JavaScript BigFloat Library

(Arbitrary precision)

Version 2.0

By Henrik Vestermark (hve@hvks.com)

Contents

BigFloat

Support for arbitrary precision floating-point numbers in JavaScript

Constructor

new BigFloat(*value, precision, mode*) // Invoked as a Constructor
BigFloat(*value, precision, mode*) // Invoked as a Conversion

Arguments

value	Optional Floating-point number, BigFloat object or any JavaScript Number or String representing a number.
precision	Optional. The number of significant digits (Base 10) If the precision argument is omitted, it is treated as BigFloat.precision, which has a default value of 20 digits. If there are no arguments, it is treated as a BigFloat.zero If *BigFloat* is invoked as a conversion, the value parameter is converted to a *BigFloat number* and returned.
mode	Optional rounding mode. If omitted the default rounding mode is taken from BigFloat.rounding which has a default value of "Near";

Returns

Returns a BigFloat object initialize with the floating-point value. If *BigFloat* is invoked as a conversion, the *value* parameter is converted to a *BigFloat number* and returned. If *value* is another BigFloat number, then that is returned. If *value* is undefined a *BigFloat.zero* is returned.
Also note that precision is an optional argument, if omitted the default property BigFloat.precision is used unless the value is a BigFloat object in which case it inherits the precision from the BigFloat object.
Regardless if invoked as a new constructor or as a Conversion BigFloat constructor always return a BigFloat normalized number.

Example:

```
x=New BigFloat(1.5);       // Return a new BigFloat object with precision BigFloat.precision (20);
x=new BigFloat("1.5E2", 43);       // Return a BigFloat object with precision of 43 digits
y=new BigFloat(x);         // Return a new BigFloat object with an inherit precision of 43 digits
y=new BigFloat(x,25);      // Return a new BigFloat object rounded down from 43 to 25 digits

x=BigFloat(1.5);           // return a new BigFloat object with precision BigFloat.precision (20);
x=BigFloat("1.5E2", 43);   // Return a BigFloat object with precision of 43 digits
y=BigFloat(x);             // Return a new BigFloat object with an inherit precision of 43 digits
y=BigFloat(x,25);          // Return a new BigFloat object rounded down from 43 to 25 digits
```

Normalized Numbers.

The Internally format for a BigFloat number is:
<sign> <integer digit>.<fraction digits>E<signed expo Integer>.

is stored as an Object with the following format:

BigFloat object:
mSign: <sign>
sInt: <integer digit as a string>
sFrac: <fraction digits as a string>
mExpo: <signed expo integer>

Example of floating point numbers: +1.0E+0, -1.4567E-123
Notice a normalized number always has a 1-digit integers in front of the fraction sign '.' Moreover, fraction part representing the precision of one or more digits. If the number is an integer the fraction is '0'. The fraction is followed by a 'E' follow by the expo sign and the expo number.
When invoking the BigFloat constructor any number applied is automatically converted to a normalized BigFloat number. In addition, it is guaranteed that after any operations performed a normalized BigFloat number is returned.

It is not the most optimal way of storing the BigFloat number as a string since every +,-,*,/ has to convert the integer and fraction digits into binary number to perform the operations and then back again to the internal string format. However, it simplifies the code considerable at the expense of some performance.

There is two more fields in a BigFloat object and these are:
mPrec: The precision of the number. Which is the number of decimal fraction digits.
mMode: Rounding mode, see below.

Rounding modes:

BigFloat support handling or multiple rounding mode per BigFloat object. The rounding mod comes into play when a new BigFloat number is assigned to a BigFloat object. Depending on the rounding mode the variable is either rounded to nearest ("Near") number of the BigFloat object precision, rounded down ("Down") or rounded up. ("Up"). The rounding mode can be changed as needed using the Property rounding().

Precision:

Each BigFloat object has its own precision (number of fractional digits). The default precision is 20 decimal fractional digits. You can freely mix operations of BigFloat object with difference precision. The operation will always return the maximum precision of the two numbers. E.g.

```
var x5=new BigFloat(1.23456,5);  // create BigFloat with 5 decimal fraction digits
var xdefault=new BigFloat("1.2345678");  // create it with default precision (20digits)
var y= BigFloat.add(xdefault,x5);  // y will be assign with the precision of the operation which is the default precision in this case
x5.assign(y);   // Assign x5 with y rounding to the precision of x5.
```

Since you can't overload the "=" operator in JavaScript you have to use the method BigFloat.assign() to store the result with precision of the left hand side of the "=" operator. Otherwise left hand side precision will be overwritten.

