ComplexBF

Support for ComplexBF arbitrary precision number arithmetic in JavaScript. Require BigFloat.js as the based of the ComplexBF packages.BigFloat.js delivered the underlying arbitrary precision arithmetic that is the based of the ComplexBF packages. To get acquainted please read the BigFloat JavaScript package document.

Constructor

new ComplexBF(*real*,*imaginary*)	// Constructor
ComplexBF(value)	// Conversion

Arguments

real	The real part of a ComplexBF number
imaginary	The imaginary part of a ComplexBF number

If the imaginary argument is omitted it is treated as zero
If there are no arguments it is treated as a ComplexBF zero
If *ComplexBF* is invoked as a conversion the value parameter is converted to a *ComplexBF number* and returned.

Returns

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

Properties

abs	Return the magnitude of a ComplexBF number
arg	Return the angle of the polar representation of the ComplexBF number.
conj	Return a new Conjugated ComplexBF object.
imag	return the Imaginary part of the ComplexBF number.
negate	Return a new Negated ComplexBF Object.
norm	Return the norm of the ComplexBF number.
real	Return the real part of the ComplexBF number.
toExponential	Converts a ComplexBF number to a string using exponential notation with the specified number of digits after tge Decimal place.
toFixed	Converts a ComplexBF number to a string that contains a specified number of digits after the decimal place.
toPrecision	Convert a ComplexBF 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 ComplexBF number to a string using a specified radix(base)
valueOf	The primitive real number value of this ComplexBF number object.

Methods

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

Constants

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

Miscellaneous

parseComplexBF()	Parse a ComplexBF float number string

ComplexBF.abs()

Return the absolute magnitude of the ComplexBF number

Synopsis

ComplexBF object.abs()

Returns

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

Example

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

See Also

ComplexBF.norm()

ComplexBF.arg()

Return the angle of the polar representation of the ComplexBF number

Synopsis

ComplexBF object.arg()

Returns

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

Example

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

See Also

ComplexBF.add()

Add two ComplexBF numbers

Synopsis

ComplexBF.add(a,b)

Arguments

a,b	The ComplexBF numbers to be added.

Returns

The result of the ComplexBF addition.

Example

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

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

See Also

ComplexBF.div(), ComplexBF.mul(), ComplexBF.sub()

ComplexBF.acos()

Return the arc cosine of the ComplexBF number

Synopsis

ComplexBF.acos(a)

Returns

Return the arc cosine of the ComplexBF number.

Example

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

See Also

ComplexBF.asin(), ComplexBF.atan()

ComplexBF.acosh()

Return the arc cosine hyperbolic of the ComplexBF number

Synopsis

ComplexBF.acosh(a)

Returns

Return the arc cosine hyperbolic of the ComplexBF number.

Example

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

See Also

ComplexBF.asinh(), ComplexBF.atanh()

ComplexBF.asin()

Return the arcsine of the ComplexBF number

Synopsis

ComplexBF.asin(a)

Returns

Return the arc sine of the ComplexBF number.

Example

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

See Also

ComplexBF.acos(), ComplexBF.atan()

ComplexBF.asinh()

Return the arc sine hyperbolic of the ComplexBF number

Synopsis

ComplexBF.asinh(a)

Returns

Return the arc sine hyperbolic of the ComplexBF number.

Example

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

See Also

ComplexBF.acosh(), ComplexBF.atanh()

ComplexBF.atan()

Return the arctangent of the ComplexBF number

Synopsis

ComplexBF.atan(a)

Returns

Return the arc tangent of the ComplexBF number.

Example

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

See Also

ComplexBF.acos(), ComplexBF.asin()

ComplexBF.atanh()

Return the arc tangent hyperbolic of the ComplexBF number

Synopsis

ComplexBF.atanh(a)

Returns

Return the arc tanh hyperbolic of the ComplexBF number.

Example

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

See Also

ComplexBF.acosh(), ComplexBF.asinh()

ComplexBF.beta()

Return the beta function of the ComplexBF numbers

Synopsis

ComplexBF.beta(x,y)

Returns

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

Example

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

See Also

ComplexBF.gamma(), ComplexBF.lgamma()

ComplexBF.cos()

Return the cosine of the ComplexBF number

Synopsis

ComplexBF.cos(a)

Returns

Return the cosine of the ComplexBF number.

Example

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

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

See Also

ComplexBF.sin(), ComplexBF.tan()

ComplexBF.cosh()

Return the cosine hyperbolic of the ComplexBF number

Synopsis

ComplexBF.cosh(a)

Returns

Return the cosine hyperbolic of the ComplexBF number.

