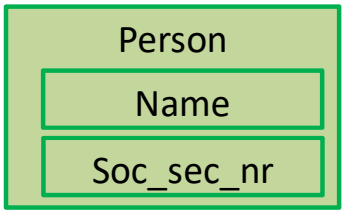
What is a Data Object?

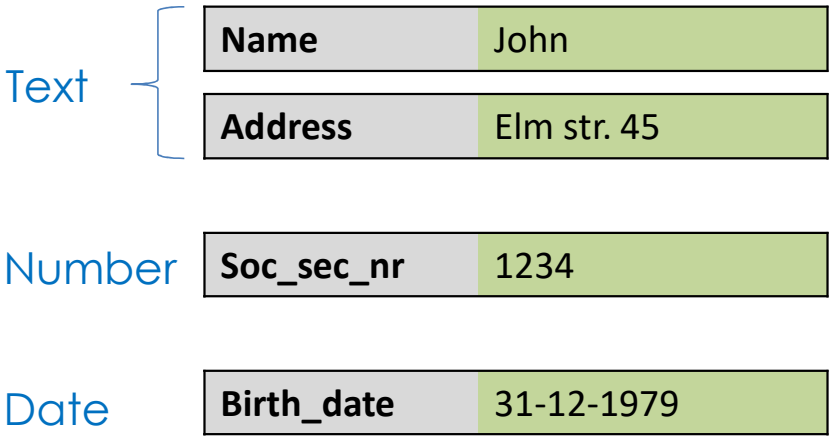
- A data object is storage for a piece of information.

- Types of data objects: text, number, date, and collection.
- Composite data objects can be created through nesting.
  - Person.name
  - Person.social_sec_nr



Session Data: Simple Data Objects

- In session, data is stored to data objects.
- Each simple data object stores a single value.



Session: Complex Data Objects

- Simple data objects can be grouped into a larger data object, which is also called a complex data object.

- Items can be referenced using the dot (.) Notation like:

  Person.Name = 'John'
  Person.Address.Street = 'Elm str.'
  Person.Soc_sec_nr = 1234

- Multiple levels of nesting is allowed.

- The 'Query collection'-part allows quick selection of data from complex data objects.

Person	
Name	John
Address	
Street	Elm str.
House nr.	45
Zip code	AA1111
Birth_date	31-12-1979
Soc_sec_nr	1234

Naming Conventions for Path/Data Objects

Build Smart Applications Fast

All variables are defined by their PATH. Path = [container_hierarchy].[object_name]
Container_hierarchy is OPTIONAL. Object_name is MANDATORY
Multi-level containers are allowed. A container for one variable cannot match the path for another.

Allowed

Multi-level containers [abc].[def].[xyz].object ✓

group1.object
group1.group2.object
group1.group2.group3.object

Not Allowed

[existing_path].object ✘

Container = Path

group1.object.object2
group1.group2.object.object2

containers (shaded yellow) already exists as path above

LogicNets

Example Data Model

Build Smart Applications Fast

ALLOWED

section1.summary.status
section1.part_A.status
section1.part_A.detail
section1.part_A.action
section1.part_A.notes
section1.part_B.status
section1.part_B.detail
section1.part_B.action
section1.part_B.notes

Model:
[section#].[part#].[data-type]

NOT ALLOWED

section1.status
section1.status.part_A
section1.status.part_B

section1.status.part_A.detail
section1.status.part_A.action
section1.status.part_A.notes
…

