**Python Trignometric Functions**

Python includes following functions that perform trigonometric calculations.

|  |  |
| --- | --- |
| Sr.No. | Function & Description |
| 1 | acos(x) :  Return the arc cosine of x, in radians. Description The method **acos()** returns the arc cosine of x, in radians. Syntax Following is the syntax for **acos()** method −  acos(x)  **Note** − This function is not accessible directly, so we need to import math module and then we need to call this function using math static object. Parameters  * **x** − This must be a numeric value in the range -1 to 1. If x is greater than 1 then it will generate an error.  Return Value This method returns arc cosine of x, in radians. Example The following example shows the usage of acos() method.  #!/usr/bin/python  import math  print "acos(0.64) : ", math.acos(0.64)  print "acos(0) : ", math.acos(0)  print "acos(-1) : ", math.acos(-1)  print "acos(1) : ", math.acos(1)  When we run above program, it produces following result −  acos(0.64) : 0.876298061168  acos(0) : 1.57079632679  acos(-1) : 3.14159265359  acos(1) : 0.0 |
| 2 | asin(x) :  Return the arc sine of x, in radians. Description The method **asin()** returns the arc sine of x, in radians. Syntax Following is the syntax for **asin()** method −  asin(x)  **Note** − This function is not accessible directly, so we need to import math module and then we need to call this function using math static object. Parameters  * **x** − This must be a numeric value in the range -1 to 1. If x is greater than 1 then it will generate an error.  Return Value This method returns arc sine of x, in radians. Example The following example shows the usage of asin() method.  #!/usr/bin/python  import math  print "asin(0.64) : ", math.asin(0.64)  print "asin(0) : ", math.asin(0)  print "asin(-1) : ", math.asin(-1)  print "asin(1) : ", math.asin(1)  When we run above program, it produces following result −  asin(0.64) : 0.694498265627  asin(0) : 0.0  asin(-1) : -1.57079632679  asin(1) : 1.57079632679 |
| 3 | atan(x) :  Return the arc tangent of x, in radians. Description The method **atan()** returns the arc tangent of x, in radians. Syntax Following is the syntax for **atan()** method −  atan(x)  **Note** − This function is not accessible directly, so we need to import math module and then we need to call this function using math static object. Parameters  * **x** − This must be a numeric value.  Return Value This method returns arc tangent of x, in radians. Example The following example shows the usage of atan() method.  #!/usr/bin/python  import math  print "atan(0.64) : ", math.atan(0.64)  print "atan(0) : ", math.atan(0)  print "atan(10) : ", math.atan(10)  print "atan(-1) : ", math.atan(-1)  print "atan(1) : ", math.atan(1)  When we run above program, it produces following result −  atan(0.64) : 0.569313191101  atan(0) : 0.0  atan(10) : 1.4711276743  atan(-1) : -0.785398163397  atan(1) : 0.785398163397 |
| 4 | Atan2( y, x) :  Return atan(y / x), in radians. Description The method **atan2()** returns atan(y / x), in radians. Syntax Following is the syntax for **atan2()** method −  atan2(y, x)  **Note** − This function is not accessible directly, so we need to import math module and then we need to call this function using math static object. Parameters  * **y** − This must be a numeric value. * **x** − This must be a numeric value.  Return Value This method returns atan(y / x), in radians. Example The following example shows the usage of atan2() method.  #!/usr/bin/python  import math  print "atan2(-0.50,-0.50) : ", math.atan2(-0.50,-0.50)  print "atan2(0.50,0.50) : ", math.atan2(0.50,0.50)  print "atan2(5,5) : ", math.atan2(5,5)  print "atan2(-10,10) : ", math.atan2(-10,10)  print "atan2(10,20) : ", math.atan2(10,20)  When we run above program, it produces following result −  atan2(-0.50,-0.50) : -2.35619449019  atan2(0.50,0.50) : 0.785398163397  atan2(5,5) : 0.785398163397  atan2(-10,10) : -0.785398163397  atan2(10,20) : 0.463647609001 |
| 5 | cos(x) :  Return the cosine of x radians. Description The method **cos()** returns the cosine of x radians. Syntax Following is the syntax for **cos()** method −  cos(x)  **Note** − This function is not accessible directly, so we need to import math module and then we need to call this function using math static object. Parameters  * **x** − This must be a numeric value.  Return Value This method returns a numeric value between -1 and 1, which represents the cosine of the angle. Example The following example shows the usage of cos() method.  #!/usr/bin/python  import math  print "cos(3) : ", math.cos(3)  print "cos(-3) : ", math.cos(-3)  print "cos(0) : ", math.cos(0)  print "cos(math.pi) : ", math.cos(math.pi)  print "cos(2\*math.pi) : ", math.cos(2\*math.pi)  When we run above program, it produces following result −  cos(3) : -0.9899924966  cos(-3) : -0.9899924966  cos(0) : 1.0  cos(math.pi) : -1.0  cos(2\*math.pi) : 1.0 |
