

Sage gotchas

*examples have been changed to protect the innocent

“Gotcha (programming): a counter-intuitive, but documented, behavior in a computer system (as opposed to a bug).”

x.n()

“Returns a numerical approximation of an object ‘x’ with at least ‘prec’ bits (or decimal ‘digits’) of precision.”

x.n()

“Returns a numerical approximation of an object ‘x’ with at least ‘prec’ bits (or decimal ‘digits’) of precision.”

```
>>> 2.n(digits=10)
```

x.n()

“Returns a numerical approximation of an object ‘x’ with at least ‘prec’ bits (or decimal ‘digits’) of precision.”

```
>>> 2.n(digits=10)  
2.0000000000
```

x.n()

“Returns a numerical approximation of an object ‘x’ with at least ‘prec’ bits (or decimal ‘digits’) of precision.”

```
>>> 2.n(digits=10)
2.0000000000
2.00000000000000000000
```

x.n()

“Returns a numerical approximation of an object ‘x’ with at least ‘prec’ bits (or decimal ‘digits’) of precision.”

```
>>> 2.n(digits=10)
2.0000000000
2.00000000000000000000
2.000000000000012345678
```

Orderings

Orderings

```
>>> K.<a> = NumberField(x^2 - 3, embedding=1.71)
>>> L.<b> = NumberField(x^2 - 2, embedding=1.41)
```

Orderings

```
>>> K.<a> = NumberField(x^2 - 3, embedding=1.71)
>>> L.<b> = NumberField(x^2 - 2, embedding=1.41)
>>> a.n(), b.n()
1.73205080756888, 1.41421356237309
```

Orderings

```
>>> K.<a> = NumberField(x^2 - 3, embedding=1.71)
>>> L.<b> = NumberField(x^2 - 2, embedding=1.41)
>>> a.n(), b.n()
    1.73205080756888, 1.41421356237309
>>> a < b
```

Orderings

```
>>> K.<a> = NumberField(x^2 - 3, embedding=1.71)
>>> L.<b> = NumberField(x^2 - 2, embedding=1.41)
>>> a.n(), b.n()
    1.73205080756888, 1.41421356237309
>>> a < b
    True
```

Orderings

```
>>> K.<a> = NumberField(x^2 - 3, embedding=1.71)
>>> L.<b> = NumberField(x^2 - 2, embedding=1.41)
>>> a.n(), b.n()
    1.73205080756888, 1.41421356237309
>>> a < b
    True
>>> id(a), id(b)
    6797335688, 6797399752
```

Bit operations

Bit operations

```
>>> x = 6; y = 5
```

Bit operations

```
>>> x = 6; y = 5
>>> bin(x), bin(y)
'0b110', '0b101'
```


Bit operations

```
>>> x = 6; y = 5
>>> bin(x), bin(y)
'0b110', '0b101'
>>> x & y
```

Bit operations

```
>>> x = 6; y = 5
>>> bin(x), bin(y)
'0b110', '0b101'
>>> x & y
4
```

Bit operations

```
>>> x = 6; y = 5
>>> bin(x), bin(y)
'0b110', '0b101'
>>> x & y
4
>>> x | y
```

Bit operations

```
>>> x = 6; y = 5
>>> bin(x), bin(y)
'0b110', '0b101'
>>> x & y
4
>>> x | y
7
```

Bit operations

```
>>> x = 6; y = 5
>>> bin(x), bin(y)
'0b110', '0b101'
>>> x & y
4
>>> x | y
7
>>> x ^ y
```

Bit operations

```
>>> x = 6; y = 5
>>> bin(x), bin(y)
'0b110', '0b101'
>>> x & y
4
>>> x | y
7
>>> x ^ y
7776
```

Bit operations

```
>>> x = 6; y = 5
>>> bin(x), bin(y)
'0b110', '0b101'
>>> x & y
4
>>> x | y
7
>>> x ^ y
7776
>>> x ^^ y
```

Bit operations

```
>>> x = 6; y = 5
>>> bin(x), bin(y)
'0b110', '0b101'
>>> x & y
4
>>> x | y
7
>>> x ^ y
7776
>>> x ^^ y
3
```


Bit operations

```
>>> x = 6; y = 5
>>> bin(x), bin(y)
'0b110', '0b101'
>>> x & y
4
>>> x | y
7
>>> x ^ y
7776
>>> x ^^ y
3
>> ~x
```

Bit operations

```
>>> x = 6; y = 5
>>> bin(x), bin(y)
'0b110', '0b101'
>>> x & y
4
>>> x | y
7
>>> x ^ y
7776
>>> x ^^ y
3
>> ~x
1/6
```