Rotman

Master of Management Analytics

INTRO TO SAS PROGRAMMING

Bootcamp SAS 2 (4 hours)

August 9, 2019 Prepared by Jay Cao / TDMDL

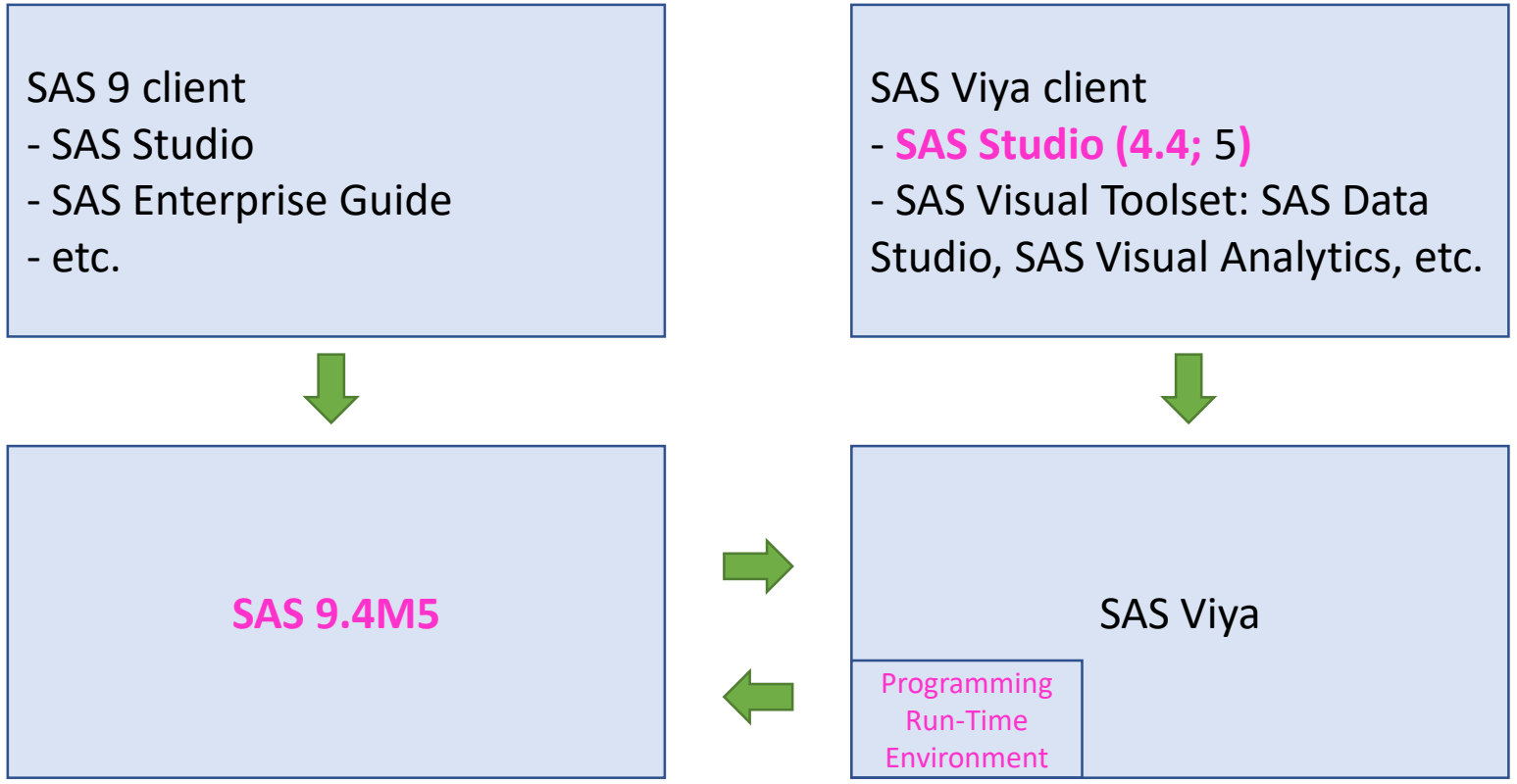
Rotman School of Management
UNIVERSITY OF TORONTO

Plan and Goal

- Plan
  - **SAS programming essentials (SAS 9)**
  - SAS Viya programming overview

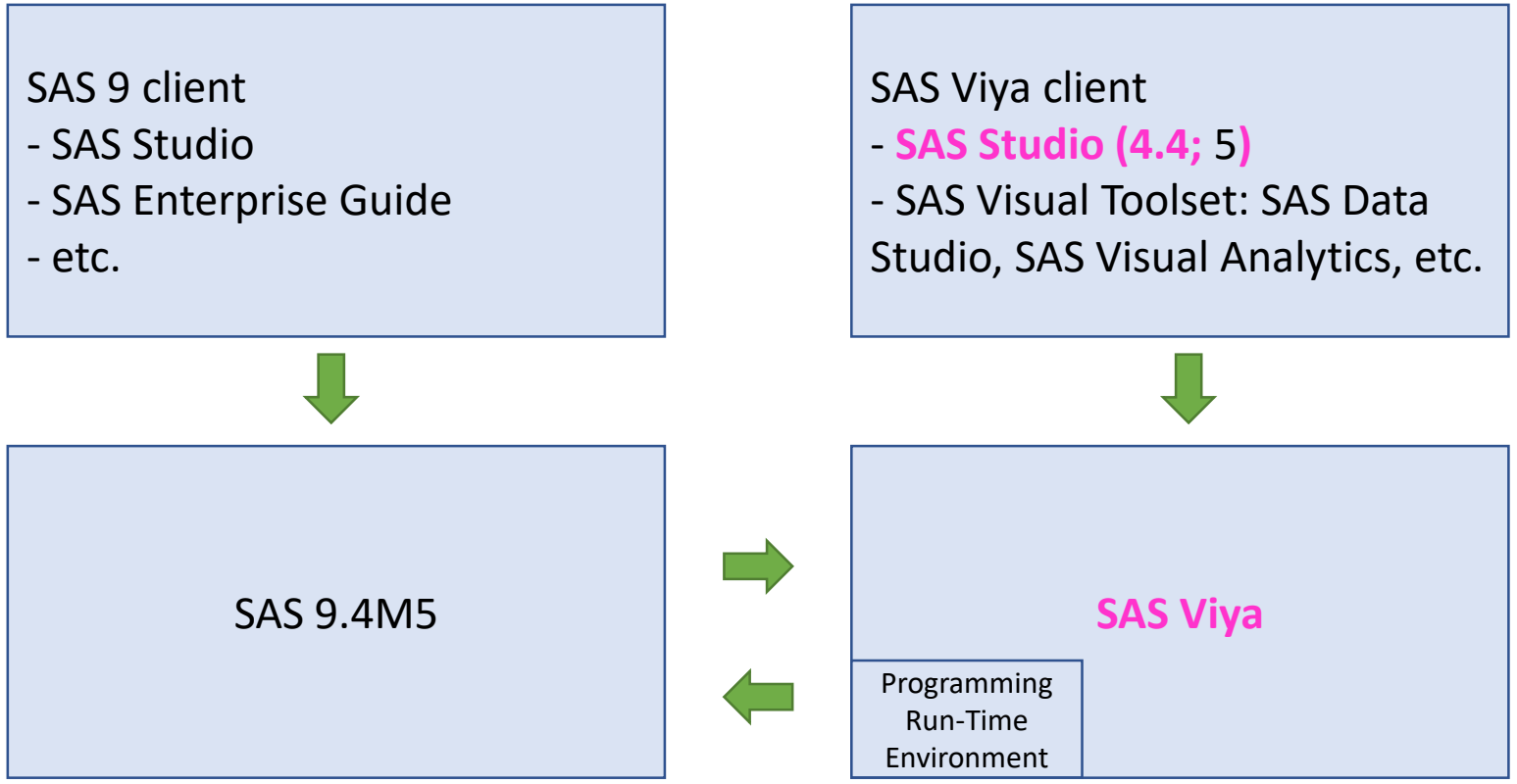
- Goal: learning the basics of
  - SAS programming language
  - Analysis workflow

SAS 9 vs SAS Viya (Workshop 2)



Ref. 1) SAS 9 and SAS Viya; 2) SAS 9 and SAS Viya Functional Comparison

SAS 9 vs SAS Viya (if time we have time)



Ref. 1) SAS 9 and SAS Viya; 2) SAS 9 and SAS Viya Functional Comparison

Learning Resources

- Coursera SAS Programming Specialization (3 courses)
  - Click Enroll and then Audit for free access
  - The first 2 courses corresponds to the following courses offered by SAS

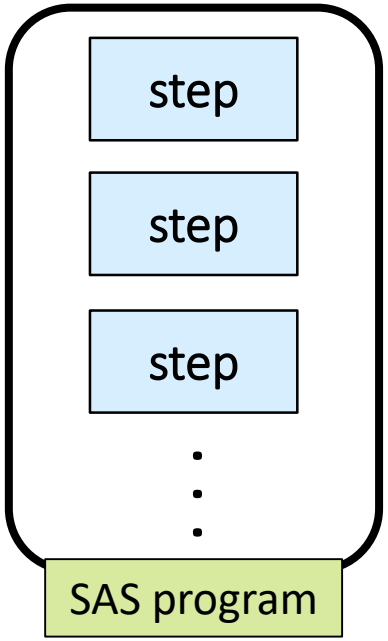
- Course Offered by SAS
  - SAS Programming 1: Essentials (free)
  - SAS Programming 2: Data Manipulation Techniques (not free)

