

Complex

Support for complex number arithmetic in JavaScript

Constructor

```
new Complex(real,imaginary)    // Constructor  
Complex(value)                // Conversion
```

Arguments

real The real part of a complex number
imaginary The imaginary part of a complex number

If the imaginary argument is omitted it is treated as zero

If there are no arguments it is treated as a complex zero

If *Complex* is invoked as a conversion the value parameter is converted to a *complex number* and returned.

Returns

Returns a Complex object initialize with the real and imaginary value. If *Complex* is invoked as a conversion the *value* parameter is converted to a *complex number* and returned. If *value* is another Complex number then that is returned if *value* is undefined, a *Complex.zero* is returned.

Method

abs Return the magnitude of a complex number
arg Return the angle of the polar representation of the Complex number.
conj Return a new Conjugated Complex object.
imag return the Imaginary part of the Complex number.
negate Return a new Negated Complex Object.
norm Return the norm of the complex number.
real Return the real part of the Complex number.
toExponential Converts a Complex number to a string using exponential notation with the specified number of digits after tge Decimal place.
toFixed Converts a Complex number to a string that contains a specified number of digits after the decimal place.
toPrecision Convert a Complex number to a string using the specified number of precision digits. Uses exponential or fixed point notation depending on the size of the number and the number of significant digits specified.
toString Convert a Complex number to a string using a specified radix(base)
valueOf The primitive real number value of this Complex number object.

Functions

abs()	Return the magnitude of a complex number
add()	Return the addition of two complex numbers
acos()	Return arc cosine of the complex number
acosh()	Return the arc cosine hyperbolic of the complex number
asin()	Return the arcsine of the complex number
asinh()	Return the arc sine hyperbolic of the complex number
atan()	Return the arc tangent of the complex number
atanh()	Return the arctangent hyperbolic of the complex number
beta()	Return the beta function of a complex number
cos()	Return cosine of the complex number
cosh()	Return the cosine hyperbolic of the complex number
div()	Return the division of two complex numbers.
equal()	Return the Boolean value (true, false) of the equality of two complex numbers.
exp()	Return the complex power of e.
gamma()	Return the gamma function of a complex number
lgamma()	Return the logarithm of the gamma function of a complex number
log()	Return the complex natural logarithm.
log10()	Return the complex base 10 logarithm.
mul()	Return the product of two complex numbers.
polar()	Return the complex number of the polar representation.
pow()	Return the complex power of x^y .
sin()	Return the sine of the complex number
sinh()	Return the sine hyperbolic of the complex number
sub()	Return the difference between two complex numbers.
sqrt()	Return the complex square root.
tan()	Return the tangent of the complex number
tanh()	Return the tangent hyperbolic of the complex number

Constants

zero	return a new Complex(0,0) object
one	return a new Complex(1,0) object
i	return a new Complex(0,1) object

Miscellaneous

parseComplex()	Parse a Complex float number string
----------------	-------------------------------------

Complex.abs()

Return the absolute magnitude of the Complex number

Synopsis

complex object.abs()

Returns

The magnitude of the complex number is returned. Special calculation are made to prevent intermediate result to overflow.

Example

```
var z = new Complex(3,4);  
z.abs()           // result 5
```

See Also

Complex.norm()

Complex.arg()

Return the angle of the polar representation of the complex number

Synopsis

complex object.arg()

Returns

Return the angle in radians of the polar representation of the complex number

Example

```
var z = new Complex(3,4);  
z.arg()           // result 0.927295...
```

See Also**Complex.add()**

Add two complex numbers

Synopsis

Complex.add(a,b)

Arguments

a,b The complex numbers to be added.

Returns

The result of the complex addition.

Example

```
var z = new Complex(3,4);  
var y=new Complex(1,2);
```

```
Complex.add(z,y)            // result (4+i6)
```

See Also

Complex.div(), Complex.mul(), Complex.sub()

Complex.acos()

Return the arc cosine of the complex number

Synopsis

Complex.acos(a)

Returns

Return the arc cosine of the complex number.

