```
>>> # Numbers
>>> # int
>>> # flaot
>>> # Decimal - float
>>> # integer - int
>>> int("2")
2
>>> 2 + 2
4
>>> # int div
>>> # quotient
>>> 9 // 3
3
>>> 9 // 2
4
>>> 9 / 2
4.5
>>> 9 / 3
3.0
>>>
>>> # +, *, -
>>> 2.0 * 4
8.0
>>> 2 * 4
8
>>> # +, *, - return value's data type is float if one of the operand is a float
>>> 2 - 3
-1
>>> 3-2
1
>>> -2
-2
>>> x = 8
>>> -x
-8
>>> 2 + 8
10
>>> # without using + operator add 2 numbers
>>> 2 - -8
10
>>> 2 ** 3
8
>>> 6 ** 2
36
```

```
>>> pow(2, 3)
8
>>> 4 % 3
1
>>> 5 %3
2
>>> 6%3
0
>>> pow(2, 3, 2)
0
>>> (2 **3) % 2
0
>>> pow(6, 3, 4)
0
>>> 6 ** 3
216
>>> 216 %4
0
>>> ()
()
>>> (6 ** 3) % 4
0
>>> 1000000000000
1000000000000
>>> 1_000_000 == 1000000
True
>>> 1_000_000
1000000
>>> 1_0000_00
1000000
>>> x = 100
>>> x = 100  # integer literal
>>> 216/4
54.0
>>> 54 * 4
216
>>> y = int("7")
>>> y
7
>>> 34.89
34.89
>>> 100_00.22_33
10000.2233
>>> 100_00._22_33
```

```
SyntaxError: invalid syntax
>>> 100.0.is_integer()
True
>>> 100.88.is_integer()
False
>>> # e notation
>>> 18e3
18000.0
>>> 18 * 1000
18000
>>> 1.8765e2
187.65
>>> 12e900
inf
>>> -12e900
-inf
>>> -12e10
-120000000000.0
>>> -120_000_000_000.0
-120000000000.0
>>> float
<class 'float'>
>>> float("12.6")
12.6
>>> int(12.6)
12
>>> int(12.9999)
12
>>> int("12.6")
Traceback (most recent call last):
  File "<pyshell#66>", line 1, in <module>
    int("12.6")
ValueError: invalid literal for int() with base 10: '12.6'
>>> # "12.6" -> 12
>>> int(float("12.6"))
12
>>> abs(-12)
12
>>> abs(-12.88)
12.88
>>> abs(12.88)
12.88
>>> round(1.99)
2
```

```
>>> round(1.99334,4)
1.9933
>>> round(1.99334,3)
1.993
>>> round(1.99334,1)
2.0
>>> round(1.9936,3)
1.994
>>> round(1.46,1)
1.5
>>> round(1.44,1)
1.4
>>> 0.1
0.1
>>> x = 0.1
>>> x + x + x
0.30000000000000004
>>> round(1.5)
2
>>> round(2.5)
2
>>> round(3.5)
4
>>> round(576.5)
576
>>> round(6.5)
6
>>> n = 123.45
>>> f"{n}"
'123.45'
>>> f"{n:.1f}"
'123.5'
>>> r = 1234.56789
>>> f"{r:.1f}"
'1234.6'
>>> f"{r:.3f}"
'1234.568'
>>> q = 12345678
>>> f'{q:,}'
'12,345,678'
>>> #f'{var_name:.<precision>f}'
>>> #f'{var_name:,}'
>>> q = 12345678.12345
>>> f"{r:,.3f}"
```

```
'1,234.568'
>>> f"{q:,.3f}"
'12,345,678.123'
>>> #f'{var_name:,.<precision>f}'
>>> pow(9, 0.5)  # MAth notation 9 ^ 0.5
3.0
>>> # x ^ 0.5 is a sqrt
>>> int(pow(9, 0.5))
3
>>> sqrt_n = pow(9, 0.5)
>>> sqrt_n
3.0
>>> int(sqrt_n)
3
>>> int(8.6)
8
>>> round(8.6)
9
>>> round(8.51)
9
>>> round(8.4)
8
>>> round(8.5)
8
>>> round(9.5)
10
>>> # Comparison
>>> True
True
>>> False
False
>>> dir("")
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format__',
'__ge__', '__getattribute__', '__getitem__', '__getnewargs__', '__gt__', '__hash__', '__init__',
'__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mod__', '__mul__', '__ne__',
'__new__', '__reduce__', '__reduce_ex__', '__repr__', '__rmod__', '__rmul__', '__setattr__',
'__sizeof__', '__str__', '__subclasshook__', 'capitalize', 'casefold', 'center', 'count', 'encode',
'endswith', 'expandtabs', 'find', 'format', 'format_map', 'index', 'isalnum', 'isalpha', 'isascii',
'isdecimal', 'isdigit', 'isidentifier', 'islower', 'isnumeric', 'isprintable', 'isspace', 'istitle', 'isupper',
'join', 'ljust', 'lower', 'lstrip', 'maketrans', 'partition', 'removeprefix', 'removesuffix', 'replace', 'rfind',
'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines', 'startswith', 'strip', 'swapcase', 'title',
'translate', 'upper', 'zfill']
>>> 2 < 3
True
```

```
>>> 2 > 3
False
>>> 2 <= 3
True
>>> 'a' < 'b'
True
>>> 'b' < 'a'
False
>>> 'a' < 'A'
False
>>> ord('a')
97
>>> ord('A')
65
>>> 'A' > 'b'
False
>>> ord('A')
65
>>> ord('b')
98
>>> # and or not
>>> # not - reverses Truth status
>>> not True
False
>>> not False
True
>>> not 1
False
>>> not 3
False
>>> True == 1
True
>>> True == 2
False
>>> o = True
>>> not o
False
>>> p= False
>>> not p
True
>>> # and
>>> p
False
>>> q
```

```
12345678.12345
>>> True and False
False
>>> # and ret True only when both are True
>>> # or ret True if ANY one of the operands are true
>>> True or False
True
>>> False or False
False
>>> not True and False
False
>>> not True or False
False
>>> False and not True
False
>>> # '==' value equality
>>> 2 == 2
True
>>> 2 == 3
False
>>> not True == False
True
>>> False == not True
SyntaxError: invalid syntax
>>> False == (not True)
True
>>> # < <= == >= > 1st
>>> # not 2nd
>>> # and 3rd
>>> # or 4th
>>> True and False or True
True
>>> True and not False or not True
True
>>> True and True or False
True
>>> True or False
True
>>> # !=
>>> 2 != 3
True
>>>
```