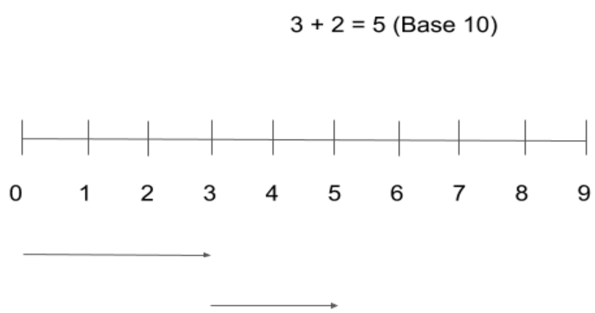
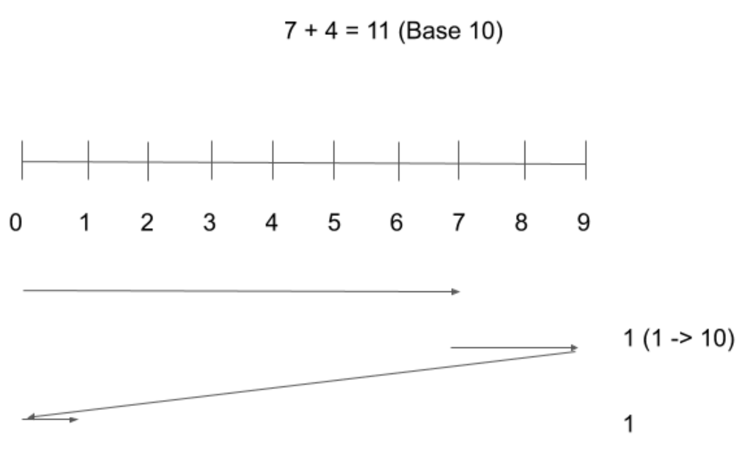
Base 10
This is a number system that has 10 different symbols (digits) that represent numbers. It starts from Zero.

If we pick the Western Arabic set we have: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Using what we learned up above about positional value, Let's compute 3 + 2 = 5



But what happens when we add 7 to 4?

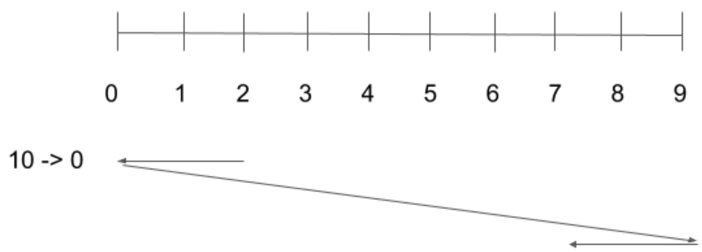


Starting from 7, we move to the right by the number of steps indicated by the second parameter.

We mark we crossed the right boundary once by incrementing the position at the immediate left by one digit (Remember: every single digit (“3”, “5”, “9”, etc) has an invisible line of zeros before it that goes towards the left towards infinity) and we continue counting from the other end of the bar (this is called rollover).

What about 12 - 5?

12 - 5 = 7 (Base 10)



Starting from 2, we move to the left by the number of steps indicated by the second parameter. We mark we crossed the left boundary once by decreasing the second position by one digit, and we continue counting backward from the other end of the bar (**rollover**).

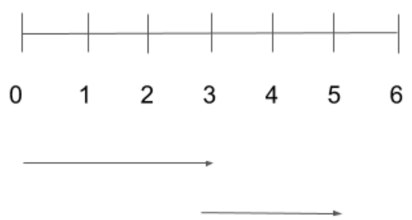
Base 7

This is a number system that uses 7 different symbols (digits) to represent numbers. It starts from Zero.

To make our visual representation easier, we'll use the first 7 symbols from Western Arabic set as our symbols: 0, 1, 2, 3, 4, 5, 6, but we could invent our own set of symbols like: !, @, #, $, %, ^, &, *.