Example

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

See Also

ComplexBF.sinh(), ComplexBF.tanh()

ComplexBF.conj()

Return the conjugated form of the ComplexBF number

Synopsis

ComplexBF object.conj()

Returns

Return the ComplexBF conjugated form.

Example

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

See Also

ComplexBF.negate()

ComplexBF.div()

Divide two ComplexBF numbers

Synopsis

ComplexBF.div(a,b)

Arguments

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

Returns

The result of the ComplexBF division a/b.

Example

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

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

See Also

ComplexBF.add(), ComplexBF.mul(), ComplexBF.sub()

ComplexBF.equal()

Compare two ComplexBF numbers for equality

Synopsis

ComplexBF.equal(a,b)

Arguments

a,b	The ComplexBF numbers to be compare for

Returns

The Boolean value of the equal comparison.

Example

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

if( ComplexBF.equal(z,y))…            // result false
if( ComplexBF.equal(z,z))…            // result true
if( !ComplexBF.equal(z,y))…                  // result true ! to do "notequal" comparison
```

See Also

ComplexBF.exp()

Compute e^x

Synopsis

ComplexBF.exp(x)

Arguments

x	A ComplexBF 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 ComplexBF( 3, 4 );

ComplexBF.exp(z)…           // e^z approximately (-13.128+i5.200)
```

See Also

ComplexBF.log(), ComplexBF.log10(), ComplexBF.pow()

ComplexBF.gamma()

Return the gamma function of the ComplexBF number

Synopsis

ComplexBF.gamma(x)

Returns

Return the gamma function of the ComplexBF number x.

Example

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

See Also

ComplexBF.lgamma(), ComplexBF.beta()

ComplexBF.i

Return ComplexBF i

Synopsis

ComplexBF.i

Returns

The ComplexBF constant i (0+i1).

Example

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

See Also

ComplexBF.one, ComplexBF.zero

ComplexBF.imag()

Return the imaginary part of the ComplexBF number

Synopsis

ComplexBF object.imag()

Returns

Return the imaginary part of the ComplexBF number.

Example

```
var z = new ComplexBF( 3, 4 );
z.imag()                                // result 4
```

See Also

ComplexBF.real()

ComplexBF.lgamma()

Return the logarithm gamma function of the ComplexBF number

Synopsis

ComplexBF.lgamma(x)

Returns

Return the logarithm gamma function of the ComplexBF number x.

Example

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

See Also

ComplexBF.gamma(), ComplexBF.beta()

ComplexBF.log()

Compute the natural logarithm of x

Synopsis

ComplexBF.log(x)

Arguments

x	A ComplexBF numbers not equal to zero

Returns

Return log(x)

Example

var z = new ComplexBF(3, 4);

ComplexBF.log(z)… // log(x) approximately (1.609+i0.927)

See Also

ComplexBF.exp(), ComplexBF.log10(),

ComplexBF.log10()

Compute the base-10 logarithm of x

Synopsis

ComplexBF.log10(x)

Arguments

x	A ComplexBF numbers not equal to zero

Returns

Return log10(x)

Example

var z = new ComplexBF(3, 4);

ComplexBF.log10(z)… // log10(3+i4) approximately (0.699+i0.403)

See Also

ComplexBF.exp(), ComplexBF.log(),

ComplexBF.mul()

Multiply two ComplexBF numbers

Synopsis

ComplexBF.mul(a,b)

Arguments

a,b	The ComplexBF numbers to be multiplied.

Returns

The result of the ComplexBF multiplication.

Example

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

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

See Also

ComplexBF.add(), ComplexBF.div(), ComplexBF.sub()

ComplexBF.negate()

Return the negated ComplexBF number

Synopsis

ComplexBF object.negate()

Returns

Return the negated ComplexBF number.

Example

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

See Also

ComplexBF.conj()

ComplexBF.norm()

Return the norm (square magnitude) of the ComplexBF number

Synopsis

ComplexBF object.norm()

Returns

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

Example

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

See Also

ComplexBF.abs()

ComplexBF.one

Return ComplexBF one

Synopsis

ComplexBF.one

Returns

The ComplexBF constant one (1+i0).

Example

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

See Also

ComplexBF.zero, ComplexBF.i

ComplexBF.polar()

Convert polar coordinates into a ComplexBF number

Synopsis

ComplexBF.polar(mag,arg)

Arguments

mag	Magnitude of ComplexBF number
arg	Angle of ComplexBF number.

Returns

Return the ComplexBF number.