Example

```
var z = new Complex(3,4);  
Complex.acos(z)            // result (0.9368124611557193-i2.3055090312434685)
```

See Also

Complex.asin(), Complex.atan()

Complex.acosh()

Return the arc cosine hyperbolic of the complex number

Synopsis

Complex.acosh(a)

Returns

Return the arc cosine hyperbolic of the complex number.

Example

```
var z = new Complex(3,4);  
Complex.acosh(z) // result (2.305509031243477+i0.9368124611557199)
```

See Also

Complex.asinh(), Complex.atanh()

Complex.asin()

Return the arcsine of the complex number

Synopsis

Complex.asin(a)

Returns

Return the arc sine of the complex number.

Example

```
var z = new Complex(3,4);  
Complex.asin(z) // result (0.6339838656391773+i2.3055090312434685)
```

See Also

Complex.acos(), Complex.atan()

Complex.asinh()

Return the arc sine hyperbolic of the complex number

Synopsis

Complex.asinh(a)

Returns

Return the arc sine hyperbolic of the complex number.

Example

```
var z = new Complex(3,4);  
Complex.asinh(z) // result (2.2999140408792695+i0.9176168533514787)
```

See Also

Complex.acosh(), Complex.atanh()

Complex.atan()

Return the arctangent of the complex number

Synopsis

Complex.atan(a)

Returns

Return the arc tangent of the complex number.

Example

```
var z = new Complex(3,4);  
Complex.atan(z) // result (1.4483069952314644+i0.15899719167999904)
```

See Also

Complex.acos(), Complex.asin()

Complex.atanh()

Return the arc tangent hyperbolic of the complex number

Synopsis

Complex.atanh(a)

Returns

Return the arc tanh hyperbolic of the complex number.

Example

```
var z = new Complex(3,4);  
Complex.atanh(z) // result (0.11750090731143381+i1.4099210495965755)
```

See Also

Complex.acosh(), Complex.asinh()

Complex.beta()

Return the beta function of the complex numbers

Synopsis

Complex.beta(x,y)

Returns

Return the beta function of the complex number a and b.

Example

```
var x=new Complex(3,4), y=Complex(2,-3);  
Complex.beta(x,y) // result (0.000594954...,0.000791355...)
```

See Also

Complex.gamma(), Complex.lgamma()

Complex.cos()

Return the cosine of the complex number

Synopsis

Complex.cos(a)

Returns

Return the cosine of the complex number.

Example

```
var z = new Complex(3,4);
```

```
Complex.cos(z)           // result (-27.034945603074224-i3.851153334811777)
```

See Also

Complex.sin(), Complex.tan()

Complex.cosh()

Return the cosine hyperbolic of the complex number

Synopsis

```
Complex.cosh(a)
```

Returns

Return the cosine hyperbolic of the complex number.

Example

```
var z = new Complex(3,4);  
Complex.cosh(z)           // result (-6.580663040551157-i7.581552742746545)
```

See Also

Complex.sinh(), Complex.tanh()

Complex.conj()

Return the conjugated form of the complex number

Synopsis

```
complex object.conj()
```

Returns

Return the complex conjugated form.

Example

```
var z = new Complex(3,4);  
z.conj()                   // result (3-i4)
```

See Also

Complex.negate()

Complex.div()

Divide two complex numbers

Synopsis

Complex.div(a,b)

Arguments

a,b The complex numbers to be divided. Special calculation are made to prevent intermediate result to overflow.

Returns

The result of the complex division a/b .

Example

```
var z = new Complex(3,4);  
var y=new Complex(1,2);
```

```
Complex.div(z,y)            // result (2.2+i0.4)
```

See Also

Complex.add(), Complex.mul(), Complex.sub()

Complex.equal()

Compare two complex numbers for equality

Synopsis

Complex.equal(a,b)

Arguments

a,b The complex numbers to be compare for

Returns

The Boolean value of the equal comparison.

