

Everything You Hoped You'll Never Have to Know About Numbers in Python

...but you really do

Moshe Zadka – <https://cobordism.com>

2020

Numbers in Math

"God gave us the natural numbers; the rest of it we have only ourselves to blame" – with apologies to Leopold Kronecker

- ▶ Natural numbers
- ▶ Integers
- ▶ Rationals
- ▶ Real numbers
- ▶ Complex numbers
- ▶ Are quaternions numbers? Who knows!

Numbers in Python

- ▶ `int`
- ▶ `float`
- ▶ `complex`
- ▶ `fractions`
- ▶ `decimal`
- ▶ Is `gmpy2` relevant? Who knows!

Integers, division, and multiplication

```
>>> (4/3) * 3  
4.0
```

floats

Just pay a \$100 for IEEE-754...

floats

Just pay a \$100 for IEEE-754...
(if you did it before 2019, pay again, they revised it)...

floats

Just pay a \$100 for IEEE-754...
(if you did it before 2019, pay again, they revised it)...
read all 70 pages...

floats

Just pay a \$100 for IEEE-754...
(if you did it before 2019, pay again, they revised it)...
read all 70 pages...
and memorize them.

floats: IEEE-754

- ▶ roundTiesToEven, the floating-point number nearest to the infinitely precise result shall be delivered; if the two nearest floating-point numbers bracketing an unrepresentable infinitely precise result are equally near, the one with an even least significant digit shall be delivered

floats: surprise!

```
>>> 1 + 2 - 2 - 1  
0
```

```
>>> 0.1 + 0.2 - 0.2 - 0.1  
2.7755575615628914e-17
```

floats: surprise strikes again!

```
>>> a = 2**-53
```

```
>>> (a + a) + 1 == a + (a + 1)  
False
```

fractions: a different sort of surprise

```
>>> before = datetime.now()
>>> res = sum(inverses[:10000])
>>> after = datetime.now()
>>> print("It took", after-before)
>>> print("Size of output", len(str(res)))
>>> print("Approximate value", float(res))
It took 0:01:08.524281
Size of output 90743
Approximate value 1.2538568497816165
```

fractions: a different sort of surprise

```
>>> before = datetime.now()
>>> res = sum(map(float, inverses[:10000]))
>>> after = datetime.now()
>>> print("It took", after-before)
>>> print("Size of output", len(str(res)))
>>> print("Approximate value", float(res))
It took 0:00:00.003096
Size of output 18
Approximate value 1.2538568497816087
```

fractions: a different sort of surprise

```
Approximate value 1.2538568497816165 # With precise
Approximate value 1.2538568497816087 # With floats
12345678901234
```

Decimals: Hidden state

Quote from documentation:

```
>>> getcontext().prec = 6
>>> Decimal(1) / Decimal(7)
Decimal('0.142857')
>>> getcontext().prec = 28
>>> Decimal(1) / Decimal(7)
Decimal('0.1428571428571428571428571429')
```

Decimals: Hidden state

Fixed it for you

```
>>> getcontext().prec = 6
....121 lines ...
>>> Decimal(1) / Decimal(7)
Decimal('0.142857')
>>> getcontext().prec = 28
....537 lines ...
>>> Decimal(1) / Decimal(7)
Decimal('0.1428571428571428571428571429')
```


Decimals: Using correctly

```
>>> getcontext().prec = 6
>>> # 6853 lines elided
... with localcontext() as ctx:
...     ctx.prec = 10
...     Decimal(1) / Decimal(7)
...
Decimal('0.1428571429')
```

Final thoughts

Think before you number!