

# More on Binary

Q1: How to convert int to binary?

① use a built-in function:

$$a = 11 = 8 + 2 + 1$$

$$b = \text{bin}(a)$$

print(b)

> 0b1011

don't lose

0x8

② write your own function:

Why? Well, maybe we will eventually want to convert int to base 7, for some good reason. There is no built-in function for that!!

use the mod function!

# Algorithm:

Use the modulo operation  $(\%)$   $+$ ,  $-$ ,  $*$ ,  $/$   
= remainder after integer division.

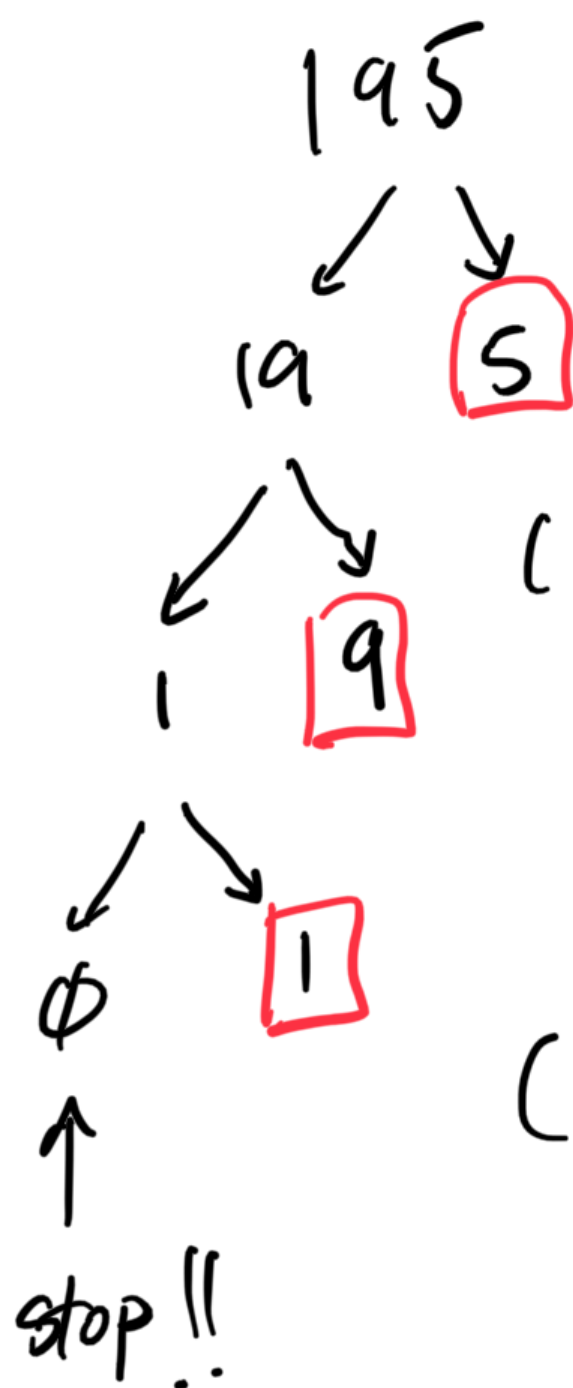
Example: (i)  $195 \% 10 \Rightarrow 19 \times 10 + 5$   
 $= 5$  ↑  
remainder

= the last digit!!!

(ii)  $195 / 10 = \text{int}(19.5)$   
 $= 19$

= the other digits!!!

(iii) keep repeating.



$$19 \% 10 = 9 \quad \checkmark$$

$$19 / 10 = 1 \quad \checkmark$$

$$1 \% 10 = 1$$

$$1 / 10 = 0 \leftarrow \text{end!}$$

Here's the cool thing: this algorithm

works for any base !!!! 😊

So, to convert to binary, we just use 2.

Ex.

11:

$$11 \% 2 = \boxed{1} \checkmark$$

$$11 / 2 = 5 \checkmark$$

5:

$$5 \% 2 = \boxed{1} \checkmark$$

$$5 / 2 = 2 \checkmark$$

2:

$$2 \% 2 = \boxed{0} \checkmark$$

$$2 / 2 = 1 \checkmark$$

1:

$$1 \% 2 = \boxed{1} \checkmark$$

$$1 / 2 = \phi \leftarrow \text{stop} \checkmark$$

$$\begin{array}{r} 64 \ 16 \ 4 \ 1 \\ 23 \end{array}$$

$$2 \times 4 + 3$$

$$8 + 3$$

$$= 11$$

Ans. 11

$\Rightarrow 1011,$

11/15 10

=====

---

Have a look at :

Convert\_\*.py

in Week1 - Examples for  
more information.

---