Example

```
var z = new Complex(3,4);
var y=new Complex(1,2);

if( Complex.equal(z,y))... // result false
if( Complex.equal(z,z))... // result true
if( !Complex.equal(z,y))... // result true ! to do “notequal” comparison
```

See Also**Complex.exp()**

Compute e^x **Synopsis**

Complex.exp(x)

Arguments

x A complex numbers to be used as the exponent

Returns

e^x , e raised to the power of the specified exponent x , where e is the base of the natural logarithm, with a value of approximately 2.71828.

Example

```
var z = new Complex(3,4);

Complex.exp(z)... // ez approximately (-13.128+i5.200)
```

See Also

Complex.log(), Complex.log10(), Complex.pow()

Complex.gamma()

Return the gamma function of the complex number

Synopsis

Complex.gamma(x)

Returns

Return the gamma function of the complex number x .

Example

```
var x=new Complex(3,4);  
Complex.gamma(x) // result (0.00522553...,0.1725470...)
```

See Also

Complex.lgamma(), Complex.beta()

Complex.i

Return complex i

Synopsis

Complex.i

Returns

The complex constant i ($0+i1$).

Example

```
var z = Complex.i; // z=(0+i1)
```

See Also

Complex.one, Complex.zero

Complex.imag()

Return or Set the imaginary part of the complex number

Synopsis

complex object.imag(*i*)

Arguments

i The optional imaginary value when setting the imaginary number to a new value. If omitted the call returns the actual imaginary value of the complex number.

Returns

Return the imaginary part of the complex number.

Example

```
var z = new Complex(3,4);  
z.imag();           // result 4  
z.imag(5);         // set z to (3+i5) and return 5
```

See Also

Complex.real()

Complex.lgamma()

Return the logarithm gamma function of the complex number

Synopsis

Complex.lgamma(x)

Returns

Return the logarithm gamma function of the complex number x.

Example

```
var x=new Complex(3,4);  
Complex.lgamma(x)    // result (-1.756626...,4.742664...)
```

See Also

Complex.gamma(), Complex.beta()

Complex.log()

Compute the natural logarithm of x

Synopsis

Complex.log(x)

Arguments

x A complex numbers not equal to zero

Returns

Return log(x)

Example

```
var z = new Complex(3,4);
```

```
Complex.log(z)...                    // log(x) approximately (1.609+i0.927)
```

See Also

Complex.exp(), Complex.log10(),

Complex.log10()

Compute the base-10 logarithm of x

Synopsis

Complex.log10(x)

Arguments

x A complex numbers not equal to zero

Returns

Return log₁₀(x)

Example

```
var z = new Complex(3,4);
```

```
Complex.log10(z)...                    // log10(3+i4) approximately (0.699+i0.403)
```

See Also

Complex.exp(), Complex.log(),

Complex.mul()

Multiply two complex numbers

Synopsis

Complex.mul(a,b)

Arguments

a,b The complex numbers to be multiplied.

Returns

The result of the complex multiplication.

Example

```
var z = new Complex(3,4);  
var y=new Complex(1,2);
```

```
Complex.mul(z,y)            // result (-5+i10)
```

See Also

Complex.add(), Complex.div(), Complex.sub()

Complex.negate()

Return the negated complex number

Synopsis

complex object.negate()

Returns

Return the negated complex number.

Example

```
var z = new Complex(3,4);  
z.negate()                    // result (-3-i4)
```

See Also

Complex.conj()

Complex.norm()

Return the norm (square magnitude) of the Complex number

Synopsis

complex object.norm()

Returns

The norm (squared magnitude) of the complex number is returned.

Example

```
var z = new Complex(3,4);  
z.norm()           // result 25
```

See Also

Complex.abs()

Complex.one

Return complex one

Synopsis

Complex.one

Returns

The complex constant one (1+i0).

Example

```
var z = Complex.one;      // z=(1+i0)
```

See Also

Complex.zero, Complex.i

Complex.polar()

Convert polar coordinates into a complex number

Synopsis

Complex.polar(mag,arg)

Arguments

mag Magnitude of complex number
arg Angle of Complex number.

Returns

Return the complex number.