Programming Environment: SAS Studio

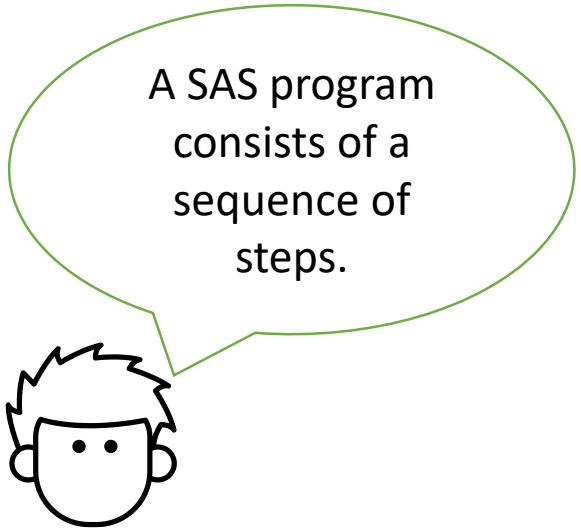
- Login to SAS Studio
  - rac.rotman.utoronto.ca

- SAS Studio Point-and-Click
  - SAS video tutorial: Getting Started with SAS Studio

SAS Program Structure



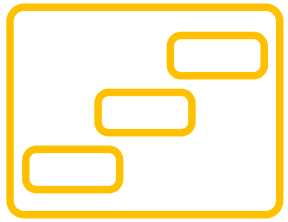

```
data myclass;
     set sashelp.class;
     heightcm=height*2.54;
run;

proc print data=myclass;
run;

proc means data=myclass;
     var age heightcm;
run;
```



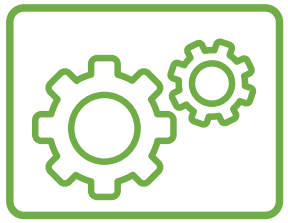
Ref. SAS Programming 1: Essentials

SAS Program Structure

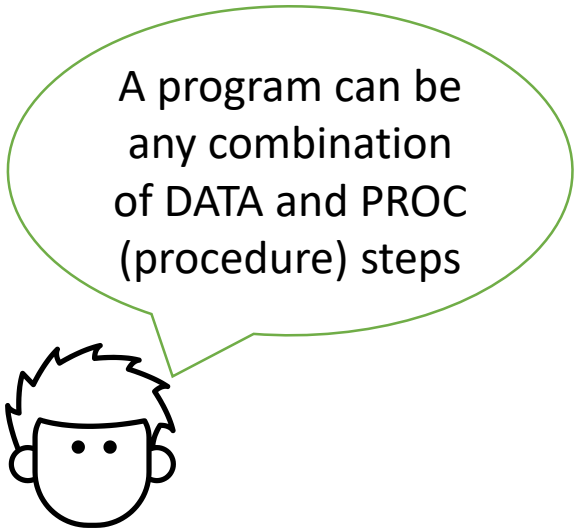
DATA step



PROC step



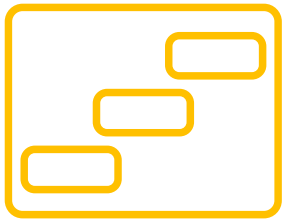
```
data myclass;
     set sashelp.class;
     heightcm=height*2.54;
run;

proc print data=myclass;
run;

proc means data=myclass;
     var age heightcm;
run;
```



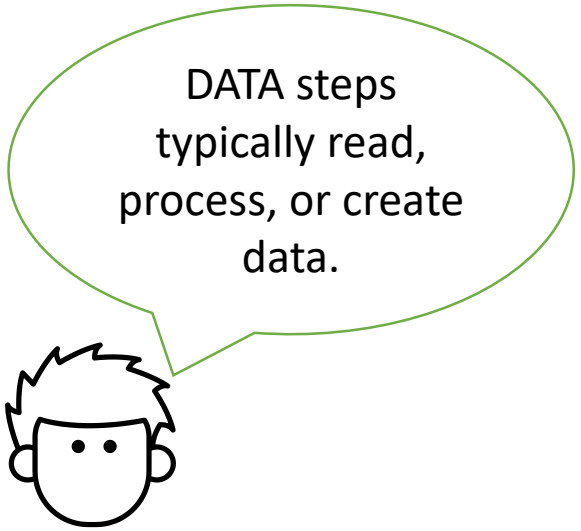
Ref. SAS Programming 1: Essentials

SAS Program Structure

DATA step



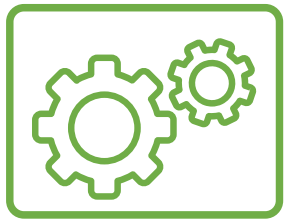
```
data myclass;
     set sashelp.class;
     heightcm=height*2.54;
run;

proc print data=myclass;
run;

proc means data=myclass;
     var age heightcm;
run;
```



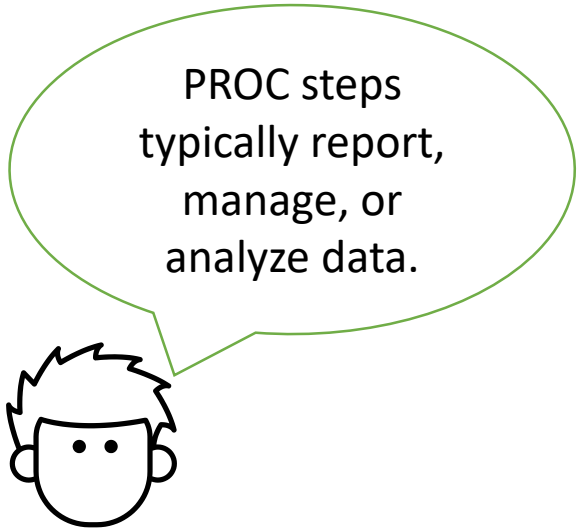
Ref. SAS Programming 1: Essentials

SAS Program Structure

PROC step



```
data myclass;
     set sashelp.class;
     heightcm=height*2.54;
run;

proc print data=myclass;
run;

proc means data=myclass;
     var age heightcm;
run;
```



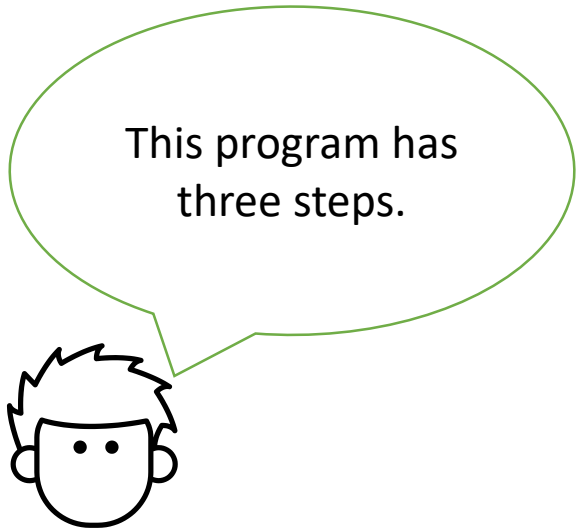
Ref. SAS Programming 1: Essentials

SAS Program Structure

Steps begin with either DATA or PROC.

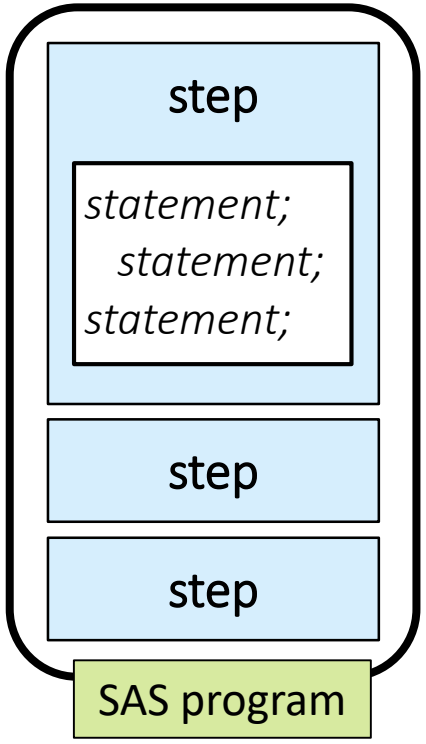
Steps end with RUN. Some PROCs end with QUIT.

```
data myclass;
     set sashelp.class;
     heightcm=height*2.54;
run;

proc print data=myclass;
run;

proc means data=myclass;
     var age heightcm;
run;
```

