Statistical Analysis System

SAS
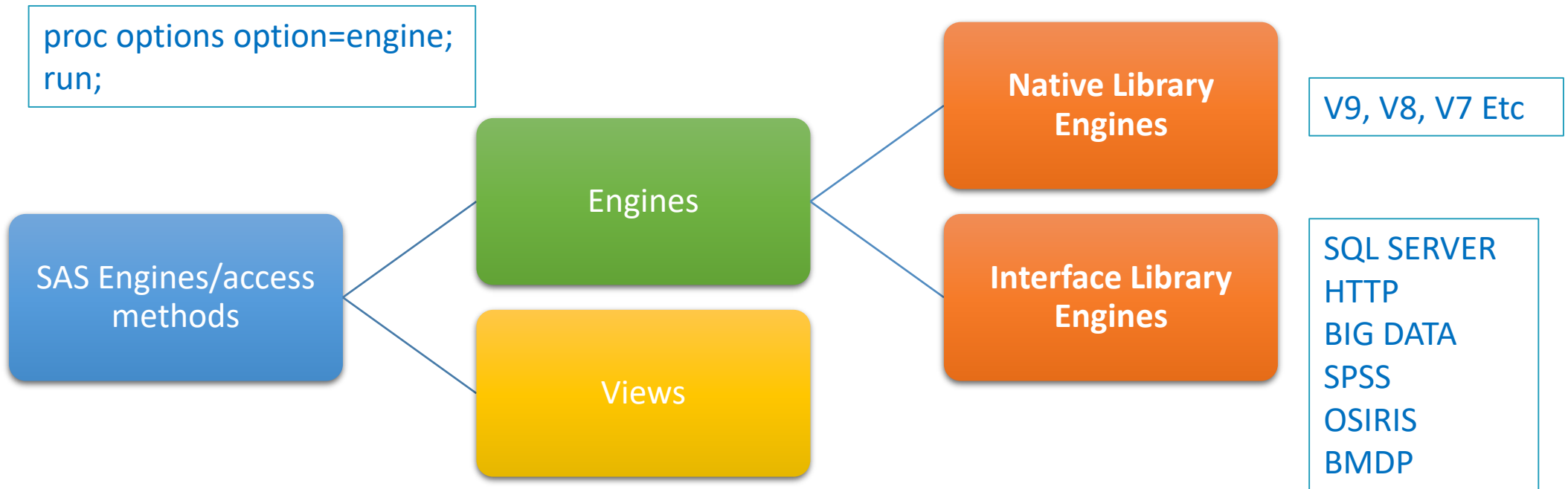
Brief History

Anthony J. Barr

Jim Goodnight

➢ **Sept 1962 -** Began assistantship with North Carolina State University Computing Center. Program was written on IBM 1410 assembler.

➢ **June 1976 -** SAS Institute, Inc. was incorporated. (Release of 1976 version of SAS.)
https://www.sas.com/en_us/home.html