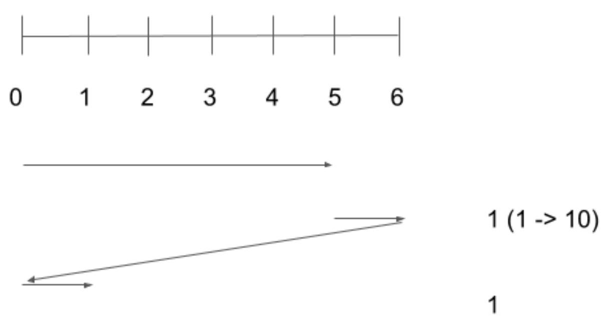
Let's compute 3 + 2 = 5

3 + 2 = 5 (Base 7)



But what happens when we add 5 and 3? 5+3 = 11

5 + 3 = 11 (Base 7)



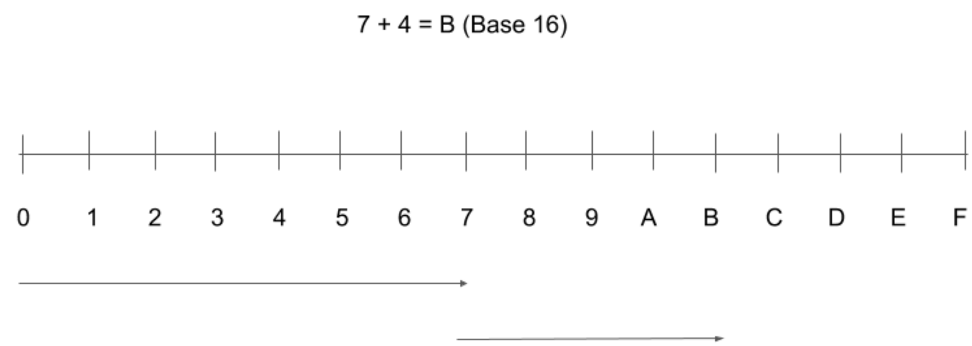
In this case starting from the "5" mark we move to the right one step and we reach the end of the domain ("6"), one more step to rollover to the beginning of the domain (the "0") and the final third step to reach the "1" mark.

Base 16

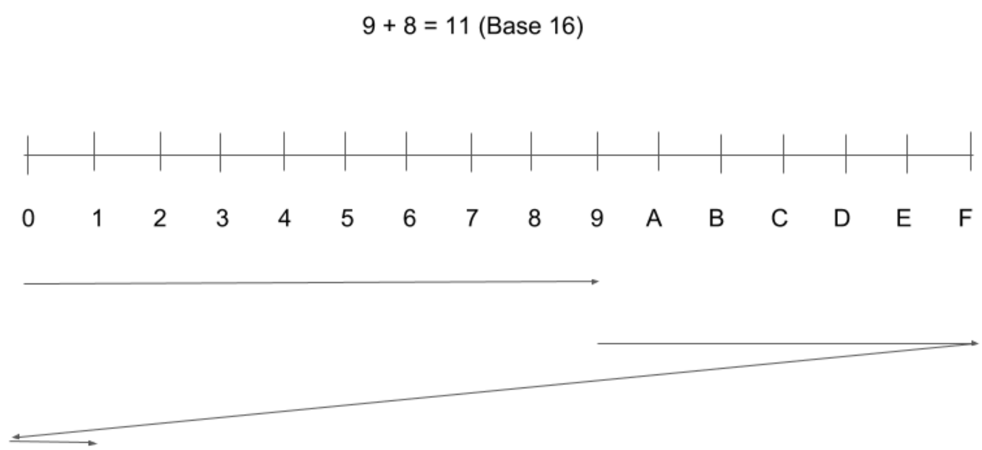
This is a number system that uses 16 different symbols (digits) to represent numbers. It starts from Zero.

