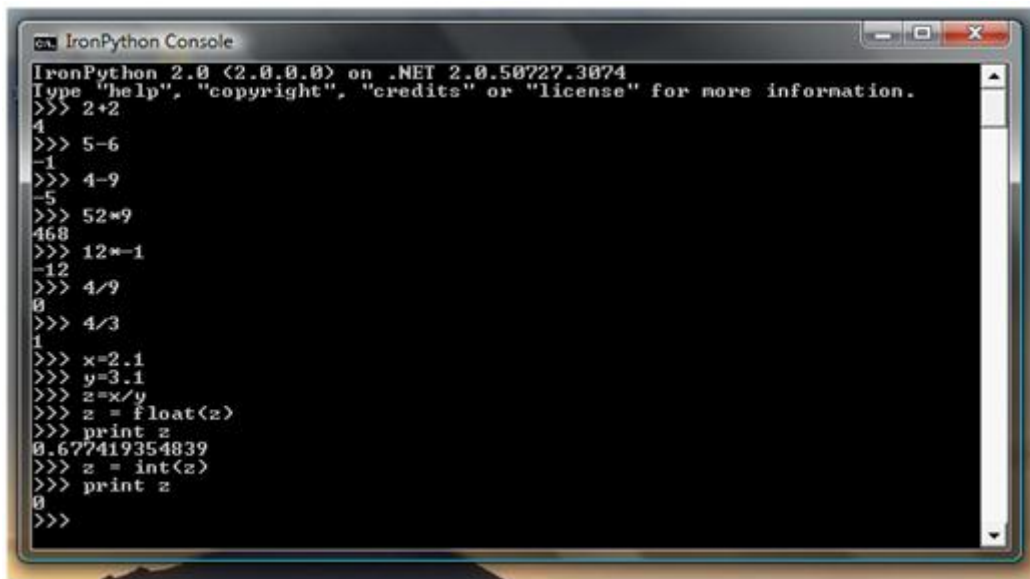


Mathematics in IronPython

In this article we will see Numeric Funtions and calculations in IronPython

Here is 4 basic mathematical operators in IronPython ;

Addition	+
Substraction	-
Division	/
Multiplication	•



```

IronPython 2.0 (2.0.0.0) on .NET 2.0.50727.3074
Type "help", "copyright", "credits" or "license" for more information.
>>> 2+2
4
>>> 5-6
-1
>>> 4-9
-5
>>> 52*9
468
>>> 12*-1
-12
>>> 4/9
0
>>> 4/3
1
>>> x=2.1
>>> y=3.1
>>> z=x/y
>>> z = float(z)
>>> print z
0.677419354839
>>> z = int(z)
>>> print z
0
>>>

```

```
>>> 2+2 #we can use this expressions same as math in daily life
4
```

```
>>> 5-6
-1
```

```
>>> 4-3
1
```

```
>>> 52*9
468
```

```
>>> 12*-1
-12
```

```
>>> 4/3
1 #After division result returns as an integer
```

```
>>> x=2.1
>>> y=3.1
>>> z=x/y
>>> z=float(z) #if we set z as a float, result returns as a float.
>>> print z
```



```

0.677419354839
>>> z=int(z)      #if we set z as an integer, result returns as a integer.
>>> print z
0

>>> x=3
>>> y=8
>>> x=str(x)      #Let's see what happens when add converted intergers as strings
>>> y=str(y)
print x+y
38

```

Because of no pre-declaration, division result returns defaultly as an integer. We can convert this result to other types.

Power of a number

This method returns x to the power of y the value of x^y .

```

4^3 = >>> 4**3
64

4^(2^6) = 4**2**6
>>> 4**2**2
256

```

Complex Numbers

This gives us the form of $x + yj$, real and imaginary numbers. "j" represents the square of (-1)

```

>>> (4+4j)+(2-5j)
(6-1j)
>>> (4+4j)/(2-5j)
(-0.41379310344827586+0.96551724137931039j)

>>> (4+4j)*(2-5j)
(28-12j)

```

We can use 4 basic mathematical operations with complex numbers.

