Binary Numbers

Numbers in python

`integers` and `floats`

Integers are easy, floats are not

```python
x = 0
for i in range(10):
    x += 0.1

print(x == 1)
print(x, '==', 10*0.1)
```

Float errors

Operations on some floats introduces a very small error

This happens because floats aren’t stored exactly in a program.

Python and every other programming language uses "floating point" to approximate real numbers.

Floats

The way floats are stored in a program is through something called `binary`

Specifically, "floating point"

This means that it becomes difficult to make comparisons between floats and integers

Float representation

The way floats are represented on a computer can be different depending on the hardware, not the programming language

Humans like the `base 10` system of doing math, 'decimal'. But computers use `base 2`, 'binary'

Every single number in a computer, as well as everything else we see, is represented as a sequence of `bits` (0 or 1)

- When humans write down a number, we use a base 10 understanding
- for example 0.1 stands for the real number $1/10$

Binary representation

The reason we use binary is that it’s easy to build binary

This works okay for integer arithmetic, but not so well for floats

Binary representation

A base 10 representation of a number is a sum of powers of 10, scaled by integers from 0 to 9

$$\begin{aligned} 1507 &= 1 * 10^3 + 5 * 10^2 + 0 * 10^1 + 7 * 10^0 \\ &= 1000 + 500 + 0 + 7 \end{aligned}$$

This is how base 10 works, as well as every other base system, so in base 2, so if you want to convert it

$$\begin{aligned} 1507_{10} &= 1 * 2^{10} + 1 * 2^8 + 1 * 2^7 + 1 * 2^6 + 1 * 2^5 + 1 * 2^1 + 1 * 2^0 \\ &= 1024 + 256 + 128 + 64 + 32 + 2 + 1 \\ &= 2^{10} + 2^8 + 2^7 + 2^6 + 2^5 + 2^1 + 2^0 \\ &= 10111011100_2 \end{aligned}$$

Binary representation

When we input a decimal number, the computer needs to convert it into binary

A recipe way of doing this is

$$x = 19_{10} = 1 * 2^4 + 0 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0 = 10011_2$$

The way we figure out a binary representation is to repeatedly divide by 2 and keep track of the remainders

Example

Given the number 19_{10}, what is its binary representation?

$19/2 = 9r1$

That represents that last binary bit

Then again, $9/2 = 4r1$

Then again, $4/2 = 2r0$

Then again, $2/2 = 1r0$

Then again, $1/2 = 0r1$

So the binary bit is 10011_2

Exercise

Convert the following numbers to binary

1. 23_{10}
2. 42_{10}
3. 100_{10}

1/4 sheet of paper, show your solution