Number

Kittikun Jitpairod

Numbers Types

Python primarily uses two types of numbers: integers and floating-point numbers (floats). Understanding these is essential for calculations and data manipulation in Python.

Integers (`int`)

Integers are whole numbers that can be positive, negative, or zero. They don't have decimal points.

```
-3, -1, 0, 1, 42, 1000
```

Floating-point numbers (`float`)

Floats are numbers with decimal points. They can be positive or negative, and can use scientific notation.

```
3.14, -0.001, 2.0, 1.0e-4  # 1.0e-4 is scientific notation for 0.0001
```

Converting between `int` and `float`

Python allows easy conversion between integers and floats:

```
x = float(5) # Integer to float
print(f"float(5) = {x}")
```

```
float(5) = 5.0
```

```
y = int(5.7) # Float to integer
print(f"int(5.7) = {y}")
```

```
int(5.7) = 5
```

Precision and limitations

Important considerations when working with numbers in Python:

1. Integers in Python 3 have unlimited precision.
2. Floats have limited precision, which can lead to unexpected results:

```
print(0.1 + 0.2)
```

```
0.30000000000000004
```

This unexpected result occurs due to how computers represent floating-point numbers in binary. Most decimal fractions cannot be represented exactly as binary fractions. For example, 0.1 in binary is a repeating fraction:

```
0.0001100110011001100110011001100110011001100110011...
```

This leads to tiny rounding errors. When doing math with these slightly inaccurate numbers, errors can add up, causing unexpected results. For precise calculations, especially with money, use Python's `decimal` module instead of floats.