Example

var z = ComplexBF.polar(4, 0.5);

ComplexBF.polar(4,0.5)   // result (3.510+i1.918)

See Also

ComplexBF.pow()

Compute $x^{\wedge y}$

Synopsis

ComplexBF.pow(x,y)

Arguments

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

Returns

X to the power of y. x^y

Example

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

ComplexBF.pow(x,y)…   // x^y approximately (-0.4198+i0.6605)

See Also

ComplexBF.exp()

ComplexBF.real()

Return the real part of the ComplexBF number

Synopsis

ComplexBF object.real()

Returns

Return the real part of the ComplexBF number.

Example

```
var z = new ComplexBF( 3, 4 );
z.real()                        // result 3
```

See Also

ComplexBF.imag()

ComplexBF.sin()

Return the sine of the ComplexBF number

Synopsis

ComplexBF.sin(a)

Returns

Return the sine of the ComplexBF number.

Example

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

See Also

ComplexBF.cos(), ComplexBF.tan()

ComplexBF.sinh()

Return the sine hyperbolic of the ComplexBF number

Synopsis

ComplexBF.sinh(a)

Returns

Return the sine hyperbolic of the ComplexBF number.

Example

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

See Also

ComplexBF.cosh(), ComplexBF.tanh()

ComplexBF.sub()

Subtract two ComplexBF numbers

Synopsis

ComplexBF.sub(a,b)

Arguments

a,b	The ComplexBF numbers to be subtracted.

Returns

The result of the ComplexBF subtraction.

Example

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

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

See Also

ComplexBF.add(), ComplexBF.div(), ComplexBF.mul()

ComplexBF.sqrt()

Compute a ComplexBF square root

Synopsis

ComplexBF.sqrt(x)

Arguments

x	A ComplexBF 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 ComplexBF( 3, 4 );

ComplexBF.sqrt(z)…            //  Result (2+i1)
```

See Also

ComplexBF.tan()

Return the tangent of the ComplexBF number

Synopsis

ComplexBF.tan(a)

Returns

Return the tangent of the ComplexBF number.

Example

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

See Also

ComplexBF.cos(), ComplexBF.sin()

ComplexBF.tanh()

Return the tangent hyperbolic of the ComplexBF number

Synopsis

ComplexBF.tanh(a)

Returns

Return the tanh hyperbolic of the ComplexBF number.

Example

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

See Also

ComplexBF.cosh(), ComplexBF.sinh()

ComplexBF.toExponential()

Format a number using exponential notation

Synopsis

ComplexBF.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 ComplexBF number is always formatted as: (*real_part* ±i *imaginary_part*)

Returns

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

Example

```
var z = new ComplexBF( 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

ComplexBF.toFixed(), ComplexBF.toPrecision(), ComplexBF.toString()

ComplexBF.toFixed()

Format a number using fixed-point notation

Synopsis

ComplexBF.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 ComplexBF number is always formatted as: (*real_part* ±i *imaginary_part*)

Returns

A string representations of the ComplexBF number, that does not used exponential notation and has exactly *digits* digits after the decimal point. The *ComplexBF number* is rounded as necessary, and the fraction part is padded with zeros if necessary so that it has the specified length. If the *ComplexBF number* is greater than 1e+21, this method simple calls *number*.toString() and return a string in exponential notation.

Example

```
var z = new ComplexBF( 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

ComplexBF.toExponential(), ComplexBF.toPrecision(), ComplexBF.toString()

ComplexBF.toPrecision()

Format the significant digits of a ComplexBF number

Synopsis

ComplexBF.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 tyo convert the ComplexBF number to a base-10 value. A ComplexBF number is always formatted as: (*real_part* ±i *imaginary_part*)

Returns

A string representations of the *ComplexBF 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 ComplexBF( 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

ComplexBF.toExponential(), ComplexBF.toFixed(), ComplexBF.toString()

ComplexBF.toString()

Format the significant digits of a ComplexBF number

Synopsis

ComplexBF.toString(radix)

Arguments

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

(*real_part* ±i *imaginary_part*)

Returns

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

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

See Also

ComplexBF.toExponential(), ComplexBF.toFixed(), ComplexBF.toPrecision()

ComplexBF.valueof()

Return the primitive number value

Synopsis

ComplexBF object.valueof()

Returns

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

Example

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

See Also

ComplexBF.zero

Return ComplexBF zero

Synopsis

ComplexBF.zero

Returns

The ComplexBF constant zero (0+i0).

Example

```
var z = ComplexBF.zero;
```

See Also

ComplexBF.one, ComplexBF.i

parseComplexBF()

Convert a string to a ComplexBF number

Synopsis

parseComplexBF(s)

Arguments

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

Returns

parseComplexBF() parses and return a new ComplexBF number contained in s. parseComplexBF() return a ComplexBF NaN number if parsing fails. A ComplexBF 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. parseComplexBF() can also parsed string omitting the leading and trailing parentheses.

Example

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

See Also