



EXPLORER

OPEN EDITORS

- User Settings C:\Users\pcurti...
- percipio06\_math\_functions....

PYTHON

- Automate-Boring-Stuff
- my\_code
- Percipio\_Python3-Course
  - 01\_Start
  - 02\_Data-Sequence Types
    - percipio04\_int\_types.py
    - percipio05\_float\_type.py
    - percipio06\_math\_functions....
    - percipio07\_boolean\_type.py
    - percipio08\_Strings.py
    - percipio09\_float\_type.py
    - percipio10\_bytes\_type.py
    - percipio11\_bytearray\_type.py
    - percipio12\_list\_type.py
    - percipio13\_tuple\_type.py
    - percipio14\_slice\_type.py
    - percipio14a\_list\_copy\_boole..
  - 03\_Collections-Mapping-Loo...
  - 04\_Modules-Functions
  - 05\_Classes
  - 06\_Working-with-Files
  - 07\_Comprehensions
  - 08\_Iterables-and-Generators
  - 09\_Exceptions
- Python Projects\_2014
- CMD\_Python\_Set-Path.txt
- Python\_Basics.txt
- Python\_Clear-Window-Comman..
- python\_exercises\_00.py
- python\_exercises\_01.py
- Python\_Tutorial\_Running-Script...
- Python\_Tutorials.md

User Settings percipio06\_math\_functions.py x

```
1 '''
2 percipio06_math_functions.py
3 Percipio video: Data & Sequence Types
4 Most useful math module functions
5 '''
6 x,y = 5.0, 10
7 print('x =', x, 'x type is: ', type(x))
8 print('y =', y, 'y type is: ', type(y))
9 import math
10 # math module provides two constants
11 pi = math.pi # 3.14
12 e = math.e # WHAT IS e?
13 print('Value of Pi is: ', pi) #
14 print('The rounded value of Pi is: ', round(pi, 4)) #
15 # The float class allows creation of special numbers
16 pos_inf = float('inf') # infinity
17 neg_inf = float('-inf') # negative infinity
18 not_a_num = float('nan') # not a number
19 # The math module provides functions to detect these numbers
20 print('math.isinf(pos_inf) = ', math.isinf(pos_inf)) # detect positive infinity
21 print('math.isinf(neg_inf) = ', math.isinf(neg_inf)) # detect negative infinity
22 print('math.isnan(not_a_num) = ', math.isnan(not_a_num)) # detect not-a-number
23 # Beware these special numbers propagate with errors
24 print('pos_inf * x = ', pos_inf * x) # results in positive infinity
25 print('neg_inf / y = ', neg_inf / y) # results in negative infinity
26 print('pos_inf + neg_inf = ', pos_inf + neg_inf) # yeilds not-a-number
27 print('not_a_num - y = ', not_a_num - y) # any operation with nan generally results in nan
28 # A nan value is never equal to another nan value, even itself
29 print('not_a_num == not_a_num = ', not_a_num == not_a_num) # results false
30 # The math module provides many other functions
31 print('math.factorial (5) =', math.factorial(5)) # shows number theory
32 # Logarithmic and power functions
33 print('math.log(x) = ', math.log(x)) #
34 print('math.log10 (x) = ', math.log10 (x)) #
35 print('math.exp (x) = ', math.exp (x)) #
36 print('math.pow(x, x) = ', math.pow(x, x)) # power of
37 print('math.sqrt(25) = ', math.sqrt(25)) # squareroot
```







EXPLORER

{} User Settings percipio06\_math\_functions.py x

OPEN EDITORS

- {} User Settings C:\Users\pcurti...
- percipio06\_math\_functions....

PYTHON

- Automate-Boring-Stuff
- my\_code
- Percipio\_Python3-Course
  - 01\_Start
  - 02\_Data-Sequence Types
    - percipio04\_int\_types.py
    - percipio05\_float\_type.py
    - percipio06\_math\_functions....
    - percipio07\_boolean\_type.py
    - percipio08\_Strings.py
    - percipio09\_float\_type.py
    - percipio10\_bytes\_type.py
    - percipio11\_bytearray\_type.py
    - percipio12\_list\_type.py
    - percipio13\_tuple\_type.py
    - percipio14\_slice\_type.py
    - percipio14a\_list\_copy\_boole..
  - 03\_Collections-Mapping-Loo...
  - 04\_Modules-Functions
  - 05\_Classes
  - 06\_Working-with-Files
  - 07\_Comprehensions
  - 08\_Iterables-and-Generators
  - 09\_Exceptions
- Python Projects\_2014
- CMD\_Python\_Set-Path.txt
- Python\_Basics.txt
- Python\_Clear-Window-Comman..
- python\_exercises\_00.py
- python\_exercises\_01.py
- Python\_Tutorial\_Running-Script...
- Python\_Tutorials.md

```
37 print('math.sqrt(25) = ', math.sqrt(25)) # squareroot
38 # trigonometric functions
39 print('math.cos(x) =', math.cos(x)) #
40 print('math.acos(0.284) =', math.acos(0.284)) #
41 # angular conversion functions
42 print('math.degrees(x) =', math.degrees(x)) #
43 print('math.radians(286.5) =', math.radians(286.5)) #
44 # hyperbolic functions
45 print('math.acosh(x) =', math.acosh(x)) #
46 print('math.asinh(x) =', math.asinh(x)) #
47 '''
48 RESULT:
49 x = 5.0 x type is: <class 'float'>
50 y = 10 y type is: <class 'int'>
51 Value of Pi is: 3.141592653589793
52 The rounded value of Pi is: 3.1416
53 math.isinf(pos_inf) = True
54 math.isinf(neg_inf) = True
55 math.isnan(not_a_num) = True
56 pos_inf * x = inf
57 neg_inf / y = -inf
58 pos_inf + neg_inf = nan
59 not_a_num - y = nan
60 not_a_num == not_a_num = False
61 math.factorial(5) = 120
62 math.log(x) = 1.6094379124341003
63 math.log10(x) = 0.6989700043360189
64 math.exp(x) = 148.4131591025766
65 math.pow(x, x) = 3125.0
66 math.sqrt(25) = 5.0
67 math.cos(x) = 0.28366218546322625
68 math.acos(0.284) = 1.2828330039201725
69 math.degrees(x) = 286.4788975654116
70 math.radians(286.5) = 5.000368306963754
71 math.acosh(x) = 2.2924316695611777
72 math.asinh(x) = 2.3124383412727525
73 '''
```