Other Common Numeric Functions

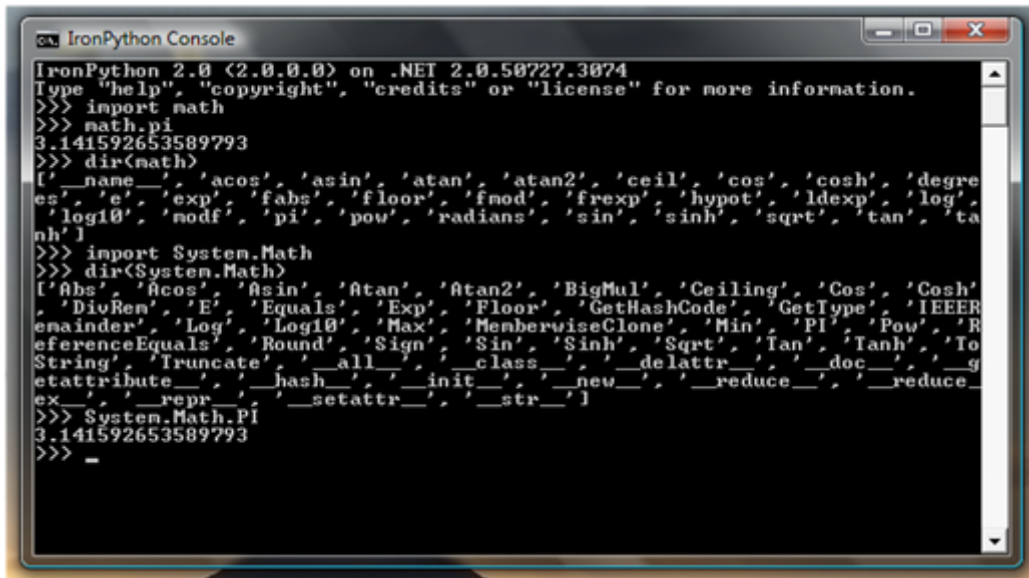
```

>>> import math    # we imported Math library
>>> math.pi        # we call PI in Math library
3.14159265359

```

Now we can use math library of python or math library of .NET with IronPython.





```
IronPython 2.0 (2.0.0.0) on .NET 2.0.50727.3074
Type "help", "copyright", "credits" or "license" for more information.
>>> import math
>>> math.pi
3.141592653589793
>>> dir(math)
['_name_', 'acos', 'asin', 'atan', 'atan2', 'ceil', 'cos', 'cosh', 'degrees', 'e', 'exp', 'fabs', 'floor', 'fmod', 'frexp', 'hypot', 'ldexp', 'log', 'log10', 'modf', 'pi', 'pow', 'radians', 'sin', 'sinh', 'sqrt', 'tan', 'tanh']
>>> import System.Math
>>> dir(System.Math)
['_Abs', '_Acos', '_Asin', '_Atan', '_Atan2', '_BigMul', '_Ceiling', '_Cos', '_Cosh', '_DivRem', '_E', '_Equals', '_Exp', '_Floor', '_GetHashCode', '_GetType', '_IEEERemainder', '_Log', '_Log10', '_Max', '_MemberwiseClone', '_Min', '_PI', '_Pow', '_ReferenceEquals', '_Round', '_Sign', '_Sin', '_Sinh', '_Sqrt', '_Tan', '_Tanh', '_ToString', '_Truncate', '__all__', '__class__', '__delattr__', '__doc__', '__getattribute__', '__hash__', '__init__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__setattr__', '__str__']
>>> System.Math.PI
3.141592653589793
>>> -
```

Abs(), Pow() Functions

Abs function returns absolute value of negative numbers. Pow function returns the value x to the power of y (x^y)

```
>>> abs(-42), 2**4, pow(2, 4)
(42, 16, 16)
```

If you have any questions or discover any errors / typos please let me know ik@ibrahimkivanc.com

All The Best!



| Ibrahim KIVANÇ
| ik@ibrahimkivanc.com
| www.ibrahimkivanc.com