Method

assign()	Assign a BigFloat to another BigFloat object preserving left hand side precision.
changesign	Change the sign of the BigFloat number and return it
exponent	Set or Return the signed exponent of the BigFloat number
exposign	Return the sign of the exponent of the BigFloat number. The sign is returned as a string of either "+" or "-"
fraction	Return fraction part of the BigFloat object.
integer	Return the Integer part of the BigFloat number.
precision	Return the precision of the BigFloat number. Number of fractional digits. If called with an argument, then set the precision with the argument and return the new precision
rounding	Return or set the rounding mode for the BigFloat Object. If called with an argument, it set the rounding mode with the argument and return the new rounding mode.
sign	Return the sign of the BigFloat number. The sign is return as a string of either "+" or "-"
toBigInt()	Convert a BigFloat number to a BigInt number.
toExponential	Converts a BigFloat number to a string using exponential notation with the specified number of digits after the Decimal place.
toFixed	Converts a BigFloat number to a string that contains a specified number of digits after the decimal place.
toInteger	Convert a BigFloat number to a integer number.
toPrecision	Convert a BigFloat 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 BigFloat number to a string using a specified radix(base)
valueOf	The primitive value number of this BigFloat number object.

Functions

abs()	Return the absolute value of a BigFloat number
acos()	Return the BigFloat acos() of the BigFloat argument
acosh()	Return the arc cosine hyperbolic of the BigFloat number
add()	Return the addition of two BigFloat numbers
asin()	Return the BigFloat asin() of the BigFloat argument
acosh()	Return the arc sine hyperbolic of the BigFloat number
atan()	Return the BigFloat atan() of the BigFloat argument
atanh()	Return the arc tangent hyperbolic of the BigFloat number
atan2()	Return the BigFloat atan2 of the quotient of its arguments
ceil()	Round up the BigFloat to nearest integer greater than the number
cos()	Return the BigFloat cos() of the BigFloat argument
cosh()	Return the cosine hyperbolic of the BigFloat number
div()	Return the division of two BigFloat numbers.
E()	Return Euler's number as BigFloat number at the requested priority
equal()	Return the Boolean value (true, false) of the equality of two BigFloat numbers.
exp()	Return the BigFloat power of e.
floor()	Round down the BigFloat to nearest integer less than the number
greater()	Return the Boolean value (true, false) of the Boolean operator > of two BigFloat numbers.
greaterequal()	Return the Boolean value (true, false) of the Boolean operator < =of two BigFloat numbers.
less()	Return the Boolean value (true, false) of the Boolean operator < of two BigFloat numbers.
lessequal()	Return the Boolean value (true, false) of the Boolean operator < =of two BigFloat numbers.
LN2()	Return LN2 as a BigFloat number at the requested priority
LN10()	Return LN10 as a BigFloat number at the requested priority
log()	Return the BigFloat natural logarithm.
log10()	Return the BigFloat base 10 logarithm.
minus(0	return the negative value of the BigFloat number
mod()	Return the modulo of two BigFloat numbers.
mul()	Return the product of two BigFloat numbers.
notequal()	Return the Boolean value (true, false) of the Boolean operator != of two BigFloat numbers.
PI()	Return PI as a BigFloat number at the requested precision
pow()	Return the BigFloat power of x^y.
round()	Round a BigFloat number to a given precision
sin()	Return the BigFloat Sin of the BigFloat argument
sinh()	Return the sine hyperbolic of the BigFloat number
sub()	Return the difference between two BigFloat numbers.
sqrt()	Return the BigFloat square root.
tan()	Return the BigFloat Tan of the BigFloat argument
tanh()	Return the tangent hyperbolic of the BigFloat number

trunc () Returns the integer part of a number by removing any fractional

Constants

zero return a new BigFloat(0) object
one return a new BigFloat(1) object

Miscellaneous

parseBigFloat() Parse a BigFloat number string and return a BigFloat object

API

BigFloat.abs()

Return the absolute value of the BigFloat number

Synopsis

BigFloat object.abs()

Returns

The absolute value of the BigFloat number is returned.

