02-real-and-complex-numbers

September 16, 2022

1 Real numbers

Real numbers, or floating point numbers, are represented in Python according to the IEEE 754 double-precision binary floating-point format, which is stored in 64 bits of information.

[1]:
```
pi = 3.1415926536
```

[2]:
```
from math import pi
pi
```

[2]:
```
3.141592653589793
```

[3]:
```
radius = 4.5
```

[4]:
```
area = pi * (radius ** 2)
area
```

[4]:
```
63.61725123519331
```

[5]:
```
import sys
sys.float_info
```

[5]:
```
sys.float_info(max=1.7976931348623157e+308, max_exp=1024, max_10_exp=308,
min=2.2250738585072014e-308, min_exp=-1021, min_10_exp=-307, dig=15,
mant_dig=53, epsilon=2.220446049250313e-16, radix=2, rounds=1)
```

[6]:
```
3 * 0.1 - 0.3    # shouldn't this be 0!!!
```

[6]:
```
5.551115123125783e-17
```

2 Fractions and decimals (optional)

[7]:
```
from fractions import Fraction
Fraction(10, 6)
```

[7]:
```
Fraction(5, 3)
```

[8]:
```
Fraction(1, 3) + Fraction(2, 3)
```

[8]:
```
Fraction(1, 1)
```

[9]:
```
1 == Fraction(1, 3) + Fraction(2, 3)
```

[9]:
```
True
```

[10]:
```
f = Fraction(10, 6)
```

[11]:
```
f.denominator
```

[11]:
```
3
```

[12]:
```
f.numerator
```

[12]:
```
5
```

[13]:
```
from decimal import Decimal
Decimal(3.14)             # pi, from float, so approximation issues
```

[13]:
```
Decimal('3.140000000000000124344978758017532527446746826171875')
```

[14]:
```
Decimal('3.14')          # pi, from a string, so no approximation issues
```

[14]:
```
Decimal('3.14')
```

[15]:
```
Decimal(0.1) * Decimal(3) - Decimal(0.3)      # from float, we still have the␣
↪issue
```

[15]:
```
Decimal('2.77555756156515654042363166 8E-17')
```

[16]:
```
Decimal('0.1') * Decimal(3) - Decimal('0.3')  # from string, all perfect
```

[16]:
```
Decimal('0.0')
```

3 Complex numbers (optional)

Python gives you complex numbers support out of the box.

[17]:
```
c = 3 + 2j
d = 1 - 1j
```

[18]:
```
c
```

[18]:
```
(3+2j)
```

[19]:
```
c.real
```

[19]:
```
3.0
```

[20]:
```
c.imag
```

[20]: 2.0

[21]:
```
c.conjugate()
```

[21]: (3-2j)

[22]:
```
c ** 2
```

[22]: (5+12j)

[23]:
```
c + d
```

[23]: (4+1j)

[24]:
```
c * d
```

[24]: (5-1j)

4 Exercises

Go here...