

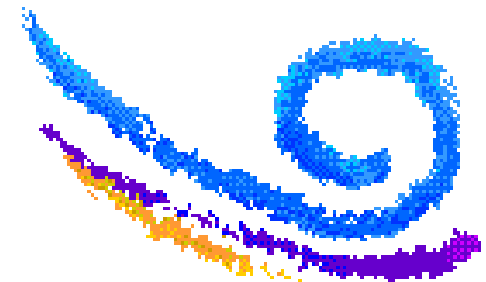
# COBOL Coding Overview

## Format and Structure

# COBOL Coding Format

---

- ▶ COBOL source programs are written in a file with fixed length lines (80 columns)
- ▶ Lines are logically divided into these areas:
  - columns 1- 6: blank or, optionally, sequence numbers
  - column 7: '-' (dash) indicates a continuation  
'\*' (asterisk) indicates a comment line  
'D' indicates a debugging line
  - columns 8-11: **Area A**, starting position area for headers (division headers, section headers, paragraph headers, and data headers (levels 01 or 77))
  - columns 12-72: **Area B**, all other code
  - columns 73-80: blank or, optionally, program name



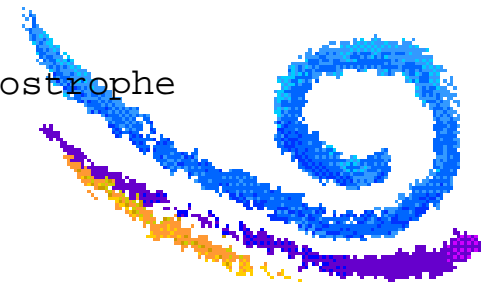
# COBOL Character Set

---

- ▶ COBOL source programs are written using the following set of characters, the basic character set:
  - 52 alphabetic characters (a-z, and A-Z; words are case-insensitive, but quoted values are not; case insensitivity introduced in COBOL II)
  - 10 character digits (0-9)
  - 18 special characters:

<u>Glyph</u>	<u>Name</u>
	space
.	decimal point, period
<	less than
(	left parenthesis
+	plus sign
\$	dollar sign, currency symbol
*	asterisk
)	right parenthesis
;	semi-colon

<u>Glyph</u>	<u>Name</u>
:	colon
-	minus sign, dash, hyphen
/	slash, stroke
,	comma
>	greater than
=	equals sign
"	double quote
'	single quote, apostrophe
_	underscore



# COBOL Separators

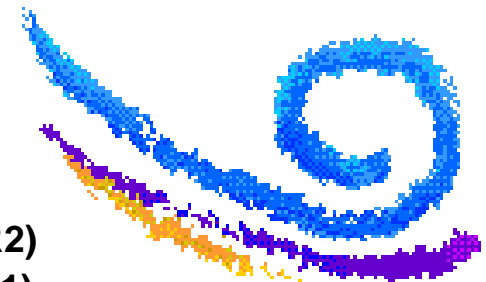
---

- ▶ The following sequences are designated as COBOL separators; a separator delimits a character string:

␣	space
,␣	comma
.␣	period
;␣	semi-colon
(	left parenthesis
)	right parenthesis
:	colon
"␣	end double-quoted string
'␣	end single-quoted literal
==	pseudo-text delimiter
x' or x"	begin hexadecimal literal
z' or z"	begin null-terminated literal
n' or n"	begin DBCS or national literal
g' or g"	begin DBCS literal
nx' or nx"	begin hexadecimal national literal

**Note that x, z, n, and g may be upper case or lower case.**

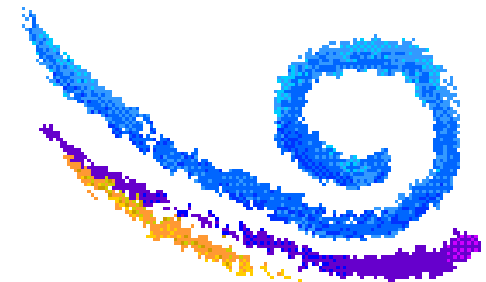
- \* Hexadecimal literals were introduced in COBOL II
- \* Null-terminated literals were introduced in COBOL for MVS & VM (V1R2)
- \* National (Unicode) literals were introduced in Enterprise COBOL (V3R1)