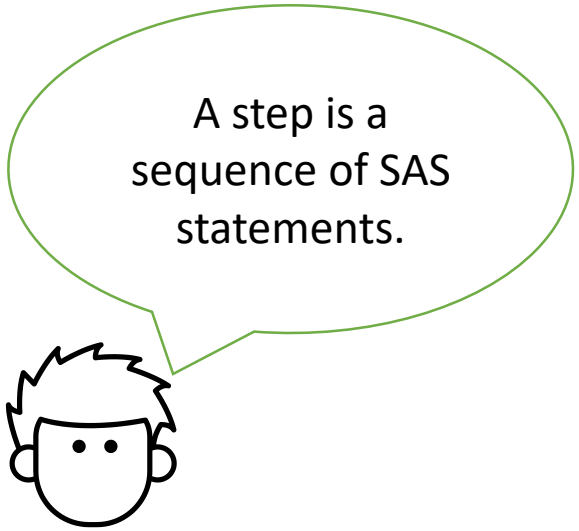


Ref. SAS Programming 1: Essentials

SAS Program Structure



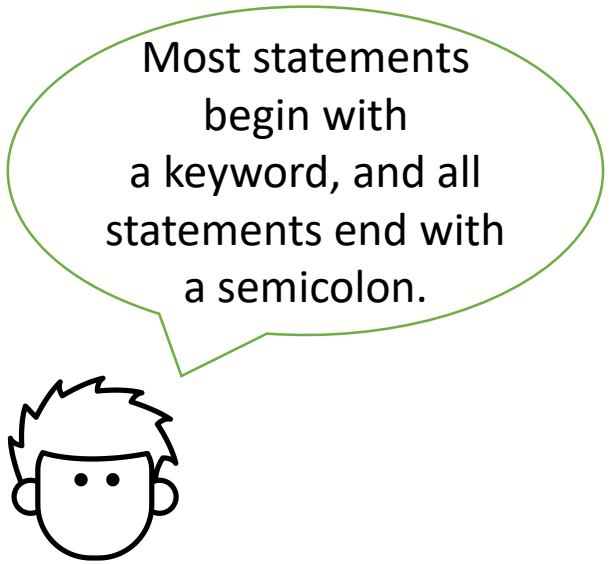

```
data myclass;
    set sashelp.class;
    heightcm=height*2.54;
run;

proc print data=myclass;
run;

proc means data=myclass;
    var age heightcm;
run;
```



Ref. SAS Programming 1: Essentials

SAS Statement Syntax

```
data myclass;
     set sashelp.class;
     heightcm=height*2.54;
run;

proc print data=myclass;
run;

proc means data=myclass;
     var age heightcm;
run;
```



Ref. SAS Programming 1: Essentials

Global Statements

TITLE . . . ;

OPTIONS . . . ;

LIBNAME . . . ;

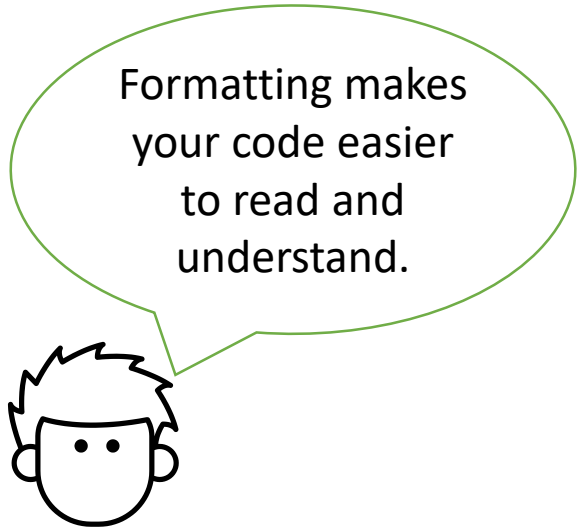
Ref. SAS Programming 1: Essentials

SAS Program Syntax: Format

These are the same to SAS.

```
data myclass;set sashelp.class;run;
proc print data=myclass;run;
```

```
data myclass;
    set sashelp.class;
run;

proc print data=myclass;
run;
```



Ref. SAS Programming 1: Essentials

SAS Program Syntax: Case

```
data under13;
     set sashelp.class;
     where AGE<13;
     drop heIGht Weight;
run;
```

Unquoted values can be in any case.

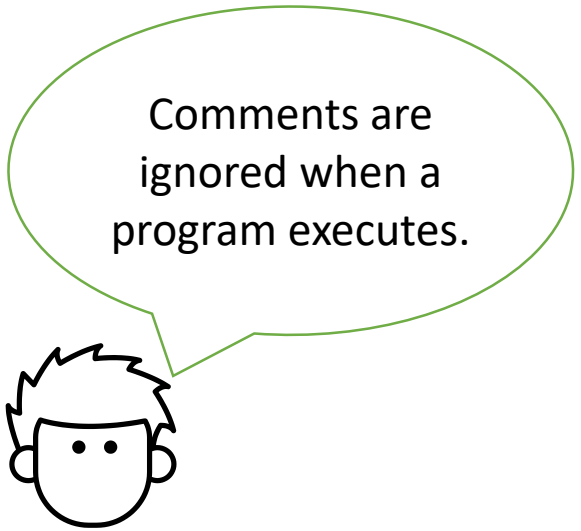
Ref. SAS Programming 1: Essentials

SAS Program Syntax: Comments

```
/* students under 13 yo */

data under13;
     set sashelp.class;
     where Age<13;
     *drop Height Weight;
run;
```

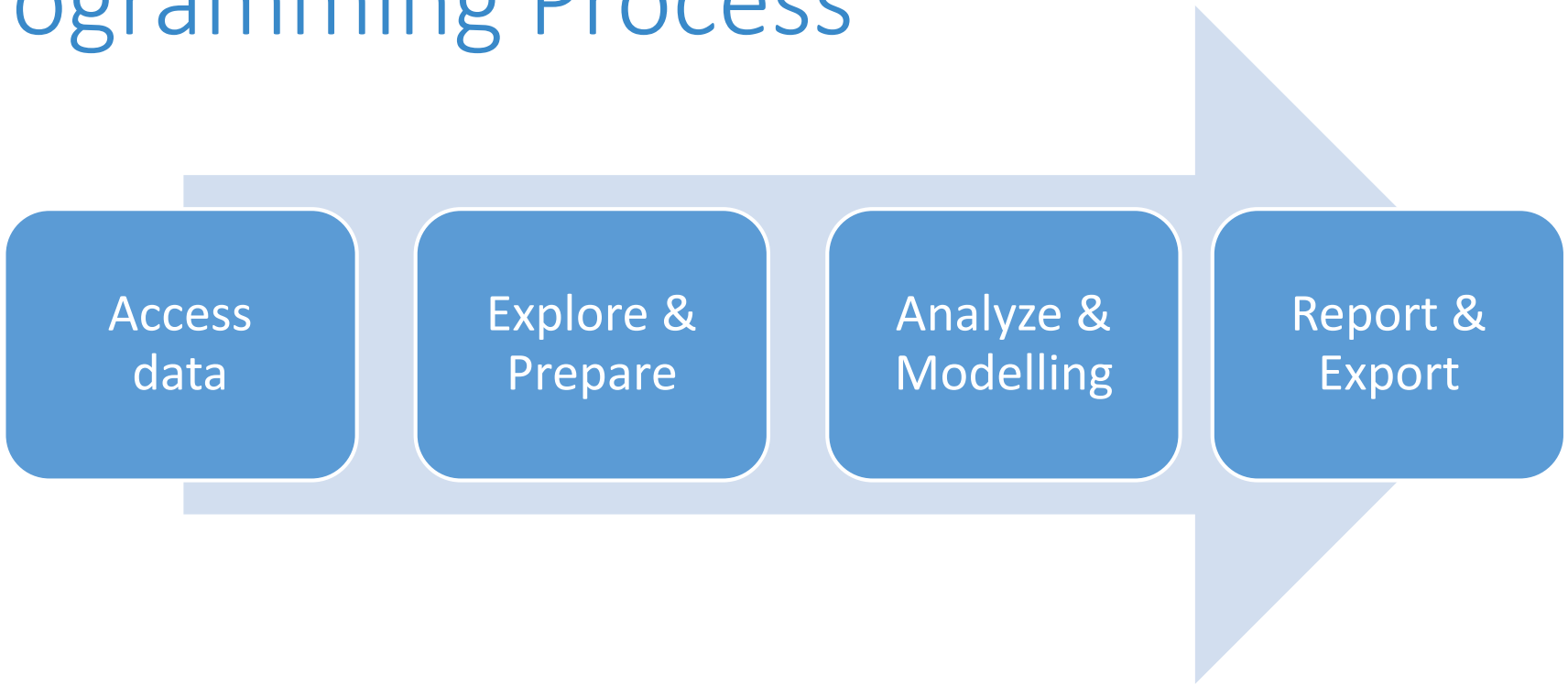
comments out everything between /* and */

comments out a single statement ending in a semicolon



Ref. SAS Programming 1: Essentials

SAS Programming Process



Ref. SAS Programming 1: Essentials

SAS Programming Process

This workshop

PROC IMPORT	**PROC CONTENTS**	**PROC REG**	**PROC …**
LIBNAME	**PROC PRINT**	**PROC LOGISTIC**	
	PROC MEANS		**PROC EXPORT**
	PROC UNIVARIATE		
	PROC FREQ		
	DATA STEP		
	PROC SQL		

