

What is "001101"? Type String: Java
 or "a a b b a b" string C++
const char* C/C++

String: list <char> list of characters.

or array <char> specify an indexing number from 0 or from 1 (or...?)

A (finite) set of characters is called an alphabet

Examples: {0, 1} Binary - default if not specified.
 {0, 1, ..., 9} digits

Harmonized by ASCII
UTF encoding < UNICODE ("wide characters")

Symbols = Σ (Greek Sigma) for alphabet

Roman x, y, z, w, v, u... for variable strings.
 a, b, c, characters, variable or literal.

In " $\Sigma = \{a, b\}$ " the 'a' and 'b' are literal chars.

(2) Strings are used to represent numbers: Natural #s $\mathbb{N} = \{0, 1, 2, \dots\}$
 $\mathbb{N}^+ = \{1, 2, 3, \dots\}$
 ~~\mathbb{Z}~~ includes negative integers.

Binary Strings and Numbers $\Sigma = \{0, 1\}$

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
h	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	<u>1101</u>

$0 = 000000$
 $1 = 000001$
 \vdots
 4th entry: $13 = 001101$
 \vdots
 $63 = 111111$

Alphabetical listing
 of $\{0, 1\}^6$
 \equiv the set of all six-bit strings.
 Soon we will write $(0+1)^6$ or $(0+1)^6$.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
*	λ	0	1	00	01	10	11	000	001	010	011	100	101	110
1	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	<u>1101</u>	1110
1	1	2	3	4	5	6							13	

All Binary strings: (λ) denotes the empty string, aka "", aka ϵ .
 In "Lex Order": (λ) Also finite.

Σ^* denotes the set of all possible strings over an alphabet Σ .
 $\Sigma = \{0, 1\}$ can write $\{0, 1\}^*$ or $(0+1)^*$

- "001101" As a Mapping: $f: \{0, 1, 2, 3, 4, 5\} \rightarrow \{0, 1\}$
 $f(0)=0$ $f(1)=0$ $f(2)=1$ $f(3)=1$ $f(4)=0$ $f(5)=1$
- As a Bit-Vector of a Set: $\{2, 3, 5\}$ excluding $\{0, 1, 4\}$ from the domain $\{0, 1, 2, 3, 4, 5\}$
 001101.0.1
- As the Relation of the Mapping: $= \{(0,0), (1,0), (2,1), (3,1), (4,0), (5,1)\}$
 (LNO of 1st lecture)