Example

```
var x = new BigFloat( -3.34 );
x.abs()                                  // result 3.34
```

See Also

BigFloat.acos()

Calculate acos() of a BigFloat numbers

Synopsis

BigFloat.acos(x)

Arguments

x The BigFloat numbers to be calculate acos().

Returns

The result of the BigFloat acos(a).

Example

```
var x = new BigFloat( 0.5 );

BigFloat.acos(x)                 // result approx 1.0471975511965976…
```

See Also

BigFloat.asin(), BigFloat.atan(), BigFloat.atan2()

BigFloat.add()

Add two BigFloat numbers

Synopsis

BigFloat.add(a,b)

Arguments

a,b The BigFloat numbers to be added.

Returns

The result of the BigFloat addition. The precision is the largest precision of the two BigFloat Numbers.

Example

```
var x = new BigFloat( 1.2345 );
var y=new  BigFloat(0.98765);

BigFloat.add(x,y)                // result 2.11100  rounded to precisión 5
```

See Also

BigFloat.div(), BigFloat.mul(), BigFloat.sub(), BigFloat.mod()

BigFloat.asin()

Calculate asin() of the BigFloat number

Synopsis

BigFloat.asin(x)

Arguments

x The BigFloat numbers to calculate asin().

Returns

The result of the BigFloat asin().

Example

```
var x = new BigFloat( 0.75 );

BigFloat.asin(x)                // result 0.848062078981481…
```

See Also

BigFloat.acos(), BigFloat.atan(), BigFloat.atan2()

BigFloat.assign()

Assign a new BigFloat number to the variable

Synopsis

BigFloat object.assign(a)

Arguments

a	The BigFloat numbers to assign to the object. A special assign prototype os needed to preserve the precision of the left hand side of the object. Otherwise is just assigning two BigFloat object like a=b; then variable a inherits the precision of the BigFloat b object. With this assign prototype the a precision is maintained and the value of the BigFloat b object is rounded to the precision of the BigFloat an object.

Returns

The result of the BigFloat assign.

Example

```
var x = new BigFloat( 0.75,10 );
var y= new BigFloat(0.25,5);

y.assign(x)          // y is assign the result 0.25 and y precision is maintained at 5
x.assign(y);         // x is assigned the result 0.75 and x precision is maintained at 10
```

See Also

BigFloat.add(), BigFloat.sub(), BigFloat.div(), BigFloat.mul()

BigFloat.atan()

Calculate atan() of the BigFloat number

Synopsis

BigFloat.atan(x)

Arguments

x The BigFloat numbers to calculate atan().

Returns

The result of the BigFloat atan().

Example

var x = new BigFloat(0.75);

BigFloat.atan(x) // result 0.6435011087932844…

See Also

BigFloat.acos(), BigFloat.asin(), BigFloat.atan2()

BigFloat.atan2()

Calculate atan2() of the BigFloat number

Synopsis

BigFloat.atan2(y,x)

Arguments

y The BigFloat number of the Y coordinate of the point
x The BigFloat number of the X coordinate of the point

Returns

The result of the BigFloat atan2() which is a BigFloat number between –π and +π

Description

The BigFloat.atan2() function computes the arc tangent of the ratio y/x. The y argument can be considered the Y coordinate of a point, and the x argument can be considered the X coordinate of the point. Note the unusual order of the arguments to this function. The Y coordinate is passed before the X coordinate.

Example

```
var x=new BigFloat(2), y=new BigFloat(1);
BigFloat.atan2(y,x)              // result 0.463647609000806…
```

See Also

BigFloat.acos(), BigFloat.atan(), BigFloat.asin()

BigFloat.ceil()

Round a BigFloat number up

Synopsis

BigFloat.ceil(x)

Arguments

x The BigFloat numbers to be rounded up to.

Returns

The closed integer greater than or equal to x.