Ref. SAS Programming 1: Essentials

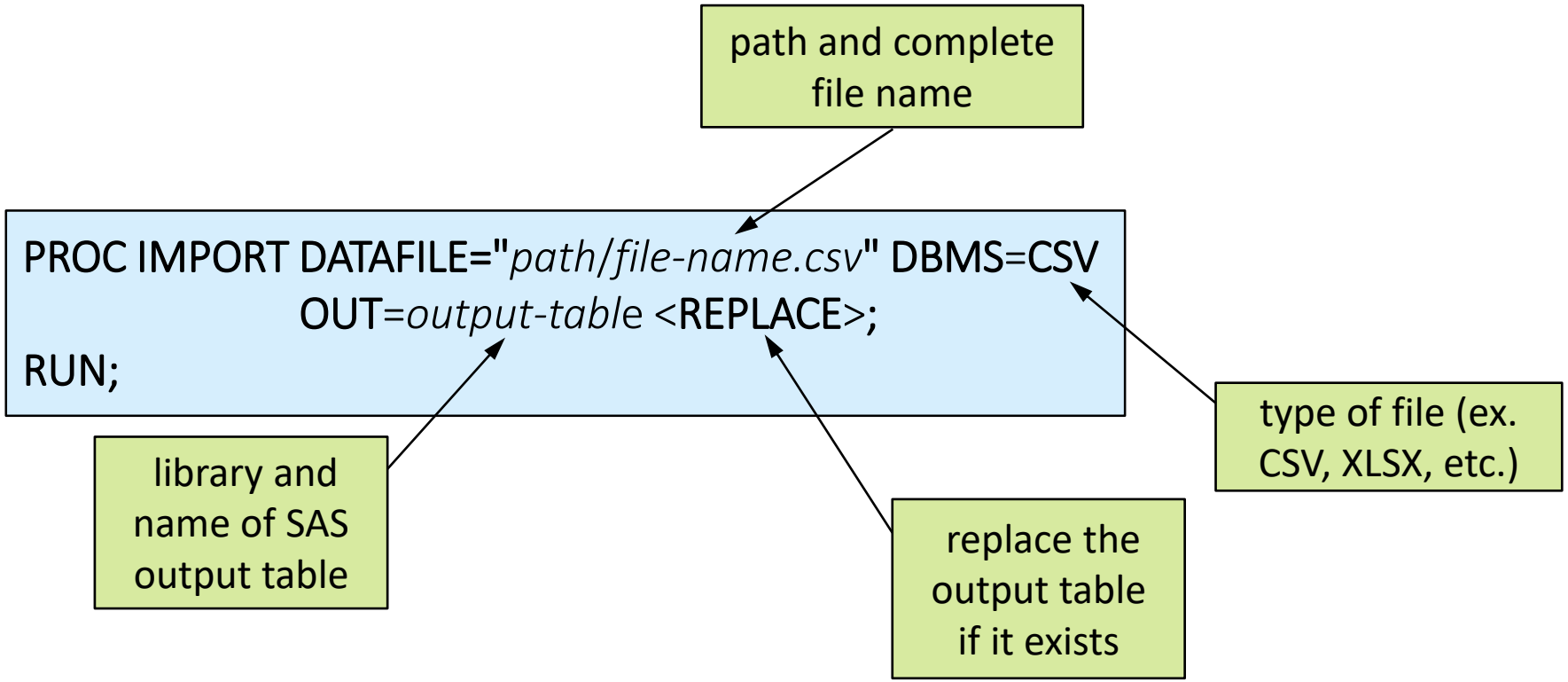
SAS Programming Process

This workshop

Access data	Explore & Prepare	Analyze & Modelling	Report & Export
PROC IMPORT	**PROC CONTENTS**	**PROC REG**	**PROC …**
LIBNAME	**PROC PRINT**	**PROC LOGISTIC**	
	PROC MEANS		**PROC EXPORT**
	PROC UNIVARIATE		
	PROC FREQ		
	DATA STEP		
	PROC SQL		

Ref. SAS Programming 1: Essentials

Import a Comma-Delimited (CSV) File



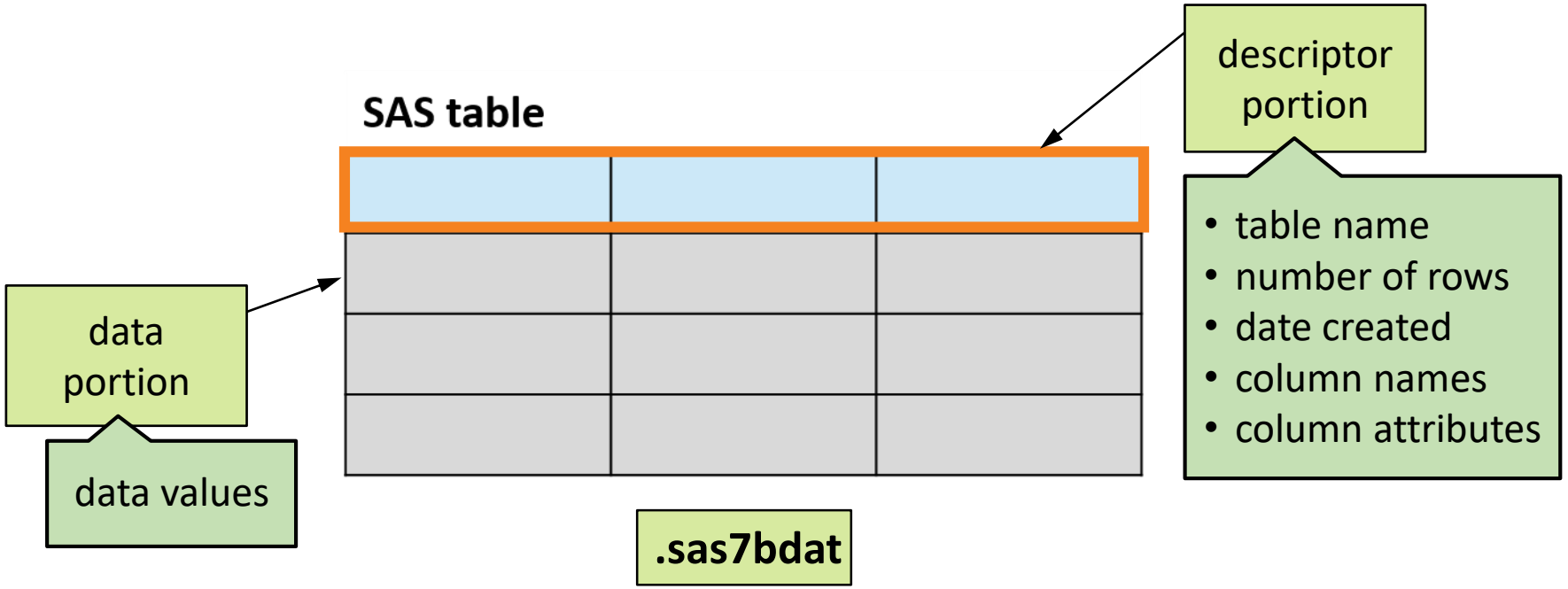
Ref. 1) SAS Programming 1: Essentials; 2) PROC IMPORT document

Import a CSV File (hands-on)

- Create folders (~/Workshop_SAS and ~/Workshop_SAS/data)
- Upload csv files
- Import Employees.csv to SAS
  - Where does the SAS data go?
  - What's a SAS table?
  - What's a library?