Since we do not have enough symbols in the Western Arabic convention to cover 16 digits, it is a convention to use a set of symbols like this one: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Let's compute 7 + 4 = B



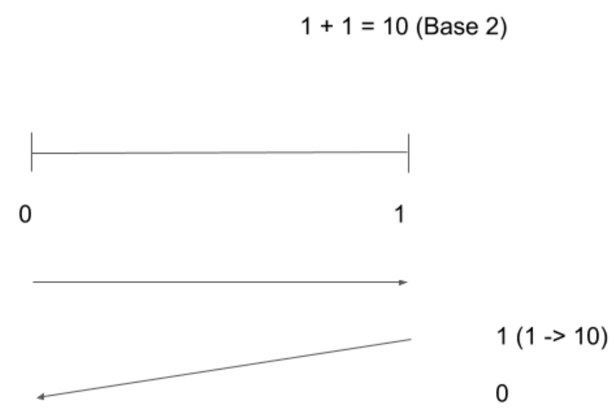
Let's now compute 9 + 8 = 11



Base 2

It means we have only two symbols (digits) that we can use to represent numbers. It starts from Zero.

To make our visual representation easier, we pick the first 2 symbols for numbers from Western Arabic set: 0, 1
Let’s compute 1 + 1 = 10



Base *X* numbers

If you want to create a Base X numbers set (with X representing whatever you want it to be), you have to define the list of X symbols and their order in regards to each other (starting from Zero). Then, by applying the same rules listed above you will be able to compute any mathematical operation.

For example, if we want to define a Base 4 numbers set, we could define our symbols as:
. (zero)
(one)
@ (two)
***** (three)
And our math will be:
1 + 2 = 3 (Base 4) ===> **# + @ = ***
3 + 2 = 11 (Base 4) ===> *** + @ = ##**

Number sets
In math we have different numeric sets, each of them representing different types of numbers and different ranges of them inside each set.
The most common ones are:

- Naturals
- Integers
- Real

Naturals
This is the set of whole numbers (no decimal point) starting from Zero and up to infinity:
0, 1, 2, 3, ….. , 11223344, …… , 3948576968,...... +∞

Integers
This set still consists of whole numbers, but includes negative numbers:
-∞, …, -847565679403, …, -7734433, ..., -123,..., -3, -2, -1, 0, 1, 2, 3, 4, …, 123432245,...., +∞

Real
This set includes all numbers from -∞ to +∞, but now also includes all the different variations of fractions or decimals.:
-∞, …, -84756567.9403, …, -123.99988876,..., -3.3, -2.0, -1.55, 0, 0.2233, 1.1., 2.009, …, 1234322.45,...., +∞

Considering how big or long these numbers may be to write down, a convention has been introduced called Scientific Notation to better manage their written representation.
In a nutshell we represent these long numbers as a power of 10 numbers.
How it works:
This notation always normalizes the number between -1.0 and 1.0 by moving the decimal point of the number: you move the decimal point one position to the right and you add to the number representation a divide by 10 to compensate. In the same way if you move the decimal point to the left you add to the representation of the number a multiply by 10 to compensate.

123.456 ⇒ 12.3456 * 10 ⇒ 1.23456 * 10 * 10 ⇒ 0.123456 * 10 * 10 * 10 (aka 0.123456 * 10^3)
0.00123456 ⇒ 0.0123456 / 10 ⇒ 0.123456 / 10 / 10 (aka 0.123456 * 10^{-2})

In order to better represent the powers of 10 you multiply or divide the number the notation uses this format: **E*sn*,** where **E** is a fixed symbol meaning it is multiplied by a power of 10, ***s*** is the sign denoting the nature of the exponent (+ or -) and ***n*** the power index. Note: when ***s*** is positive (+) you are permitted to leave it out.

Let's make few examples:
1.0 ==> 0.1E01
100 ==> 0.1E03
-1.0 ==> -0.1E01

0.01 ==> 0.1E-01
-0.01 ==> -0.1E-01
123.456 ==> 0.123456E03
0.00123456 ==> 0.123456E-02

Within this notation, the section of the number that comes after the decimal point is called **mantissa**, and the power index is called **exponent.**