2.1.3.2. Data name

A data name (or tag) is the identifier of a data value (see Section 2.1.3.3) and is a sequence of non-white-space characters starting with an underscore character `<_>` (ASCII 95).

Example:

```
_publication_author_address
```

2.1.3.3. Data value

A data value is a text string preceded by its identifying data name. Privileged keywords, such as described in Sections 2.1.3.5 to 2.1.3.8, are excluded from this definition.

2.1.3.4. Data item

A data item is a data value and its associated data name. Each data item stored in a STAR File is specified with this combination.

2.1.3.5. Data loop list

A looped list consists of the keyword `loop_` followed by

(*a*) a sequence of data names (possibly with nested `loop_` constructs); and

(*b*) a sequence of loop packets, each containing data values which are identified in the same order as the data names in (*a*).

A looped list specifies a table of data in which the data names represent the 'header descriptors' for columns of data and the packets represent the rows in the table. Looped lists may be nested to any level. Each loop level is initialized with the `loop_` keyword and is followed by the names of data items in this level. Data values that follow the nested data declarations must be in exact multiples of the number of data names. Each loop level must be terminated with a `stop_`, except the outermost (level 1) which is terminated by either a new data item or the privileged strings indicating a save frame (Section 2.1.3.6), a data block (Section 2.1.3.7), a global block (Section 2.1.3.8) or an end of file.

An example of a simple one-level loop structure is:

```
loop_
    _atom_identity_number
    _atom_type_symbol       1 C   2 C    3 O
```

Nested (multi-level) looped lists contain matching data packets [as per (*b*) above] and an additional `stop_` to terminate each level of data. Here is a simple example of a two-level nested list.

```
loop_
    _atom_id_number
    _atom_type_symbol
    loop_
        _atom_bond_id_1
        _atom_bond_id_2
        _atom_bond_order
             1 C     1 2 single  1 3 double stop_
             2 C     2 1 single             stop_
             3 O     3 1 double             stop_
```

The matching of data names to value packets is applied at each loop level. Initially the data values are matched to the data names listed in the outermost level loop. This process is iterated to successively inner levels. At the innermost loop level, data matching is maintained until a `stop_` is encountered. This returns the matching process to the next outer level. The matching process is recursive until the loop structure is depleted. Here is an example of a three-level loop structure.

```
loop_
    _atomic_name
     loop_
         _level_scheme
         _level_energy
          loop_
              _function_exponent
              _function_coefficient
 hydrogen
        (2)->[2]    -0.485813
        1.3324838E+01    1.0
        2.0152720E-01    1.0 stop_
        (2)->[2]    -0.485813
        1.3326990E+01    1.0
        2.0154600E-01    1.0 stop_
        (2)->[1]    -0.485813
        1.3324800E-01    2.7440850E-01
        2.0152870E-01    8.2122540E-01 stop_
        (3)->[2]    -0.496979
        4.5018000E+00    1.5628500E-01
        6.8144400E-01    9.0469100E-01
        1.5139800E-01    1.0000000E+01 stop_ stop_
```

2.1.3.6. Save frame

A save frame is a set of unique data items wholly contained within a data block. The frame starts with a `save_`*framecode* statement, where the *framecode* is a unique identifying code within the data block. Each frame is closed with a `save_` statement.

Example:

```
data_example
save_phenyl
    _object_class    molecular_fragment
     loop_
         _atom_identity_node
         _atom_identity_symbol 1 C 2 C 3 C 4 C 5 C 6 C
save_
loop_ _molecular_fragments   $ethyl  $phenyl $methyl
```

A save frame has the following attributes:

(*a*) A save frame may contain data items and loop structures but not other save frames [see (*f*)].

(*b*) The scope of the data specified in a save frame is the save frame in which it is specified.

(*c*) Data values in a save frame are distinct from any identical items in the parent data block.

(*d*) A save frame may be referenced within the data block in which it is specified using a data item with a value of `$`*framecode*.

Example:

```
loop_ _amino_acid_seq
      _amino_acid_data   1 $tyr 2 $arg 3 $arg 4 $leu
```

where 'arg', 'tyr' and 'leu' are frame codes identifying three save frames of data.

(*e*) A frame code must be unique within a data block.

(*f*) A save frame may not contain another save frame, but it may contain references to other save frames in the same data block using frame codes.

2.1.3.7. Data block

A data block is a set of data containing any number of unique items and save frames. A data block begins with a `data_`*blockcode* statement, where *blockcode* is a unique identifying name within a file. A data block is closed by another `data_`*blockcode* statement, a `global_` statement or an end of file.

Example:

```
data_rhinovirus
all information relevant to rhinovirus included here

data_influenza
all information relevant to influenza virus
included here
```