# COBOL Character Strings

---

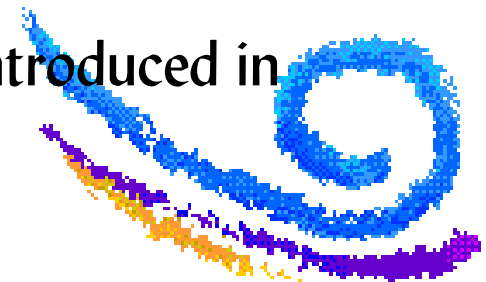
- ▶ A character string is a single character or sequence of contiguous characters that forms a word, literal, picture character string, or comment
- ▶ A character string is delimited by a separator
- ▶ Note that literals and run-time data can include characters other than the basic character set



# COBOL Words

---

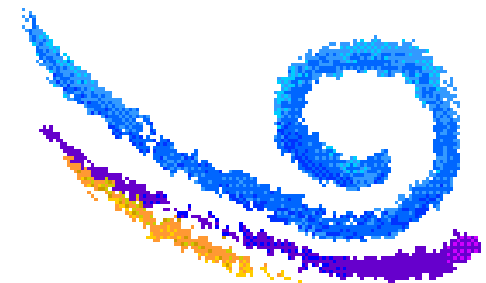
- ▶ COBOL words are 1-30 contiguous alphanumeric or dash or underscore characters, delimited by a separator
  - Dash must not be the first nor the last character, underscore must not be the first
  - Must contain at least one alpha character (except paragraph names and section names)
  
- ▶ Each COBOL word belongs to one of these types:
  - User-defined word (chosen by the programmer, not a reserved word)
  - System-name (*e.g.*: SYSIN)
  - Function names (*e.g.*: current-date) (Intrinsic functions introduced in COBOL/370 (VIRI))
  - Reserved words (see next page)



# COBOL Reserved Words

---

- ▶ A small number of COBOL words are reserved:
  - Key words (e.g.: ADD, READ, WRITE, IF,...)
  - Optional words (e.g.: IS, ARE, ... )
  - Special registers (e.g.: LINAGE-COUNTER, TALLY, ...)
  - Special character words (complete list: + - / \* \*\* < > = <= >=)
  - Figurative constants (e.g.: ZERO, SPACES, ...)
  - Special Object Identifiers (complete list: SELF, SUPER)



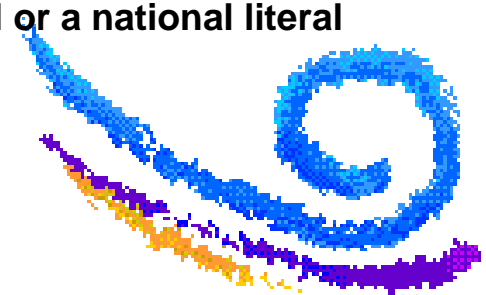
# COBOL Literals

---

▶ A literal is a representation of data; one of:

- Numeric: 1-18 digits (0-9), optional sign (+ or -), no commas
- Non-numeric: single- or double-quoted, 1-160 characters
- Hexadecimal: inside x'...' or x"...", contents only hexadecimal characters (0-9, a-f, A-F), 2-320 hex digits
- Null-terminated: inside z'...' or z"...", 1-159 characters, COBOL appends a null (x'00')
- DBCS (Double Byte Character Set): 1-28 characters inside g'...', g"...", n'...', or n"..."
- National (Unicode): 1-80 characters inside n'...' or n"..."