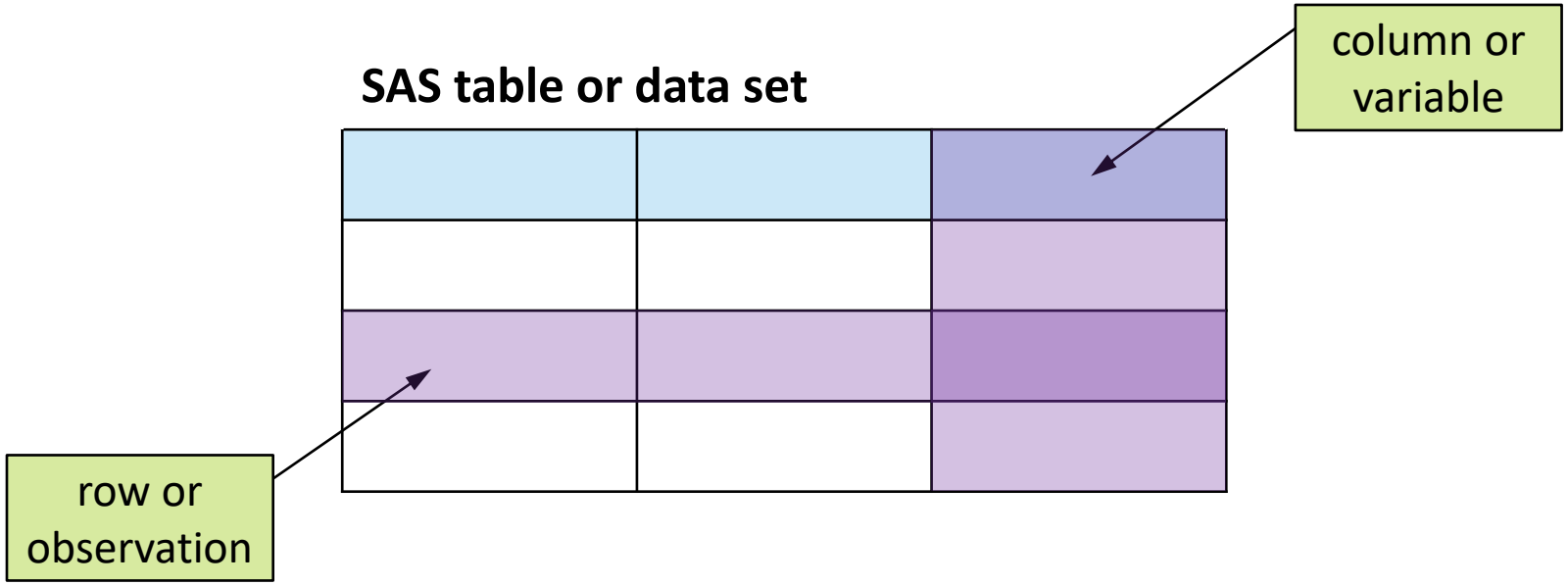
Ref. PROC IMPORT document

What's a SAS Table (1)



Ref. SAS Programming 1: Essentials

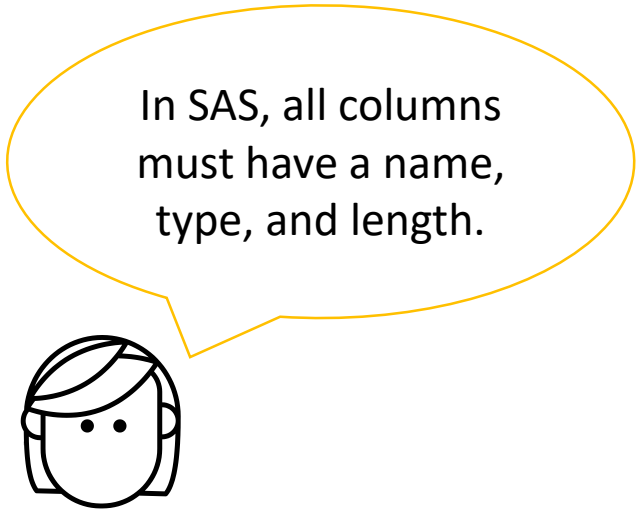
What's a SAS Table (2)