Yellow signifies data variables that will not hold a value

LogicNets

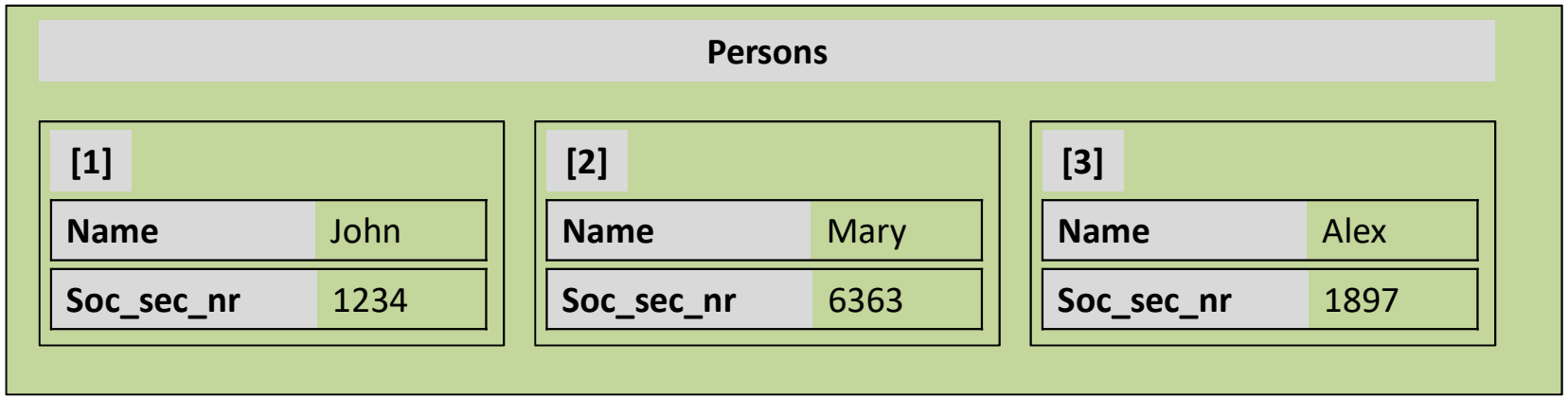
Naming Rules/Conventions for Variables

- Containers and data objects cannot contain spaces (e.g. first_name vs first name).
- When naming data objects you should avoid starting with numbers (e.g. 1_abc), or using special characters: %, &, $, 'period' etc. – they can all cause trouble.
- When creating data object names LogicNets recommends the following
  - using all lower case text (but note capitalization IS allowed if deemed necessary)
  - using longer names as they are easier to understand;
    e.g. *number_of_business_units* instead of *no_bus*.
  - using '_' (underscore) to separate meaningful words
    e.g. Object_Name instead of Object_name, objectName, or OBJECT_NAME

- Avoid special words used by LogicNets codebase: *and, break, do, else, elseif, end, false, for, function, if, in, local, nil not ,or, repeat, return, then, true, until, while*

Session: Collections

- Data objects can be put into a collection/array.



- Items in an array can be referenced using the [] notation like:
  Persons[2].Name = 'Mary'
- Hybrid arrays are supported.
- Collections parts allow the creation and manipulation of collections.

$()

- For printing the data of data objects you can use the syntax:

  $(<path of your data object>) notation.

- $() is a powerful mechanism that is used throughout LogicNets to print data to screen or to resolve variable names dynamically.

  – Default values for undefined objects can be set using

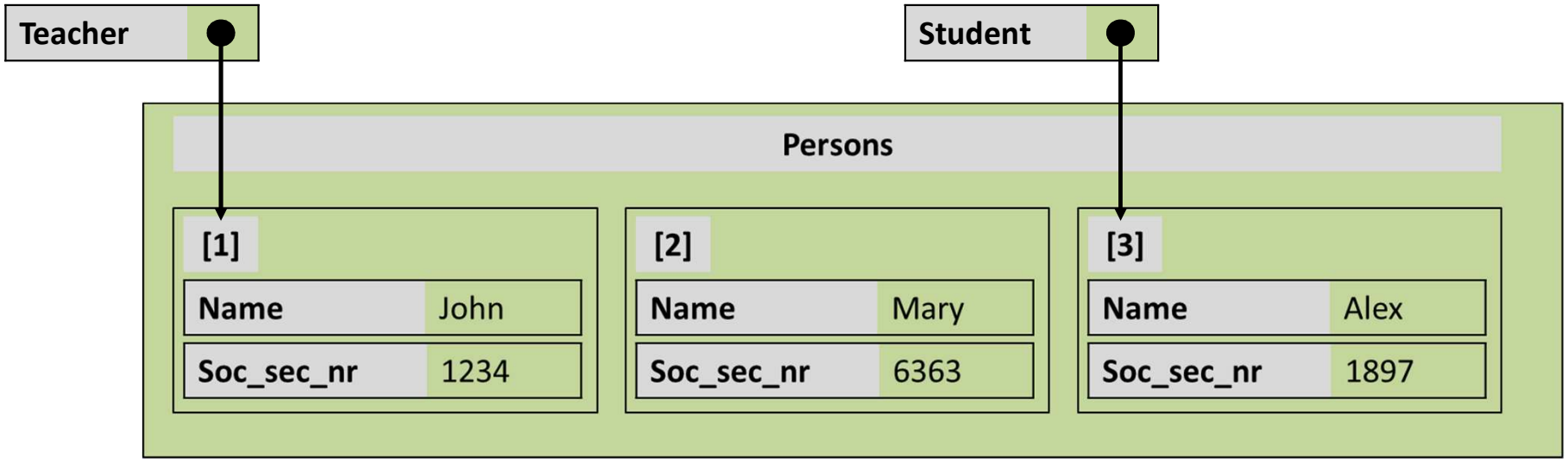
    $(<path>:<default>) notation, e.g. $(myobject:100)

  – The $() construction can be used for the default value

    e.g. $(myobject:$(default_value))

Session Reference to Data Objects

- Sometimes it is also possible to obtain a reference to a data object.



- Persons[1].Name = Teacher.Name = 'John'
- Persons[3].Name = Student.Name = 'Alex'