Description

BigFloat.ceil() computes the ceiling function. i.e. it returns the closest integer value that is greater than or equal to the function argument. BigFloat.ceil() differs from BigFloat.round() in that it always round up , rather than rounding up or down to the closest integer. Also note that BigFloat.ceil() does not round negative number to a large negative number ; it rounds them up towards zero.

Example

```
var x = new BigFloat( 1.09);

BigFloat.ceil(x)                // result 2
```

See Also

BigFloat.floor(), BigFloat.round(), BigFloat.trunc()

BigFloat.cos()

Calculate cos() of the BigFloat number

Synopsis

```
BigFloat.cos(x)
```

Arguments

x	The BigFloat numbers to calculate cos().

Returns

The result of the BigFloat cos().

Example

```
var x = new BigFloat( 0.75 );

BigFloat.cos(x)                 // result 0.731688868738209…
```

See Also

BigFloat.sin(), BigFloat.tan()

BigFloat.div()

Divide two BigFloat numbers

Synopsis

BigFloat.div(a,b)

Arguments

a,b The BigFloat numbers to be divided.

Returns

The result of the BigFloat division a/b.

Example

```
var x = new BigFloat( 3E+3 );
var y=new  BigFloat(2);

BigFloat.div(x,y)              // result (+1.5E+3)
```

See Also

BigFloat.add(), BigFloat.mul(), BigFloat.sub(), BigFloat.mod()

BigFloat.E(precision)

Calculate exp(1) to the requested precision. This is the base of the natural logarithm.

Synopsis

BigFloat.E(precision)

Arguments

precision The optional required decimal precision. If omitted the default precision of BigFloat.precision will be used.

Returns

The result of the BigFloat exp(1) to the required precision.

Example

```
BigFloat.E(10)              // result 2.7182818285 (10 decimal precision)
```

See Also

BigFloat.LN2(), BigFloat.LN10(), BigFloat.PI()

BigFloat.equal()

Compare two BigFloat numbers for equality

Synopsis

BigFloat.equal(a,b)

Arguments

a,b The BigFloat numbers to be compare for

Returns

The Boolean value of the equal comparison.

Example

```
var x = new BigFloat( 3.5);
var y = new BigFloat(1.2);

BigFloat.equal(x,y)        // result false
BigFloat.equal(x,x)        // result true
BigFloat.equal(x,y)        // result true ! to do “notequal” comparison
```

See Also
BigFloat.notequal(), BigFloat.less(), BigFloat.lessequal(), BigFloat.greater(), BigFloat.greaterequal()

BigFloat.exp()

Compute e^x

Synopsis

BigFloat.exp(x)

Arguments

x A BigFloat 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 x = new BigFloat( 2.5 );

BigFloat.exp(x)…          // e^x approximately (12.182493960703473…)
```

See Also

BigFloat.log(), BigFloat.log10(), BigFloat.pow()

BigFloat.exponent()

Return or set exponent of the BigFloat number

Synopsis

BigFloat object.exponent(e)

Arguments

e Optional. If call with an argument, then the BigFloat object will be set to this new exponent.

Returns

Return the exponent of the BigFloat number

Example

```
var x = new BigFloat( 1.2345E-6 );
x.exponent();          // result -6
x.exponent(3);         // x will be set to this new number 1.2345E+3
```

See Also

BigFloat.exposign(), BigFloat.fraction(), BigFloat.integer(), BigFloat.precision(), BigFloat.sign(), BigFloat.expo()

BigFloat.floor()

Round a BigFloat number down

Synopsis

BigFloat.floor(x)

Arguments

x The BigFloat numbers to be rounded down to.

Returns

The closed integer less than or equal to x.

Description
BigFloat.floor() computes the floor function. i.e. it returns the closest integer value that is less than or equal to the function argument. BigFloat.floor() differs from BigFloat.round() in that it always round down, rather than rounding up or down to the closest integer. Also note that BigFloat.floor() does round negative number to a large negative number ; it rounds them down towards negative Infinity.

Example

```
var x = new BigFloat( 1.09);

BigFloat.floor(x)                // result 1
```

See Also

BigFloat.ceil(), BigFloat.round(), BigFloat.trunc()

BigFloat.fraction()

Return the fraction part of the BigFloat number

Synopsis

BigFloat object.fraction()

Returns

Return the fraction part of the BigFloat number as a integer text string

Example

