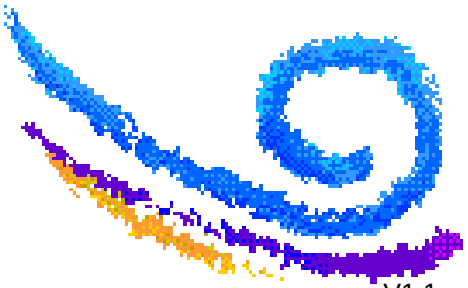
IBM Enterprise COBOL

Declaring Data Items in COBOL programs

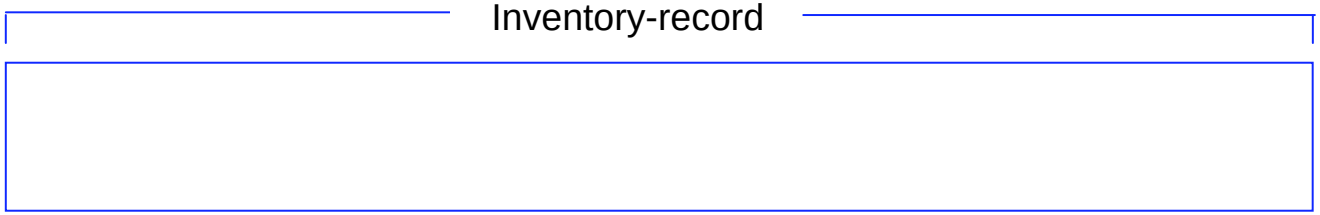
Elementary and Group Data Items

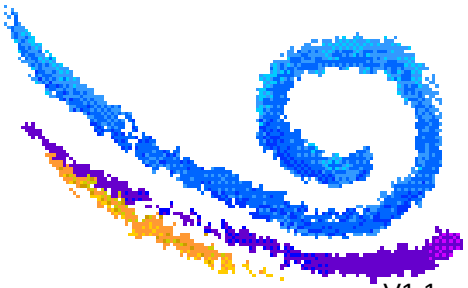
- In COBOL, a data item is either a Group item (it is sub-divided: composed of sub-items) or an Elementary item (it is not sub-divided)
  - A structure, such as a data record in a disk file, is usually a collection of fields grouped together under a group name, or record name (which must follow the usual rules for user-defined names in COBOL)



Example of a data structure: a record

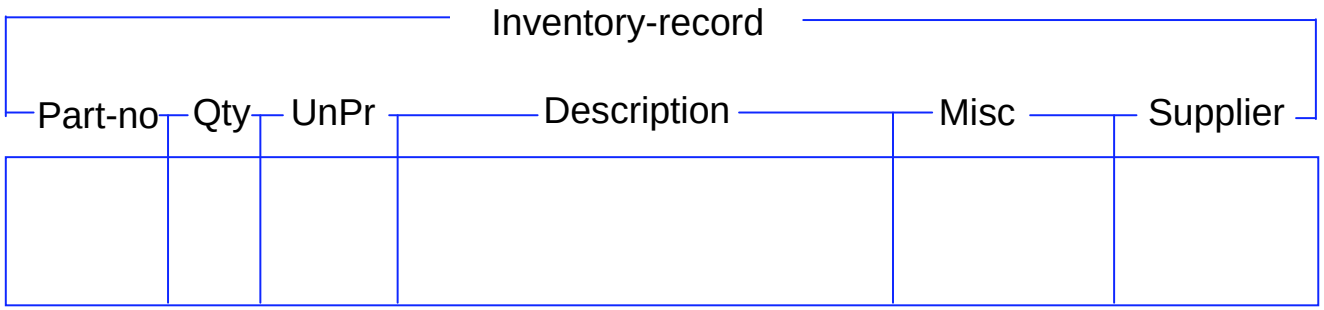
- Suppose we have a record that describes an item in our inventory on hand
  - We can think of this record as a single entity:





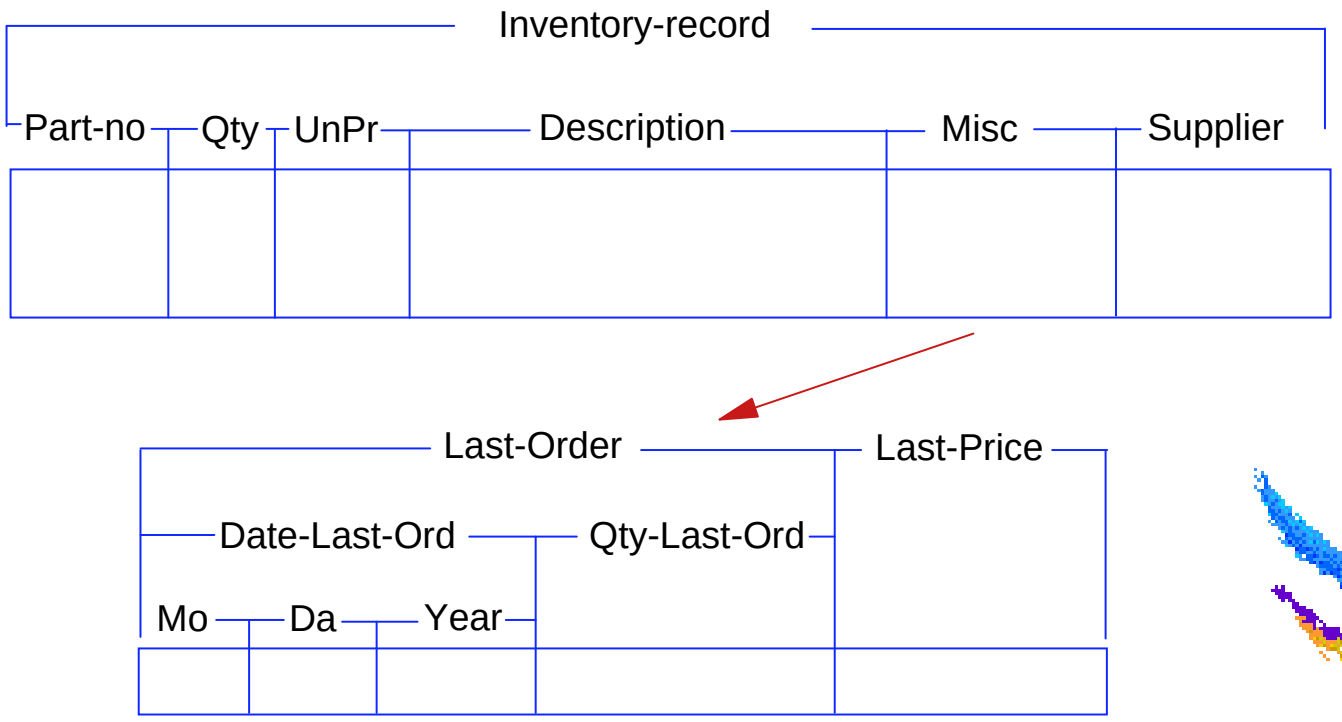
Example of a data structure: Fields in a record

- However, we really need to work with some of the fields in the record
  - For example, the record may be composed of these fields:



A data structure and sub-fields in a record

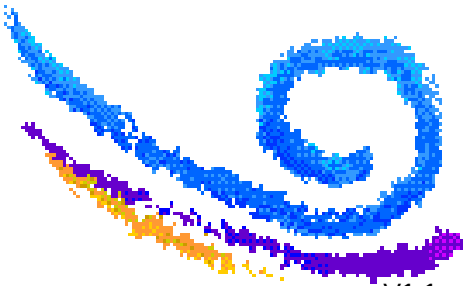
- Furthermore, some fields themselves may have sub-fields
  - For example, the 'Misc' field may be composed of these sub-fields:



Level Numbers

- When an item is declared / defined in a COBOL program it is assigned a level number (from 01-49, 66,77,88)

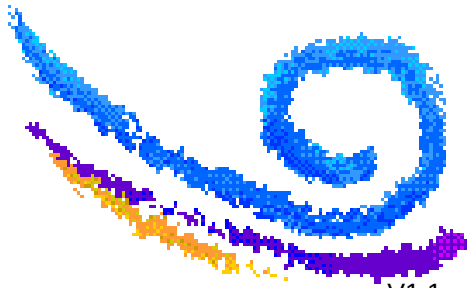
- Level number 01 is the group level
- Numbers 02-49 indicate relative depth within the hierarchy of a data structure
- Numbers 66, 77, and 88 have special meainings, discussed elsewhere