| 6 | Hypot ( x, y) :  Return the Euclidean norm, sqrt(x\*x + y\*y). Description The method **hypot()** return the Euclidean norm, sqrt(x\*x + y\*y). Syntax Following is the syntax for **hypot()** method −  hypot(x, y)  **Note** − This function is not accessible directly, so we need to import math module and then we need to call this function using math static object. Parameters  * **x** − This must be a numeric value. * **y** − This must be a numeric value.  Return Value This method returns Euclidean norm, sqrt(x\*x + y\*y). Example The following example shows the usage of hypot() method.  #!/usr/bin/python  import math  print "hypot(3, 2) : ", math.hypot(3, 2)  print "hypot(-3, 3) : ", math.hypot(-3, 3)  print "hypot(0, 2) : ", math.hypot(0, 2)  When we run above program, it produces following result −  hypot(3, 2) : 3.60555127546  hypot(-3, 3) : 4.24264068712  hypot(0, 2) : 2.0 |
| 7 | sin(x) :  Return the sine of x radians. Description The method **sin()** returns the sine of x, in radians. Syntax Following is the syntax for **sin()** method −  sin(x)  **Note** − This function is not accessible directly, so we need to import math module and then we need to call this function using math static object. Parameters  * **x** − This must be a numeric value.  Return Value This method returns a numeric value between -1 and 1, which represents the sine of the parameter x. Example The following example shows the usage of sin() method.  #!/usr/bin/python  import math  print "sin(3) : ", math.sin(3)  print "sin(-3) : ", math.sin(-3)  print "sin(0) : ", math.sin(0)  print "sin(math.pi) : ", math.sin(math.pi)  print "sin(math.pi/2) : ", math.sin(math.pi/2)  When we run above program, it produces following result −  sin(3) : 0.14112000806  sin(-3) : -0.14112000806  sin(0) : 0.0  sin(math.pi) : 1.22464679915e-16  sin(math.pi/2) : 1.0 |
| 8 | tan(x) :  Return the tangent of x radians. Description The method **tan()** returns the tangent of x radians. Syntax Following is the syntax for **tan()** method −  tan(x)  **Note** − This function is not accessible directly, so we need to import math module and then we need to call this function using math static object. Parameters  * **x** − This must be a numeric value.  Return Value This method returns a numeric value between -1 and 1, which represents the tangent of the parameter x. Example The following example shows the usage of tan() method.  #!/usr/bin/python  import math  print "tan(3) : ", math.tan(3)  print "tan(-3) : ", math.tan(-3)  print "tan(0) : ", math.tan(0)  print "tan(math.pi) : ", math.tan(math.pi)  print "tan(math.pi/2) : ", math.tan(math.pi/2)  print "tan(math.pi/4) : ", math.tan(math.pi/4)  When we run above program, it produces following result −  tan(3) : -0.142546543074  tan(-3) : 0.142546543074  tan(0) : 0.0  tan(math.pi) : -1.22460635382e-16  tan(math.pi/2) : 1.63317787284e+16  tan(math.pi/4) : 1.0 |
| 9 | Degrees (x) :  Converts angle x from radians to degrees. Description The method **degrees()** converts angle x from radians to degrees. Syntax Following is the syntax for **degrees()** method −  degrees(x)  **Note** − This function is not accessible directly, so we need to import math module and then we need to call this function using math static object. Parameters  * **x** − This must be a numeric value.  Return Value This method returns degree value of an angle. Example The following example shows the usage of degrees() method.  #!/usr/bin/python  import math  print "degrees(3) : ", math.degrees(3)  print "degrees(-3) : ", math.degrees(-3)  print "degrees(0) : ", math.degrees(0)  print "degrees(math.pi) : ", math.degrees(math.pi)  print "degrees(math.pi/2) : ", math.degrees(math.pi/2)  print "degrees(math.pi/4) : ", math.degrees(math.pi/4)  When we run above program, it produces following result −  degrees(3) : 171.887338539  degrees(-3) : -171.887338539  degrees(0) : 0.0  degrees(math.pi) : 180.0  degrees(math.pi/2) : 90.0  degrees(math.pi/4) : 45.0 |
| 10 | Radians (x) :  Converts angle x from degrees to radians. Description The method **radians()** converts angle x from degrees to radians. Syntax Following is the syntax for **radians()** method −  radians(x)  **Note** − This function is not accessible directly, so we need to import math module and then we need to call this function using math static object. Parameters  * **x** − This must be a numeric value.  Return Value This method returns radian value of an angle. Example The following example shows the usage of radians() method.  #!/usr/bin/python  import math  print "radians(3) : ", math.radians(3)  print "radians(-3) : ", math.radians(-3)  print "radians(0) : ", math.radians(0)  print "radians(math.pi) : ", math.radians(math.pi)  print "radians(math.pi/2) : ", math.radians(math.pi/2)  print "radians(math.pi/4) : ", math.radians(math.pi/4)  When we run above program, it produces following result −  radians(3) : 0.0523598775598  radians(-3) : -0.0523598775598  radians(0) : 0.0  radians(math.pi) : 0.0548311355616  radians(math.pi/2) : 0.0274155677808  radians(math.pi/4) : 0.0137077838904 |