```
var x = new BigFloat( 1.2345E-6 );
x.fraction()                    // result “2345”
```

See Also

BigFloat.expo(), BigFloat.exposign(), BigFloat.integer(), BigFloat.precision(), BigFloat.sign(), BigFloat,exponent()

BigFloat.greater()

Compare two BigFloat numbers for greater

Synopsis

BigFloat.greater(a,b)

Arguments

a,b The BigFloat numbers to be compare for

Returns

The Boolean value of the greater comparison.

Example

```
var x = new BigFloat( 3.5);
var y = new BigFloat(1.2);

BigFloat.greater(x,y)       // result true
BigFloat.greater(x,x)       // result false
BigFloat.greater(y,x)       // result false
```

See Also

BigFloat.equal(), BigFloat.notequal(), BigFloat.greaterequal(), BigFloat.less(), BigFloat.lessequal()

BigFloat.greaterequal()

Compare two BigFloat numbers for greater or equal

Synopsis

BigFloat.greaterequal(a,b)

Arguments

a,b The BigFloat numbers to be compare for

Returns

The Boolean value of the equal comparison.

Example

```
var x = new BigFloat( 3.5);
var y = new BigFloat(1.2);

BigFloat.greaterequal(x,y)      // result false
BigFloat.greaterequal(x,x)      // result true
BigFloat.greaterequal(y,x)      // result false
```

See Also
BigFloat.equal(), BigFloat.notequal(), BigFloat.greater(), BigFloat.less(), BigFloat.lessequal()

BigFloat.integer()

Return the integer part of the BigFloat number

Synopsis

BigFloat object.integer()

Returns

Return the integer part of the BigFloat number as a text string

Example

```
var x = new BigFloat( 1.2345E-6 );
x.integer()                          // result "1"
```

See Also

BigFloat.expo(), BigFloat.exposign(), BigFloat.fraction(), BigFloat.precision(), BigFloat.sign(), BigFloat.exponent()

BigFloat.less()

Compare two BigFloat numbers for less

Synopsis

BigFloat.less(a,b)

Arguments

a,b The BigFloat numbers to be compare for

Returns

The Boolean value of the less comparison.

Example

```
var x = new BigFloat( 3.5);
var y = new BigFloat(1.2);

BigFloat.less(x,y)          // result false
BigFloat.less(x,x)          // result false
BigFloat.less(y,x)          // result true
```

See Also

BigFloat.equal(), BigFloat.notequal(), BigFloat.greater(), BigFloat.greaterequal(), BigFloat.lessequal()

BigFloat.lessequal()

Compare two BigFloat numbers for less or equality

Synopsis

BigFloat.lessequal(a,b)

Arguments

a,b The BigFloat numbers to be compare for

Returns

The Boolean value of the less equal comparison.

Example

```
var x = new BigFloat( 3.5);
var y = new BigFloat(1.2);

BigFloat.lessequal(x,y)          // result false
BigFloat.lessequal(x,x)          // result true
BigFloat.lessequal(y,x)          // result true
```

See Also
BigFloat.equal(), BigFloat.notequal(), BigFloat.greater(), BigFloat.greaterequal(), BigFloat.less()

BigFloat.LN10()

Compute the natural logarithm of 10

Synopsis

BigFloat.LN10(precision)

Arguments

precision A optional precision argument. If omitted the default precision of BigFloat.precision is used.

Returns

Return the natural log(10)

Example

```
BigFloat.LN10(10)          // result is approx. 2.3025850929….
```

See Also

BigFloat.LN2(), BigFloat.log10()

BigFloat.LN2()

Compute the natural logarithm of 2

Synopsis

BigFloat.LN2(precision)

Arguments

precision	A optional precision argument. If omitted the default precision of BigFloat.precision is used.

Returns

Return the natural log(2)

Example

```
BigFloat.LN2(10)           // result is approx. 0.6931471805….
```

See Also

BigFloat.LN2(), BigFloat.log10()

BigFloat.log()

Compute the natural logarithm of x

Synopsis

BigFloat.log(x)

Arguments

x Any numbers greater than zero

Returns

Return log(x)

Example

