Numeric Functions

Let us understand some of the common numeric functions we use with Python as programming language.

- We have functions even for standard operators such as +, -, *, / under a library called as operator. However, we use operators more often than functions.
  - `add` for `+`
  - `sub` for `-`
  - `mul` for `*`
  - `truediv` for `/`
- We can use `pow` for getting power value.
- We also have `math` library for some advanced mathematical operations.
- Also, we have functions such as `min`, `max` to get minimum and maximum of the numbers passed.

```
4 + 5
```

```
9
```

```
5 % 4
```

```
1
```

```
import operator
from operator import add, sub, mul, truediv
```

```
add?
```

```
Docstring: add(a, b) -- Same as a + b.
Type:      builtin_function_or_method
```

```
add(4, 5)
```

```
9
```

```
truediv(4, 5)
```

```
0.8
```

```
from operator import mod

mod(5, 4)
```

```
1
```

```
pow(2, 3) # This is also available under math library
```

```
8
```

```
pow?
```

```
Signature: pow(x, y, z=None, /)
Docstring:
Equivalent to x**y (with two arguments) or x**y % z (with three arguments)

Some types, such as ints, are able to use a more efficient algorithm when
invoked using the three argument form.
Type:      builtin_function_or_method
```

```
import math
```

```
math.pow?
```

```
Docstring:
pow(x, y)

Return x**y (x to the power of y).
Type:      builtin_function_or_method
```

```
math.ceil(4.4)
```

```
5
```

```
math.floor(4.7)
```

```
4
```

```
round(4.4)
```

```
4
```

```
round(4.7)
```

```
5
```

```
round(4.662, 2)
```

```
4.66
```

```
math.sqrt(2)
```

```
1.4142135623730951
```

```
math.pow(2, 3)
```

```
8.0
```

```
min?
```

```
Docstring:
min(iterable, *[, default=obj, key=func]) -> value
min(arg1, arg2, *args, *[, key=func]) -> value

With a single iterable argument, return its smallest item. The
default keyword-only argument specifies an object to return if
the provided iterable is empty.
With two or more arguments, return the smallest argument.
Type:      builtin_function_or_method
```

```
min(2, 3)
```

```
2
```

```
max(2, 3, 5, 1)
```

```
5
```