Describing data items

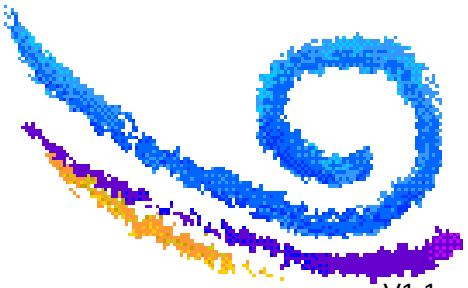
- Now, to describe this structure and its subfields we start with this:

```
01  Inventory-record
    02  Part-no
    02  Qty
    02  UnPr
    02  Description
    02  Misc
        03  Last-Order
            04  Date-Last-Ord
                05  Mo
                05  Da
                05  Year
            04  Qty-Last-Ord
        03  Last-Price
    02  Supplier
```



Notes on level numbers

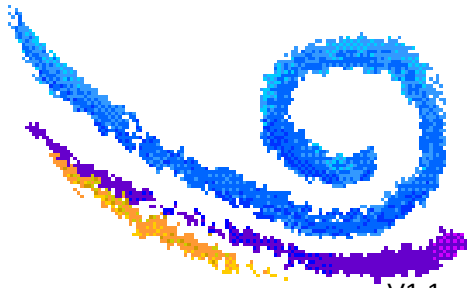
- The first level is always 01 (or just 1), and this level number must begin in Area A, while the associated name must begin in Area B
  - All deeper levels must be totally contained in Area B
  - Level numbers do not need to be assigned sequentially, but they must be ascending for deeper levels



Data items and PICTURE

- Elementary data items (those not sub-divided further) must then have a data description (usually a PICTURE clause and / or a USAGE clause)
  - Group items (those that are subdivided) must not have a data description: group items are implicitly described by the elementary items they contain

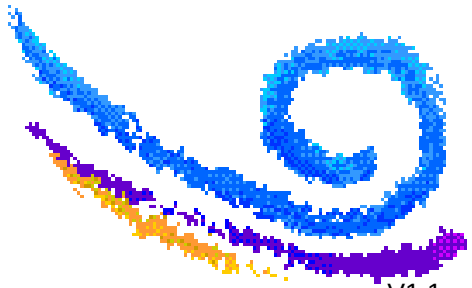
```
01  Inventory-record.
    02  Part-no                PICTURE clause.
    02  Qty                    PICTURE clause.
    02  UnPr                   PICTURE clause.
    02  Description            PICTURE clause.
    02  Misc.
        03  Last-Order.
            04  Date-Last-Ord  PICTURE clause.
                05  Mo         PICTURE clause.
                05  Da         PICTURE clause.
                05  Year       PICTURE clause.
            04  Qty-Last-Ord   PICTURE clause.
        03  Last-Price         PICTURE clause.
    02  Supplier               PICTURE clause.
```



Data item descriptions

- Note that each item description, group or elementary, must end with a period
  - An 01-level data item that is not subdivided is considered to be both a group item and an elementary item so it must have a PICTURE; for example:

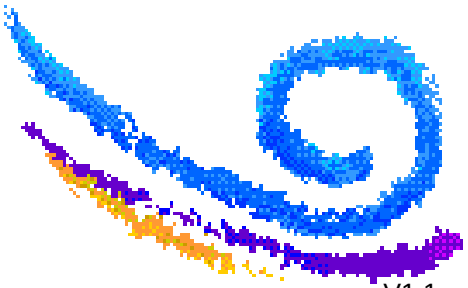
```
01  Inventory-record  PICTURE X(100).
```



Data descriptions, concluded

- The paper titled "Data Types Allowed in COBOL" contains a discussion of the PICTURE and USAGE clauses

- The paper titled "PICTURE Clauses" contains more information about PICTURE clause options



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

http://www.trainersfriend.com
303.393.8716
Sales: kitty@trainersfriend.com
Technical: steve@trainersfriend.com