```
var x = new BigFloat(2.5);
BigFloat.log(x)                // log(2.5) approximately 0.916290732…
```

See Also

BigFloat.exp(), BigFloat.log10(),

BigFloat.log10()

Compute the base-10 logarithm of x

Synopsis

BigFloat.log10(x)

Arguments

x A BigFloat number not equal to zero

Returns

Return log10(x)

Example

```
var z = new BigFloat( 3 );
```

BigFloat.log10(z)… // log10(3) approximately (1.09861228…)

See Also

BigFloat.exp(), BigFloat.log(),

BigFloat.mod()

Modulo of two BigFloat numbers

Synopsis

BigFloat.mod(a,b)

Arguments

a,b The BigFloat numbers to take modulo of.

Returns

The result of the BigFloat modulo a%b.

Example

```
var x = new BigFloat( 18.5 );
var y=new  BigFloat(4.2);

BigFloat.mod(x,y)              // result (+1.7E0)
```

See Also

BigFloat.add(), BigFloat.mul(), BigFloat.sub(), BigFloat.div()

BigFloat.minus()

Return the negative number of the BigFloat argument

Synopsis

BigFloat object.minus()

Returns

Return the negative number.

Example

```
var x = new BigFloat(2.5)
x.minus()                    // result -2.5
```

See Also

BigFloat.sub()

BigFloat.mul()

Multiply two BigFloat numbers

Synopsis

BigFloat.mul(a,b)

Arguments

a,b The BigFloat numbers to be multiplied.

Returns

The result of the BigFloat multiplication.

Example

```
var x = new BigFloat( 3 );
var y=new BigFloat(2.5);

BigFloat.mul(x,y)          // result 7.5
```

See Also

BigFloat.add(), BigFloat.div(), BigFloat.sub(), BigFloat.mod()

BigFloat.notequal()

Compare two BigFloat numbers for not equal

Synopsis

BigFloat.notequal(a,b)

Arguments

a,b The BigFloat numbers to be compare for

Returns

The Boolean value of the not equal comparison.

Example

```
var x = new BigFloat( 3.5);
var y = new BigFloat(1.2);

BigFloat.notequal(x,y)          // result true
BigFloat.notequal(x,x)          // result false
BigFloat.notequal(x,y)          // result true
```

See Also
BigFloat.equal(), BigFloat.less(), BigFloat.lessequal(), BigFloat.greater(), BigFloat.greaterequal()

BigFloat.one

Return the constant BigFloat one

Synopsis

BigFloat.one

Returns

The BigFloat constant one New BigFloat(1)

Example

```
var x = BigFloat.one;          // x=1
```

See Also

BigFloat.zero

BigFloat.PI(precision)

Calculate PI to the requested precision.

Synopsis

BigFloat.PI(precision)

Arguments

precision	The optional required decimal precision. If omitted the default precision of BigFloat.precision will be used.

Returns

The result of the BigFloat PI to the required precision.

Example

```
BigFloat.PI(10)          // result 3.14159265359 (10 decimal precision)
```

See Also

BigFloat.LN2(), BigFloat.LN10(), BigFloat.E ()

BigFloat.pow()

Compute $x^{\wedge y}$

Synopsis

BigFloat.pow(x,y)

Arguments

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

Returns

X to the power of y. x^y

Example

```
var x = new BigFloat(2.5);
var y = new BigFloat(3.5)

BigFloat.pow(x,y)…          // x^y approximately 24.7052942201…
```

See Also

BigFloat.exp(), BigFloat.log()

BigFloat.precision()

Set or Return the precision of the BigFloat number

Synopsis

BigFloat object.precision(setPrecision)

Arguments

setPrecision if call with the optional argument setPrecision it set the new precision to setPrecision and return it. If called without an argument the current precision of the BigFloat number is returned

Returns

Return the precision of the BigFloat number

Example

```
var x = new BigFloat( 1.2345E-6,10 );
x.precision();          // result 10
x.precision(12);        // change the precision from 10 to 12 and return the new precision.
```

See Also

BigFloat.expo(), BigFloat.exposign(), BigFloat.fraction(),
BigFloat.integer(),BigFloat.sign(), BigFloat.rounding()

BigFloat.round()

Round a BigFloat number to nearest integer

Synopsis

BigFloat.round(x)

Arguments

x The BigFloat numbers to be rounded up to.

Returns

The closed integer to x.

Description
BigFloat.round() round is number up or down to the nearest integer.
It round 0.5 up

Example