Example

```
var z = Complex.polar( 4, 0.5 );  
  
Complex.polar(4,0.5)        // result (3.510+i1.918)
```

See Also**Complex.pow()**

Compute x^y

Synopsis

Complex.pow(x,y)

Arguments

x A complex numbers to be raised to a power
y A complex power that x is raised to

Returns

X to the power of y. x^y

Example

```
var x = new Complex(3,4);  
var y = new Complex( 1 , 2 )
```

```
Complex.pow(x,y)... // xy approximately (-0.4198+i0.6605)
```

See Also

Complex.exp()

Complex.real()

Return or Set the real part of the complex number

Synopsis

```
complex object.real(r)
```

Arguments

r The optional real value when setting the real number to a new value. If omitted the call returns the actual real value of the complex number.

Returns

Return the real part of the complex number.

Example

```
var z = new Complex(3,4);  
z.real(); // result 3  
z.real(5); //set the real part to 5 and return the result 5
```

See Also

Complex.imag()

Complex.sin()

Return the sine of the complex number

Synopsis

```
Complex.sin(a)
```

Returns

Return the sine of the complex number.

Example

```
var z = new Complex(3,4);  
Complex.sin(z)           // result (3.853738037919377-i27.016813258003932)
```

See Also

Complex.cos(), Complex.tan()

Complex.sinh()

Return the sine hyperbolic of the complex number

Synopsis

Complex.sinh(a)

Returns

Return the sine hyperbolic of the complex number.

Example

```
var z = new Complex(3,4);  
Complex.sinh(z)          // result (-6.580663040551157-i7.581552742746545)
```

See Also

Complex.cosh(), Complex.tanh()

Complex.sub()

Subtract two complex numbers

Synopsis

Complex.sub(a,b)

Arguments

a,b The complex numbers to be subtracted.

Returns

The result of the complex subtraction.

Example

```
var z = new Complex(3,4);  
var y=new Complex(1,2);
```

```
Complex.sub(z,y)           // result (2+i2)
```

See Also

Complex.add(), Complex.div(), Complex.mul()

Complex.sqrt()

Compute a complex square root

Synopsis

```
Complex.sqrt(x)
```

Arguments

x A complex numbers to be square rooted. Special calculation are made to prevent intermediate result to overflow.

Returns

The square root of *x*.

Example

```
var z = new Complex(3,4);
```

```
Complex.sqrt(z)...         // Result (2+i1)
```

See Also**Complex.tan()**

Return the tangent of the complex number

Synopsis

```
Complex.tan(a)
```

Returns

Return the tangent of the complex number.

Example

```
var z = new Complex(3,4);  
Complex.tan(z) // result (-0.00018734620462947842+i0.9993559873814732)
```

See Also

Complex.cos(), Complex.sin()

Complex.tanh()

Return the tangent hyperbolic of the complex number

Synopsis

Complex.tanh(a)

Returns

Return the tanh hyperbolic of the complex number.

Example

```
var z = new Complex(3,4);  
Complex.tanh(z) // result (-6.580663040551157-i7.581552742746545)
```

See Also

Complex.cosh(), Complex.sinh()

Complex.toExponential()

Format a number using exponential notation

Synopsis

complex.toExponential(digits)

Arguments

Digits The number of digits that will appear after the decimal point. This may be a value between 0 and 20, inclusive. If this argument is omitted, as many digits as necessary will be used. A complex number is always formatted as:

$(real_part \pm i imaginary_part)$

Returns

A string representations of the complex number, in exponential notation, with one digit before the decimal place and *digits* digits after the decimal place. The fractional part of the complex number is rounded, or padded with zeros, as necessary, so that it has the specified length.

