_\_(...)

| x.\_\_str\_\_() <==> str(x)

|

| \_\_sub\_\_(...)

| x.\_\_sub\_\_(y) <==> x-y

|

| \_\_truediv\_\_(...)

| x.\_\_truediv\_\_(y) <==> x/y

|

| \_\_trunc\_\_(...)

| Truncating an Integral returns itself.

|

| \_\_xor\_\_(...)

| x.\_\_xor\_\_(y) <==> x^y

|

| bit\_length(...)

| int.bit\_length() -> int

|

| Number of bits necessary to represent self in binary.

| >>> bin(37)

| '0b100101'

| >>> (37).bit\_length()

| 6

|

| conjugate(...)

| Returns self, the complex conjugate of any int.

|

| -----------------------------------------------------

| Data descriptors defined here:

|

| denominator

| the denominator of a rational number in lowest terms

|

| imag

| the imaginary part of a complex number

|

| numerator

| the numerator of a rational number in lowest terms

|

| real

| the real part of a complex number

|

| --------------------------------------------------------

1. Santosg572@gmail.com [↑](#endnote-ref-1)