Positional Number System

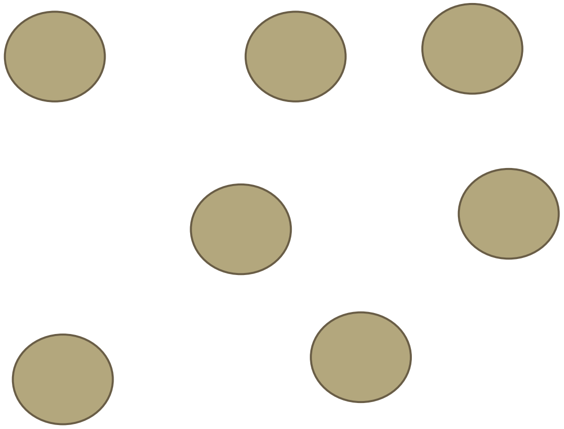
How to represent numbers

Positional Number Systems

- Computers store/process numbers, and only numbers.
- So, we need to represent numbers conveniently:
  - for machine storage
  - for machine arithmetic
- First let us look at how we represent numbers

How many items are there?

- 21
- 13
- 111
- 11
- 12
- 7



Positional Number Systems

- There are 10 types of people in the world:
    - Those who understand binary and those who don’t

- **CRITICAL**: A numeric value is invariant. It’s symbolic representation will change depending on the base used

Positional Number Systems

- Humans use the decimal (base-10) number system:
    - 10 digits (0 through 9)
    - Numbers are written as a string of these digits
    - Each string position has a “place value”
        - from the right: 1, 10, 100, 1000, …
    - Left-most digit has the largest value associated with it
        - i.e. is the most significant digit

Positional Number Systems

- We can generalize base-10 knowledge to any base-B number systems, where B >= 2
- Any given numeric value can be expressed, symbolically, in any given base

Positional Number Systems

- How many digits in base-B number system?
- What are the digits in the base-B number system?

Criteria	base-10	base-B
# of digits	10	B
range	0 – 9 (10 – 1)	0 – B-1

Positional Number Systems

- General rules for base-B:
  - # of possible digits: B
  - All possible symbols: 0 to B-1
  - Letters are used for symbols 10 to 35 starting with A
  - A=10, B=11, … , Z=35

Positional Number Systems

- Example numbers in different bases:
  - Base-5 (in this case B is 5)
  - # of possible digits = 5
  - All possible symbols: 0 , 1, 2, 3, 4

Positional Number Systems

- Example numbers in different bases:
  - Base-12 (in this case B is 12)
  - # of possible digits = 12
  - All possible symbols: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B

Positional Number Systems

- How are numbers interpreted in base-10?
  - $d_n*10^n + d_{n-1}*10^{n-1} + \ldots + d_2*10^2 + d_1*10^1 + d_0$
- How are numbers interpreted in base-B?
  - $d_n*B^n + d_{n-1}*B^{n-1} + \ldots + d_2*B^2 + d_1*B^1 + d_0$
  - generic rule for any base, B will be >=2
- Is base-1 a positional number system?

Positional Number Systems

Example numbers in different bases:

1. Base-7 (in this case B is 7)
    - Number of symbols is 7
        - (0, 1, 2, 3, 4, 5, 6)
    - $63425_7 = 6*7^4 + 3*7^3 + 4*7^2 + 2*7^1 + 5*7^0$

Positional Number Systems

Example numbers in different bases:

1. Base-12 (in this case B is 12)
    - Number of symbols is 12
        - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B
    - $9B2A4_{12} = 9*12^4 + B*12^3 + 2*12^2 + A*12^1 + 4*12^0$

Positional Numbering Systems

Decimal (Base-10)	Binary (Base-2)	Base-5	Hexadecimal (Base-16)
1	1	1	1
2	10	2	2
5	101	10	5
10	1010	20	A
15	1111	30	F
20	10100	40	14
45	101101	140	2D

Positional Numbering Systems

Base-10	1	2	4	5	9	10	11	12	14	15	16	19	20	21
Base-2	1	10	100	101	1001	1010	1011	1100	1110	1111	10000	10011	10100	10101
Base-5	1	2	4	10	14	20	21	22	24	30	31	34	40	41
Base-12	1	2	4	5	9	A	B	10	12	13	14	17	18	19
Base 16	1	2	4	5	9	A	B	C	E	F	10	13	14	15

Positional Number Systems

- ALWAYS subscript a number with its base unless perfectly clear from the context
  - 110_2 = number in Base-2
  - 10 is ambiguous! The meaning depends on the base:
    - $10_2 \mathrel{!=} 10_{10}$
    - $10_{10} == 1010_2$
    - $10_2 == 2_{10}$

Positional Number Systems

- How are numbers represented/stored in a computer?
- In electronics, two-state circuits (on/off) are relatively easy to design
  - Modern computers represent numbers in binary
- Binary numbers are also the most amenable to arithmetic circuit design (e.g. adders)

Positional Number Systems

- Binary (base-2)
  - 2 digits (0 and 1)
  - numbers are written as a string of binary digits, or “bits”
  - “place value” from the right:
    - 1 (2^0), 2 (2^1), 4 (2^2), 8 (2^3), 16 (2^4), …