Ref. SAS Programming 1: Essentials

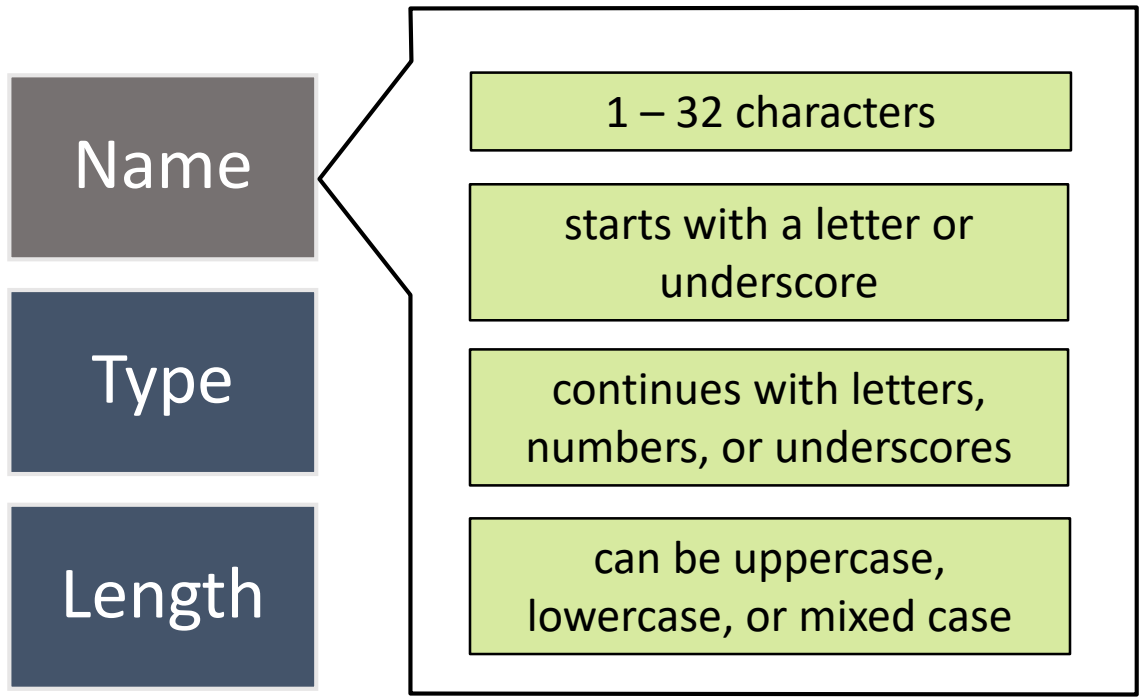
Column Attributes

- Name
- Type
- Length



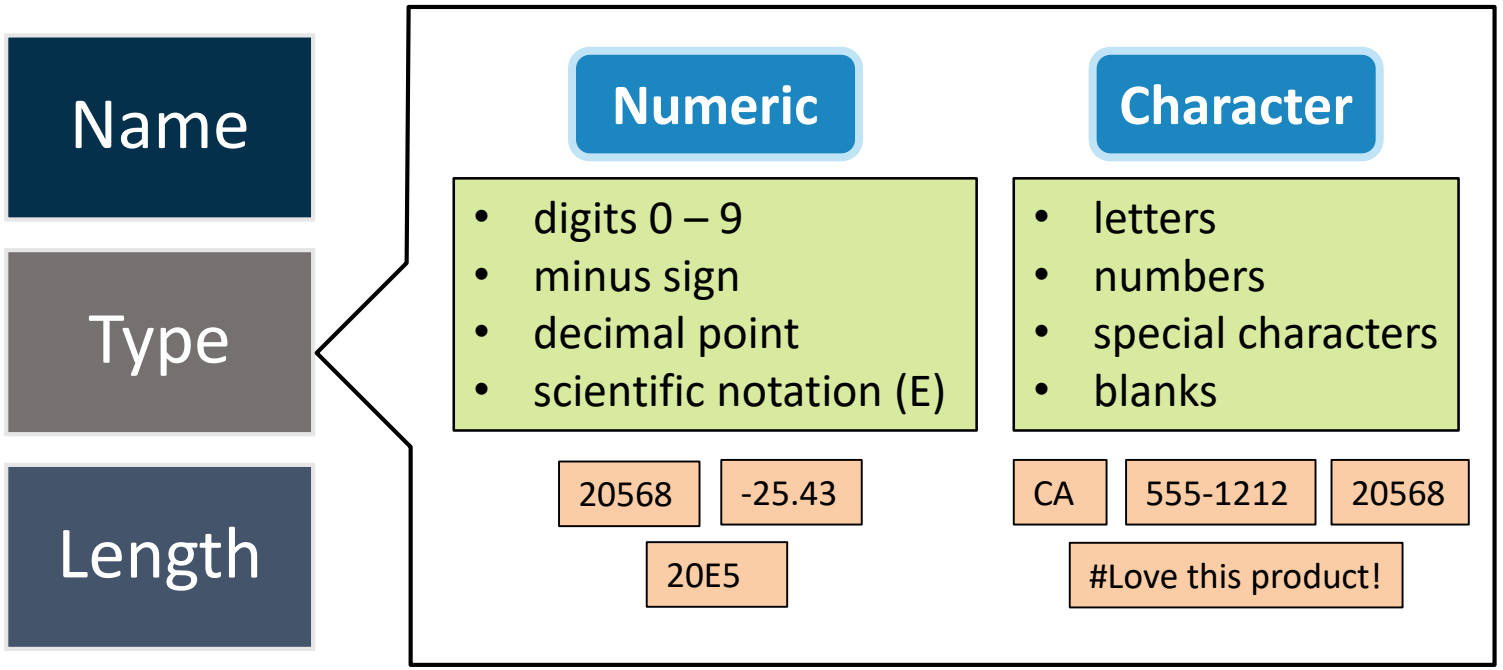
Ref. SAS Programming 1: Essentials

Column Attributes: Name



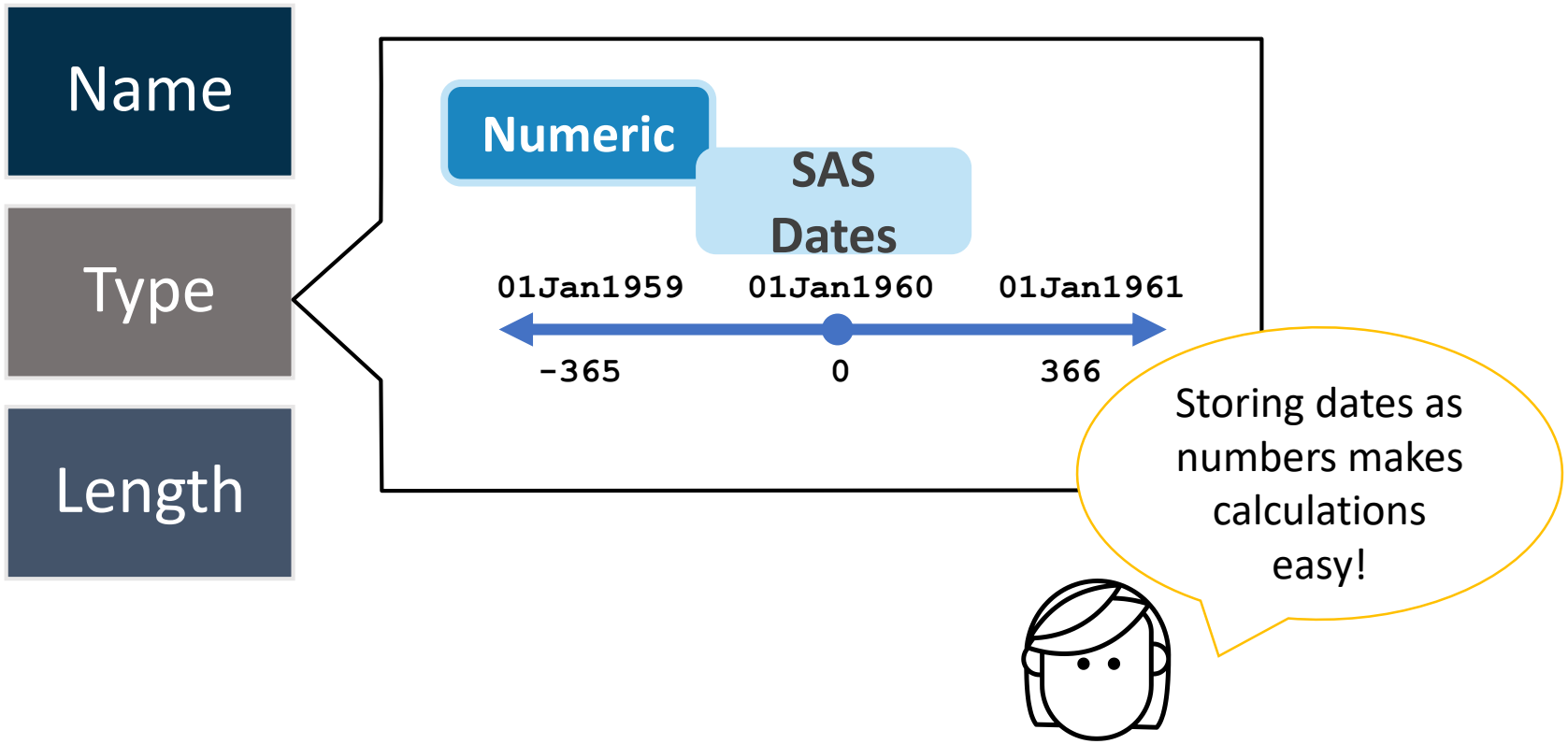
Ref. SAS Programming 1: Essentials

Column Attributes: Type



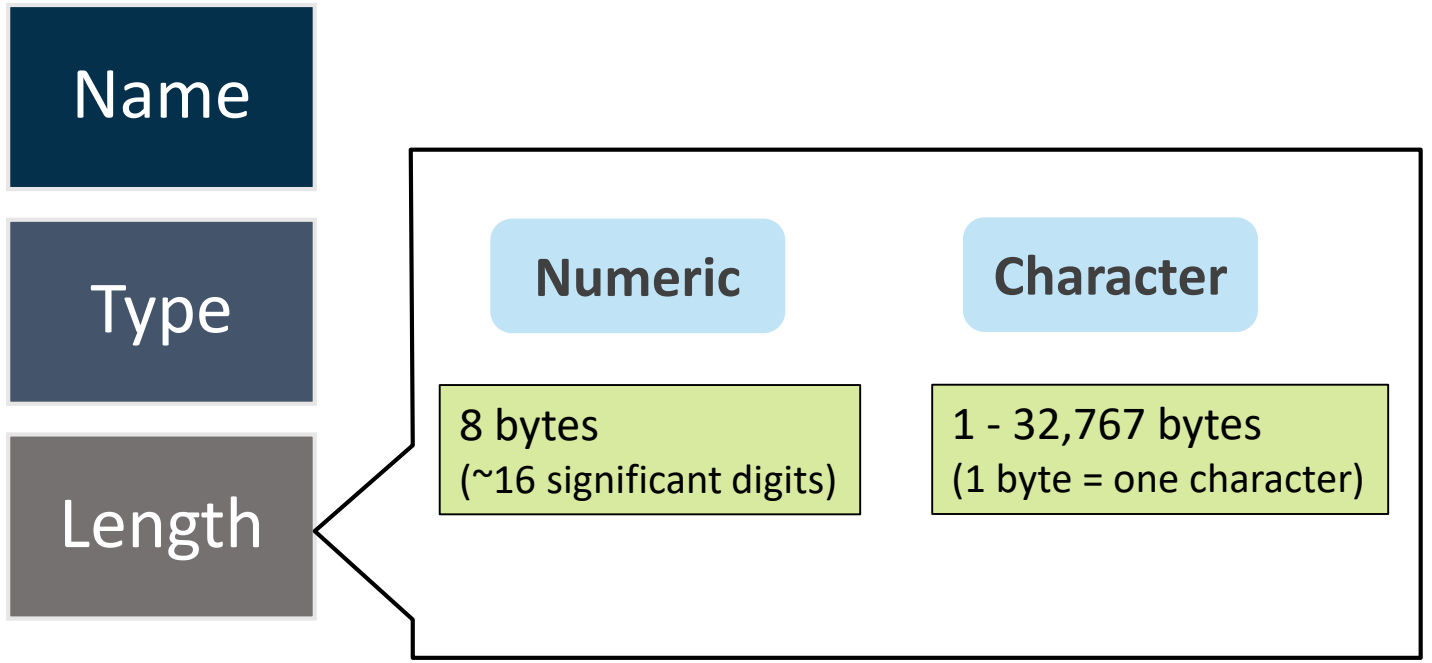
Ref. SAS Programming 1: Essentials

Column Attributes: Type



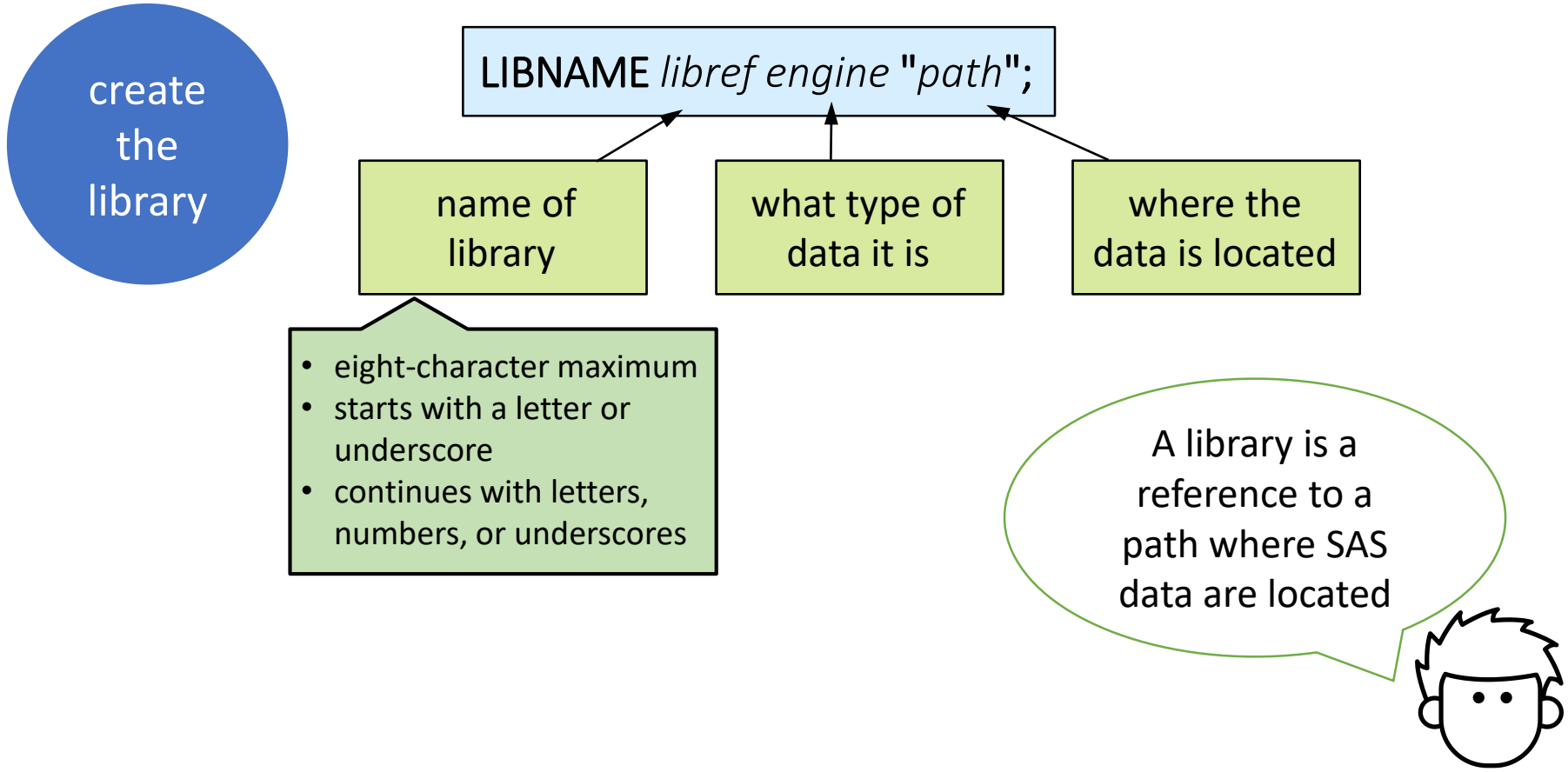
Ref. SAS Programming 1: Essentials

Column Attributes: Length



Ref. SAS Programming 1: Essentials

Use SAS Library (1)