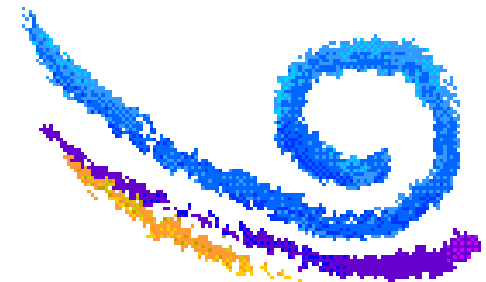
**Note: a compiler setting determines if n'...' or n"..." indicates a DBCS literal or a national literal**



# COBOL Picture Character Strings

---

- ▶ A PICTURE character string occurs only in a PICTURE (or PIC) clause and ...
  - Describes a data item  
e.g.: PIC X(20)  
  
or
  - Describes how a data item should be edited for display  
e.g.: PIC ZZ,ZZ9.99



# COBOL Program Structure

---

- ▶ A COBOL program is divided into 1-4 divisions:
  - Identification division - names the program; may specify a few attributes of the program as a whole
  - Environment division - identifies characteristics such as files available, special processing such as currency symbols, *etc.*
  - Data division - provides additional file information and identifies all data items
  - Procedure division - contains the logic of the program
- ▶ Not all divisions must be present (Identification division must be present) but those that are present must appear in this relative order

# COBOL Program Structure, 2

## ► Divisions may be further subdivided, as follows:

### IDENTIFICATION DIVISION

- Paragraphs
- Entries
- Clauses

### ENVIRONMENT DIVISION

- Sections
- Paragraphs
- Entries
- Clauses
- Phrases

### DATA DIVISION

- Sections
- Entries
- Clauses
- Phrases

### PROCEDURE DIVISION

- Sections
- Paragraphs
- Sentences
- Statements
- Phrases

Each division must have a header

\* Division name followed by **DIVISION**.

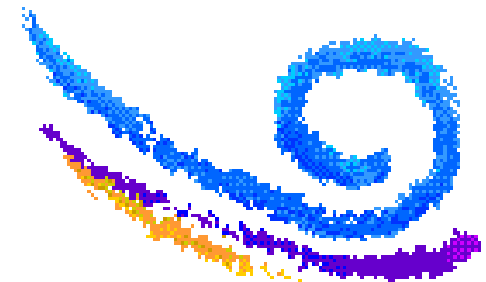
Each section must have a header

\* Section name followed by **SECTION**.

Each paragraph must have a header

\* Paragraph name followed by a period .

Remember that each division-, section-, and paragraph-header must begin in area A (columns 8-11)



# A Sample, complete COBOL program

## ► Simple, but all here:

Identification division.

Program-id. ISDF2F.

Environment division.

Input-output section.

File-control.

Select INFILE assign to INDD.

Select OUTFILE assign to OUTDD.

Data division.

File section.

FD INFILE

Block contains 0 records.

01 INREC PICTURE X(128).

FD OUTFILE

Block contains 0 records.

01 OUTREC PIC X(128).

Working-storage section.

01 Record-work pic x(128).

Procedure division.

Initialization section.

Open input INFILE

output OUTFILE.

Copyfile section.

Read INFILE into record-work

at end go to termination.

Write outrec from record-work

Go to copyfile.

Termination section.

Close INFILE OUTFILE.

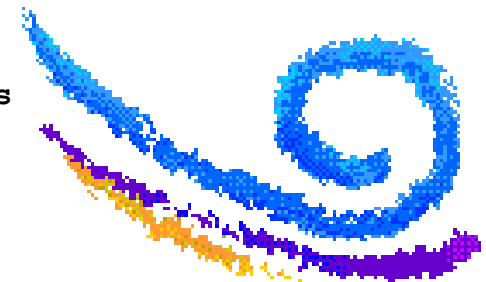
Stop Run.

Color key:

Red - division headers

Blue - section headers

Green - paragraph headers





6790 East Cedar Avenue, Suite 201  
Denver, Colorado 80224  
USA

<http://www.trainersfriend.com>  
303.393.8716

Sales: [kitty@trainersfriend.com](mailto:kitty@trainersfriend.com)  
Technical: [steve@trainersfriend.com](mailto:steve@trainersfriend.com)