

# JAVA.LANG.STRICTMATH.ATAN2 METHOD

[http://www.tutorialspoint.com/java/lang/strictmath\\_atan2.htm](http://www.tutorialspoint.com/java/lang/strictmath_atan2.htm)

Copyright © tutorialspoint.com

## Description

The **java.lang.StrictMath.atan2** method computes the phase theta by computing an arc tangent of **y/x** in the range of **-pi to pi**. It returns the angle theta from the conversion of rectangular coordinates **x, y** to polar coordinates **r, theta**.

- If the first argument is positive zero and the second argument is positive, or the first argument is positive and finite and the second argument is positive infinity, then the result is positive zero.
- If the first argument is negative zero and the second argument is positive, or the first argument is negative and finite and the second argument is positive infinity, then the result is negative zero.
- If the first argument is positive zero and the second argument is negative, or the first argument is positive and finite and the second argument is negative infinity, then the result is the *double* value closest to *pi*.
- If the first argument is negative zero and the second argument is negative, or the first argument is negative and finite and the second argument is negative infinity, then the result is the *double* value closest to *-pi*.
- If the first argument is positive and the second argument is positive zero or negative zero, or the first argument is positive infinity and the second argument is finite, then the result is the *double* value closest to *pi/2*.
- If the first argument is negative and the second argument is positive zero or negative zero, or the first argument is negative infinity and the second argument is finite, then the result is the *double* value closest to *-pi/2*.
- If both arguments are positive infinity, then the result is the *double* value closest to *pi/4*.
- If the first argument is positive infinity and the second argument is negative infinity, then the result is the *double* value closest to *3\*pi/4*.
- If the first argument is negative infinity and the second argument is positive infinity, then the result is the *double* value closest to *-pi/4*.
- If both arguments are negative infinity, then the result is the *double* value closest to *-3\*pi/4*.

## Declaration

Following is the declaration for **java.lang.StrictMath.atan2** method

```
public static double atan2(double y, double x)
```

## Parameters

- **y** -- This is the ordinate coordinate.
- **x** -- This is the abscissa coordinate.

## Return Value

This method returns the theta component of the point *r, theta* in polar coordinates that corresponds to the point *x, y* in Cartesian coordinates.

## Exception

- **NA**

## Example

The following example shows the usage of `java.lang.StrictMath.atan2` method.

```
package com.tutorialspoint;

import java.lang.*;

public class StrictMathDemo {

    public static void main(String[] args) {

        double d1 = 0.6 , d2 = 90.00;

        /* returns the theta component of the point (r, theta) in
        polar coordinates that corresponds to the point (x, y)
        in Cartesian coordinates */

        double dAbsValue = StrictMath.atan2(d1, d2);
        System.out.println("arc tangent value after conversion = " + dAbsValue);

        dAbsValue = StrictMath.atan2(d2 , d1);
        System.out.println("arc tangent value after conversion = " + dAbsValue);
    }
}
```

Let us compile and run the above program, this will produce the following result:

```
arc tangent value after conversion = 0.0066665679038682285
arc tangent value after conversion = 1.5641297588910283
```

Loading [Mathjax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js