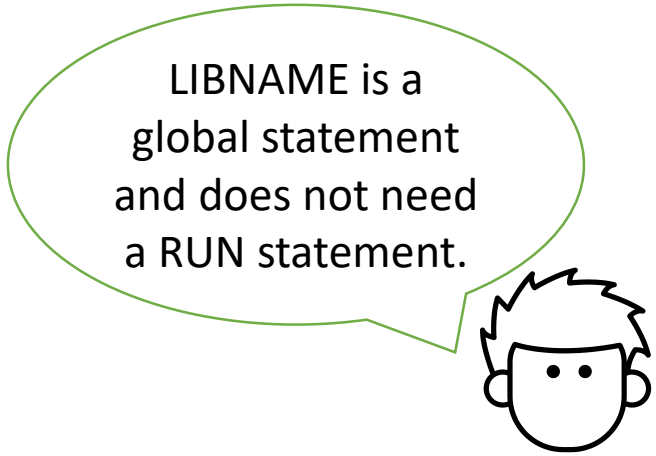


Ref. 1) SAS Programming 1: Essentials; 2) LIBNAME document

Use SAS Library (2)

```
libname mylib base "~/Workshop_SAS/data";
```

library name **mylib**	Base SAS tables	data located in ~/Workshop_SAS/data/



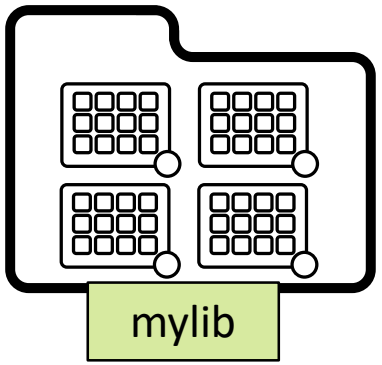
Ref. SAS Programming 1: Essentials

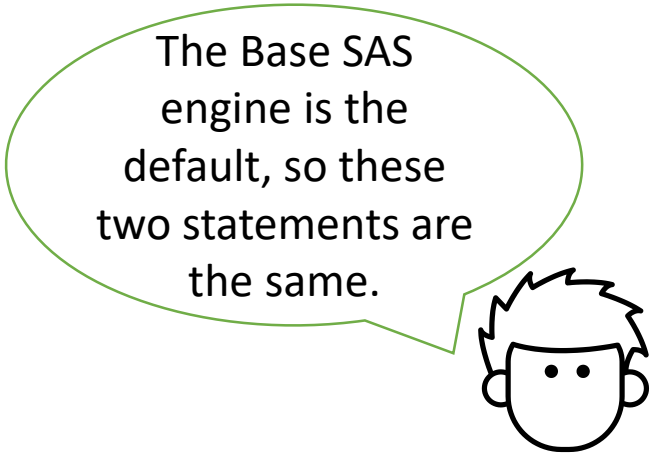
Use SAS Library (3)

create the library

```
libname mylib base "~/Workshop_SAS/data";
```

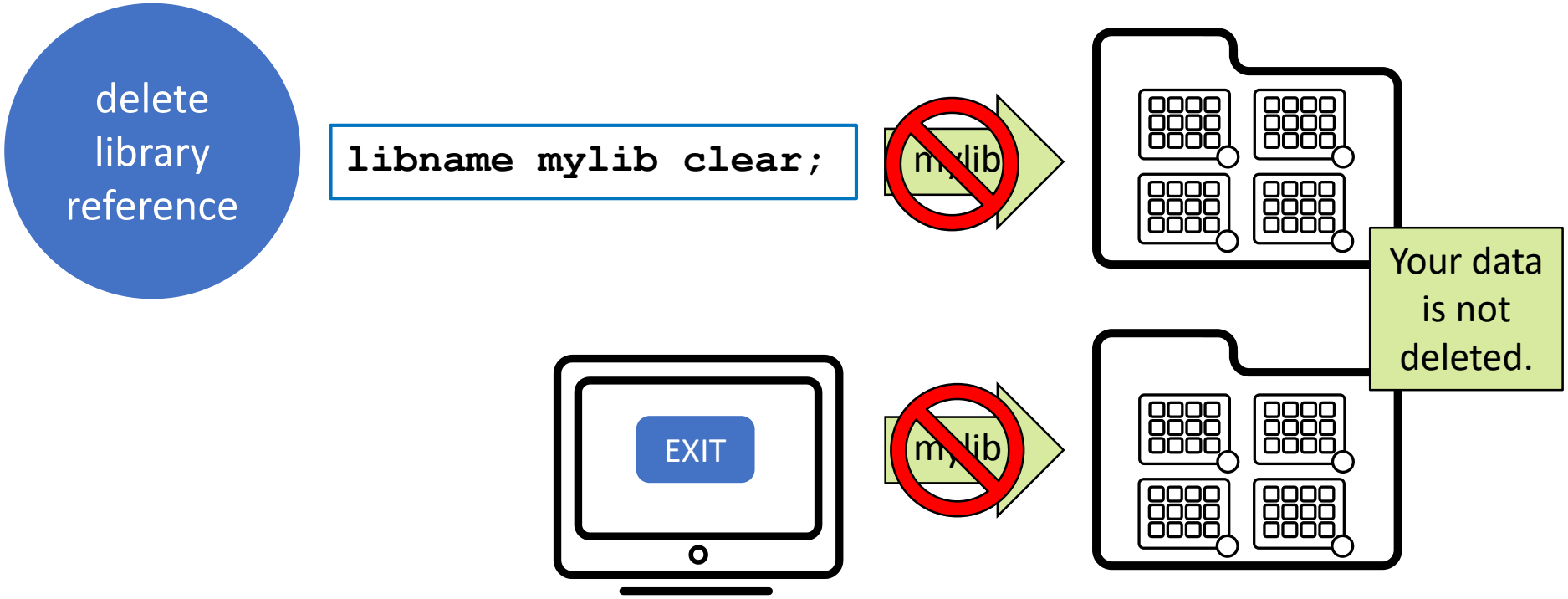
```
libname mylib "~/Workshop_SAS/data";
```





Ref. SAS Programming 1: Essentials

Use SAS Library (4)



Ref. SAS Programming 1: Essentials

Use SAS Library (hands-on)

- Create **`mylib`** library referencing to **`~/Workshop_SAS/data`**
- Import **`Employees.csv`** to **`mylib`**

- Note: defining a library is a way to read in SAS data files

SAS Programming Process

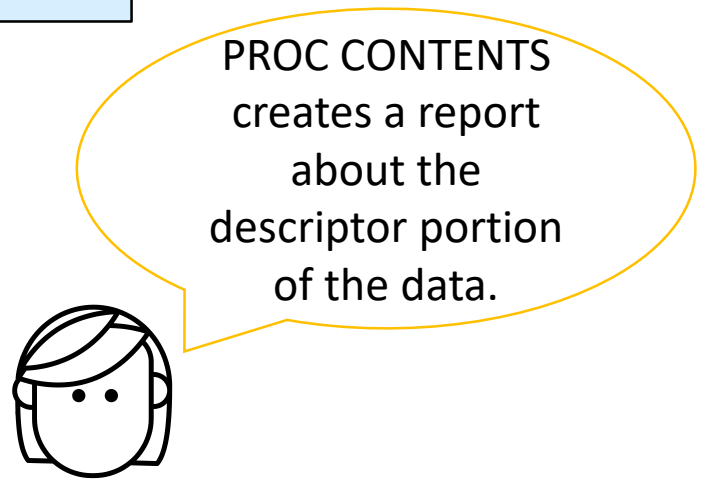
This workshop

PROC IMPORT	**PROC CONTENTS**	**PROC REG**	**PROC …**
LIBNAME	**PROC PRINT**	**PROC LOGISTIC**	
	PROC MEANS		**PROC EXPORT**
	PROC UNIVARIATE		
	PROC FREQ		
	DATA STEP		
	PROC SQL		