```
var x = new BigFloat( 1.09);
var y=new BigFloat(-2.5);

BigFloat.round(x) ;          // result 1
BigFloat.round(y);           // result -2
```

See Also

BigFloat.floor(), BigFloat.ceil(), BigFloat.trunc()

BigFloat.rounding()

Return or set the rounding mode for the BigFloat number

Synopsis

BigFloat object.rounding(setRoundingMode)

Arguments

setRoundingMode if call with the optional argument set RoundingMode it set the new rounding mode to setRoundingMode and return it. If called without an argument the current rounding mode of the BigFloat number is returned

Returns

Return the rounding mode of the BigFloat number

Example

```
var x = new BigFloat( 1.2345E-6,10 );
x.rounding();              // result “Near”
x.precision(“Up”);      // change the rounding mode from “Near” to “Up” and return the new rounding mode.
```

See Also

BigFloat.expo(), BigFloat.exposign(), BigFloat.fraction(), BigFloat.integer(),BigFloat.sign(), BigFloat.precision()

BigFloat.sign()

Return the sign of the BigFloat number

Synopsis

BigFloat object.sign()

Returns

Return the sign of the BigFloat number as a text string.

Example

```
var z = new BigFloat( 1.2345E-6 );
z.sign()                                    // result “+”
```

See Also

BigFloat.expo(), BigFloat.exponent(), BigFloat.exposign(), BigFloat.fraction(), BigFloat.integer(), BigFloat.precision()

BigFloat.sin()

Calculate sin() of the BigFloat number

Synopsis

BigFloat.sin(x)

Arguments

x The BigFloat numbers to calculate sin().

Returns

The result of the BigFloat sin().

Example

var x = new BigFloat(0.75);

BigFloat.sin(x) // result 1.30895956…E-2

See Also

BigFloat.cos(), BigFloat.tan()

BigFloat.sub()

Subtract two BigFloat numbers

Synopsis

BigFloat.sub(a,b)

Arguments

a,b The BigFloat numbers to be subtracted.

Returns

The result of the BigFloat subtraction.

Example

```
var z = new BigFloat( 1.2345 );
var y=new  BigFloat(0.98765);

BigFloat.sub(z,y)              // result  0.25800 rounded to precision 5
```

See Also

BigFloat.add(), BigFloat.div(), BigFloat.mul(), BigFloat.mod()

BigFloat.sqrt()

Compute the BigFloat square root

Synopsis

BigFloat.sqrt(x)

Arguments

x A BigFloat numbers to be square rooted.

Returns

The square root of x.

Example

```
var x = new BigFloat( 2 );

BigFloat.sqrt(x)               //  Result 1.41421356237…
```

See Also

BigFloat.tan()

Calculate tan() of the BigFloat number

Synopsis

BigFloat.tan(x)

Arguments

x The BigFloat numbers to calculate tan().

Returns

The result of the BigFloat tan().

Example

```
var x = new BigFloat( 0.75 );

BigFloat.tan(x)                 // result 1.30907171…E-2
```

See Also

BigFloat.cos(), BigFloat.sin()

BigFloat.toBigInt()

Convert a BigFloat number to a BigInt integer number

Synopsis

BigFloat.toBigInt()

Returns

A new BigInt number whose value is the integer value of the BigFloat number.

Example

```
var z = new BigFloat( 12345.6789);
z=z.toBigInt();                 // result 12345n
z = new BigFloat( 12345.6789E-3);
z=z.toBigInt();                 // result 12n
z = new BigFloat( 12345.6789E-3);
z=z.toBigInt();                 // result 1234678n
```

See Also

BigFloat.toExponential(), BigFloat.toPrecision(), BigFloat.toFixed(), BigFloat.toInteger()

BigFloat.toExponential()

Format a number using exponential notation

Synopsis

BigFloat.toExponential(digits)

Arguments

Digits The number of digits that will appear after the decimal point. This may be a value between 0 and up. If this argument is omitted, as many digits as necessary will be used.

Returns

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

Example