Example

```
var z = new Complex( 12345.6789, 12345.6789 );
z.toExponential(1);      // result (1.2e+4+i1.2e+4)
z.toExponential(5);     // result (1.23457e+4+i1.23457e+4)
z.toExponential(10);    // result (1.23456789000e+4+i1.23456789000e+4)
z.toExponential();      // result (1.23456789e+4+i1.23456789e+4)
```

See Also

`Complex.toFixed()`, `Complex.toPrecision()`, `Complex.toString()`

Complex.toFixed()

Format a number using fixed-point notation

Synopsis

`complex.toFixed(digits)`

Arguments

Digits The number of digits that will appear after the decimal point. This may be a value between 0 and 20, inclusive. If this argument is omitted, it is treated as zero. A complex number is always formatted as:

$(real_part \pm i imaginary_part)$

Returns

A string representations of the complex number, that does not use exponential notation and has exactly *digits* digits after the decimal point. The *complex number* is rounded as necessary, and the fraction part is padded with zeros if necessary so that it has the

specified length. If the *complex number* is greater than $1e+21$, this method simple calls *number.toString()* and return a string in exponential notation.

Example

```
var z = new Complex( 12345.6789, 12345.6789 );
z.toFixed(5);           // result (12345.7+i12345.7)
z.toFixed(6);           // result (12345.678900+i12345.678900)
z.toFixed();            // result (12346+i1234.6)
```

See Also

[Complex.toExponential\(\)](#), [Complex.toPrecision\(\)](#), [Complex.toString\(\)](#)

Complex.toPrecision()

Format the significant digits of a complex number

Synopsis

complex.toPrecision(digits)

Arguments

Digits The number of significant digits to appear in the returned string. This may be a value between 1 and 21, inclusive. If this argument is omitted, the *toString()* method is used instead to convert the complex number to a base-10 value. A complex number is always formatted as:
(real_part ±i imaginary_part)

Returns

A string representations of the *complex number*, that contains *precisions* significant digits. If *precision* is large enough to include all the digits of the integer part of number, the returned string uses fixed-point notation. Otherwise exponential notation is used with one digit before the decimal place and *precision* – 1 digits after the decimal place. The number is rounded or padded with zeros as necessary.

Example

```
var z = new Complex( 12345.6789, 12345.6789 );
z.toPrecision(1);       // result (1e+4+i1e+4)
z.toPrecision(3);       // result (1.23e+4+i1.2e+4)
z.toPrecision(5);       // result (12346+i12346)
```

See Also

Complex.toExponential(), Complex.toFixed(), Complex.toString()

Complex.toString()

Format the significant digits of a complex number

Synopsis

complex.toString(radix)

Arguments

Radix If omitted the base 10 will be used to convert the complex number to a string. Otherwise the radix will be used (2..36). A complex number is always formatted as:

(real_part ±i imaginary_part)

Returns

A string representations of the *complex number*, in the indicated radix.

Example

```
var z = new Complex( 12345.6789, 12345.6789 );
z.toString();           // result (1234.6789+i1234.6789)
```

See Also

Complex.toExponential(), Complex.toFixed(), Complex.toPrecision()

Complex.valueOf()

Return the primitive number value

Synopsis

complex object.valueOf()

Returns

The primitive value of the *complex number* is returned, which is the same as *complex number*.real().

Example

```
var z = new Complex(3,4);
z.valueOf()           // return 3
```

See Also**Complex.zero**

Return complex zero

Synopsis

Complex.zero

Returns

The complex constant zero ($0+i0$).

Example

```
var z = Complex.zero;
```

See Also

Complex.one, Complex.i

parseComplex()

Convert a string to a complex number

Synopsis

parseComplex(s)

Arguments

s The string to be parsed and converted to a *complex number*.

Returns

`parseComplex()` parses and return a new Complex number contained in *s*.
`parseComplex()` return a Complex NaN number if parsing fails. A complex number can either be in the format:

(real_part ±i imaginary_part)

Where either the *real_part* or the *imaginary_part* can be missing but not both at the same time. `parseComplex()` can also parsed string omitting the leading and trailing parentheses.

Example

```
var z = parseComplex( "(1.2 -i 3.4E-5)"); // result (1.2-i3.4E-5)
z = parseComplex( "(1.2)"); // result (1.2 +i0)
z = parseComplex( "(-i1.2)"); // result (0 -i1.2)
```

See Also