Ref. SAS Programming 1: Essentials

View Table and Column Attributes

```
PROC CONTENTS DATA=data-set;
RUN;
```



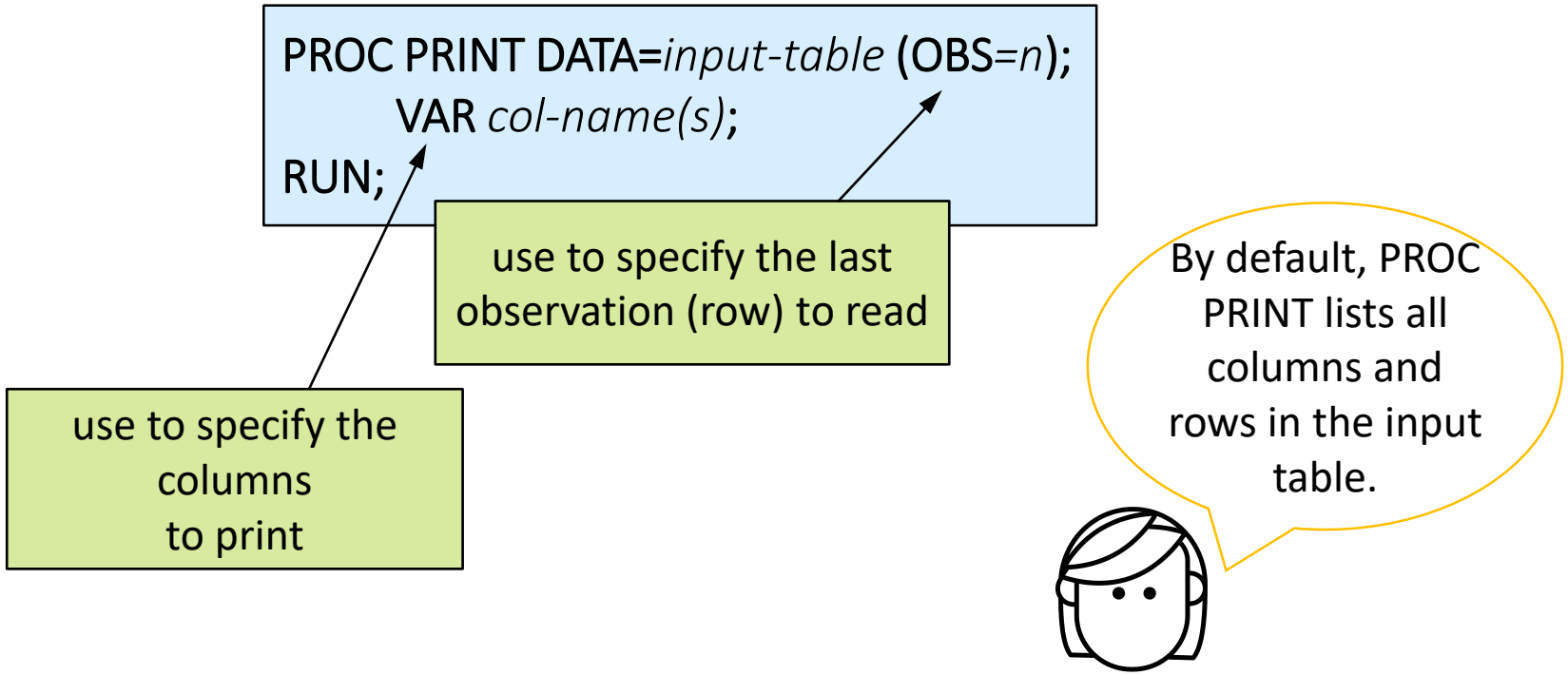
Ref. SAS Programming 1: Essentials

View Table and Column Attributes (hands-on)

- View **`mylib.employees`** table and column attributes
  - What type does column/variable **`BirthDate`** have?
  - What's format and informat?

Ref. formats and informats

PRINT Procedure



Ref. 1) SAS Programming 1: Essentials; 2) PROC PRINT Document

MEANS Procedure

```
PROC MEANS DATA=input-table;
      VAR col-name(s);
RUN;
```

use to specify the numeric columns to analyze

By default, PROC MEANS generates simple summary statistics for each numeric column in the input data.

Ref. 1) SAS Programming 1: Essentials; 2) PROC MEANS document

UNIVARIATE Procedure

```
PROC UNIVARIATE DATA=input-table;
        VAR col-name(s);
RUN;
```

use to specify the numeric columns to analyze

By default, PROC UNIVARIATE generates summary statistics for each numeric column in the input data.

Ref. 1) SAS Programming 1: Essentials; 2) PROC UNIVARIATE document

FREQ Procedure

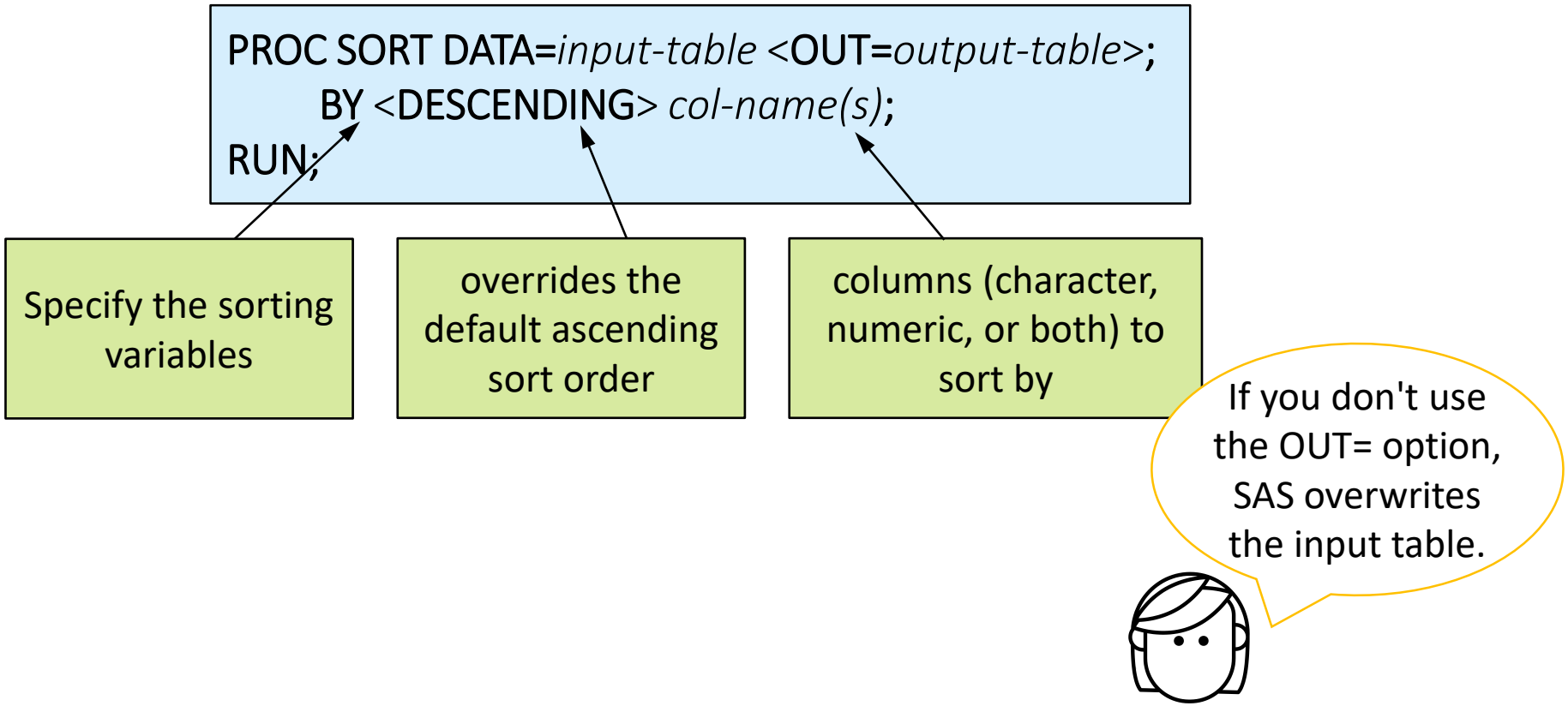
```
PROC FREQ DATA=input-table;
        TABLES col-name(s);
RUN;
```

use to specify the frequency tables to include in the results

By default, PROC FREQ creates a frequency table for each column in the input table.

Ref. 1) SAS Programming 1: Essentials; 2) PROC FREQ document

SORT Procedure



Ref. 1) SAS Programming 1: Essentials; 2) PROC SORT document

PROC MEANS and SORT (hands-on)

- Create a summary statistics table
  - Use **`sashelp.cars`** dataset and columns **`EngineSize`** and **`Hoursepower`**

- Find all (unique) baseball division and league
  - Use **`sashelp.baseball`** dataset
  - Only keep columns **`division`** and **`league`**
  - Use PROC SORT to remove duplicate
  - Write output to work.DL table
  - Hint: 1); PROC SORT extra syntax; 2) Example 7