```
var z = new BigFloat( 12345.6789);
z.toExponential(1);          // result 1.2e+4
z.toExponential(5);          // result 1.23457e+4
z.toExponential(10);         // result 1.23456789000e+4
z.toExponential();           // result 1.23456789e+4
```

See Also

BigFloat.toFixed(), BigFloat.toPrecision(), BigFloat.toString()

BigFloat.toFixed()

Format a number using fixed-point notation

Synopsis

BigFloat.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.

Returns

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

Example

```
var z = new BigFloat( 12345.6789);
z.toFixed(5);          // result 12345.7
z.toFixed(6);          // result 12345.678900
z.toFixed();           // result 12346
```

See Also

BigFloat.toExponential(), BigFloat.toPrecision(), BigFloat.toString()

BigFloat.toInteger()

Convert a BigFloat number to an integer number

Synopsis

BigFloat.toInteger()

Returns

A new BigFloat number whose value is the integer value of the BigFloat number.

Example

```
var z = new BigFloat( 12345.6789);
z=z.toInteger();                // result 12345
z = new BigFloat( 12345.6789E-3);
```

```
z=z.toInteger();                // result 12
z = new BigFloat( 12345.6789E-3);
z=z.toInteger();                // result 1234678
```

See Also

BigFloat.toExponential(), BigFloat.toPrecision(), BigFloat.toFixed(), BigFloat.toBigInt()

BigFloat.toPrecision()

Format the significant digits of a BigFloat number

Synopsis

BigFloat.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.

Returns

A string representation of the *BigFloat 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 BigFloat( 12345.6789);
z.toPrecision(1);               // result  1e+4
z.toPrecision(3);               // result 1.23e+4
z.toPrecision(5);               // result 12346
```

See Also

BigFloat.toExponential(), BigFloat.toFixed(), BigFloat.toString()

BigFloat.toString()

Format the significant digits of a BigFloat number

Synopsis

BigFloat.toString(radix)

Arguments

Radix If omitted the base 10 will be used to convert the BigFloat number to a string. Otherwise the radix will be used (2..36).

Returns

A string representation of the *BigFloat number*, in the indicated radix, returned as a normalized number.

Example

```
var z = new BigFloat( 12345.6789);
z.toString();              // result +1.2346789E+3
```

See Also

BigFloat.toExponential(), BigFloat.toFixed(), BigFloat.toPrecision()

BigFloat.trunc()

Truncate the integer part of a number by removing any fractional

Synopsis

BigFloat.trunc(x)

Arguments

x The BigFloat numbers to be truncated.

Returns

The integer part of a number by removing any fractional.

Description

BigFloat.trunc() truncate the number by removing the fractional.

Example

```
var x = new BigFloat( 1.09);
var y=new BigFloat(-2.5);

BigFloat.trunc(x) ;          // result 1
BigFloat.trunc(y);           // result -2
```

See Also

BigFloat.floor(), BigFloat.ceil(), BigFloat.round()

BigFloat.valueOf()

Return the primitive number value

Synopsis

BigFloat object.valueof()

Returns

The primitive value of the *BigFloat number* is returned in normalized form.

Example

```
var z = new BigFloat( 3 );
z.valueOf()          // return +3.0E+0
```

See Also

BigFloat.zero

Return BigFloat zero

Synopsis

BigFloat.zero

Returns

The BigFloat constant zero +0.0E+0 in the internal normalized form

Example

```
var z = BigFloat.zero;
```

See Also

BigFloat.one

parseBigFloat()

Convert a string to a BigFloat number

Synopsis

parseBigFloat(s)

Arguments

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

Returns

parseBigFloat() parses and return a new BigFloat number contained in s. parseBigFloat() return a BigFloat NaN number if parsing fails. A BigFloat number followed the standard syntax for a floating point number as for regular Numbers in JavaScript

Example

```
var z = parseBigFloat( “1.2”);          // result 1.2
z = parseBigFloat( 12.5 );              // result 1.2 E+1
z = parseBigFloat( -1.2E-5” );          // result -1.2E-5
```

See Also