Characters: ASCII

Numbers are not the only kind of data to process.
Need letters, digits, punctuation, etc.
Assigning binary codes to characters is arbitrary.
Finite set, unlike numbers.
ASCII: American Standard Code for Information Interchange
Standard: can be used by many kinds of computers, regardless of architecture.

man ascii (in UNIX):

0 nul	16 dle	32 sp	48 0	64 @	80 P	96 `	112 p
1 soh	17 dc1	33 !	49 1	65 A	81 Q	97 a	113 q
2 stx	18 dc2	34 "	50 2	66 B	82 R	98 b	114 r
3 etx	19 dc3	35 #	51 3	67 C	83 S	99 c	115 s
4 eot	20 dc4	36 $	52 4	68 D	84 T	100 d	116 t
5 enq	21 nak	37 %	53 5	69 E	85 U	101 e	117 u
6 ack	22 syn	38 &	54 6	70 F	86 V	102 f	118 v
7 bel	23 etb	39 '	55 7	71 G	87 W	103 g	119 w
8 bs	24 can	40 (	56 8	72 H	88 X	104 h	120 x
9 ht	25 em	41)	57 9	73 I	89 Y	105 i	121 y
10 nl	26 sub	42 *	58 :	74 J	90 Z	106 j	122 z
11 vt	27 esc	43 +	59 ;	75 K	91 [	107 k	123 {
12 np	28 fs	44 ,	60 <	76 L	92 \	108 l	124 |
13 cr	29 gs	45 -	61 =	77 M	93]	109 m	125 }
14 so	30 rs	46 .	62 >	78 N	94 ^	110 n	126 ~

15 si 31 us 47 / 63 ? 79 O 95 _ 111 o 127 del

Kinds of symbols

- control characters
- punctuation/operator symbols
- digits
- uppercase alphabet
- lowercase alphabet

Uppercase alphabet, lowercase alphabet, digits are each contiguous

Only 128 characters, but a byte is used to store, so msb of character is 0.

- Sometimes extended (IBM PC) to 256 characters.

Characters: EBCDIC

Since binary representation is arbitrary, other codes are possible.

EBCDIC

- **Extended Binary Coded Decimal Interchange Code**
- **Used on IBM mainframes**
- **256 values**
- **Alphabet not contiguous**
- **Chart (from www.legacyj.com)**

Characters: Unicode

Many world languages cannot be represented using 8-bit code.

Unicode

- **16-bit representation**
- **64K different symbols**
- **ASCII included as a subset (zero-extend to 16 bits)**
- **evolving standard**
  - **version 2.0 supports 38,885 distinct characters from many languages**

supported by Java (char type is 2-byte)

- **needs separate byte type**

Characters: files

What happens when you type "123" in a text editor and save it to a file?

Characters '1', '2', '3' are stored in the file, not binary integer 123.

What if you type a series of 0's and 1's?

Still just characters

How can you write binary representation of data to a file?

hex editors allow you to type hex values like F3, 2A, etc. and store as bytes

In C, use "b" in mode string with fopen to open the file

- **"rb": open for reading**
- **"wb": open for writing**
- **etc.**