SAS ARCHITECTURE

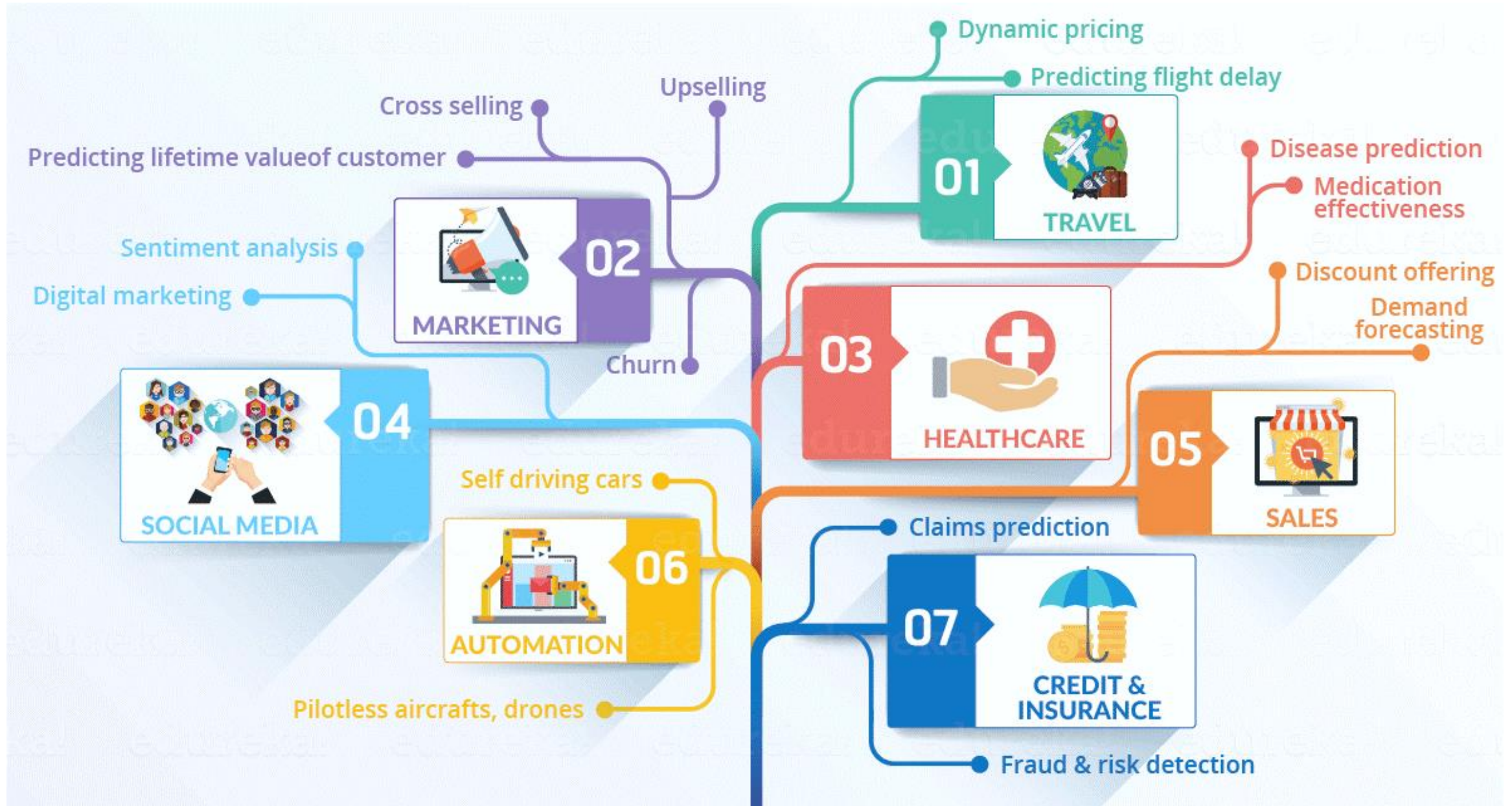➢ Multi Vendor Architecture (SAS run in almost all platforms)

| IBM Mainframes | Microsoft Windows | UNIX | LINUX | z/OS | Etc |

➢ Multi Engine Architecture.

```
proc options option=engine;
run;
```

SAS Engines/access methods

Engines

Views

Native Library Engines

V9, V8, V7 Etc

Interface Library Engines

SQL SERVER
HTTP
BIG DATA
SPSS
OSIRIS
BMDP

INTRODUCTION

Dynamic pricing

Predicting flight delay

01 TRAVEL

Disease prediction

Medication effectiveness

Cross selling

Upselling

Predicting lifetime valueof customer

Sentiment analysis

Digital marketing

02 MARKETING

Churn

Discount offering

Demand forecasting

03 HEALTHCARE

04 SOCIAL MEDIA

05 SALES

Self driving cars

06 AUTOMATION

Pilotless aircrafts, drones

Claims prediction

07 CREDIT & INSURANCE

Fraud & risk detection

COMPONENTS OF SAS PROGRAMMING

i. Base SAS

It is a basis for all SAS software. SAS has hardware agility and integrates into any computer environment. It is easy to learn and is not code intensive. It provides data ease in data readability and interpretation.

ii. SAS/GRAPH

It a data visualization technique. It helps represent structured data into graphs by enhancing the process of interpretation.

iii. SAS/STAT

It helps us perform various statistical analysis variance, regression, and psychometric analysis. Some statistical techniques like such as ANOVA procedures are specially strung in the SAS environment.

iv. SAS/ETS

It is specially used for forecasting and for the time series analysis.

v. SAS/IML

IML is called Interactive Matrix Language. It is used to translate mathematical formulas into an innovative program for matrix computation and optimization.

vi. SAS/INSIGHTS and Enterprise Miner

SAS/INSIGHTS and Enterprise Miner are used for data mining.

FEATURES OF SAS PROGRAMMING

Strong Data Analysis Abilities

Flexible 4 Generation Programming Language (4GL)

SAS Studio

Support for Various Types of Data Format

Management

Report Output Format

Data Encryption Algorithms

ADVANTAGES OF SAS

Easy to learn	Ability to handle large database	Easy to debug	Tested algorithms	SAS Customer support
	Data Security	SAS GUI	Nice Output	Huge Job Prospects

DISADVANTAGES OF SAS

Cost	SAS is not open source	Lack of graphic representation	Difficult Text Mining	Difficult than R

DIFFERENT TYPES OF SAS SESSIONS

You can run SAS in any of several ways that might be available for your operating environment:

➢ SAS windowing environment
➢ Interactive line mode
➢ Non-Interactive mode
➢ Batch (or background) mode

In addition,

➢ SAS/ASSIST software provides a menu-driven system for creating and running your SAS programs.

➢ Object Server Mode
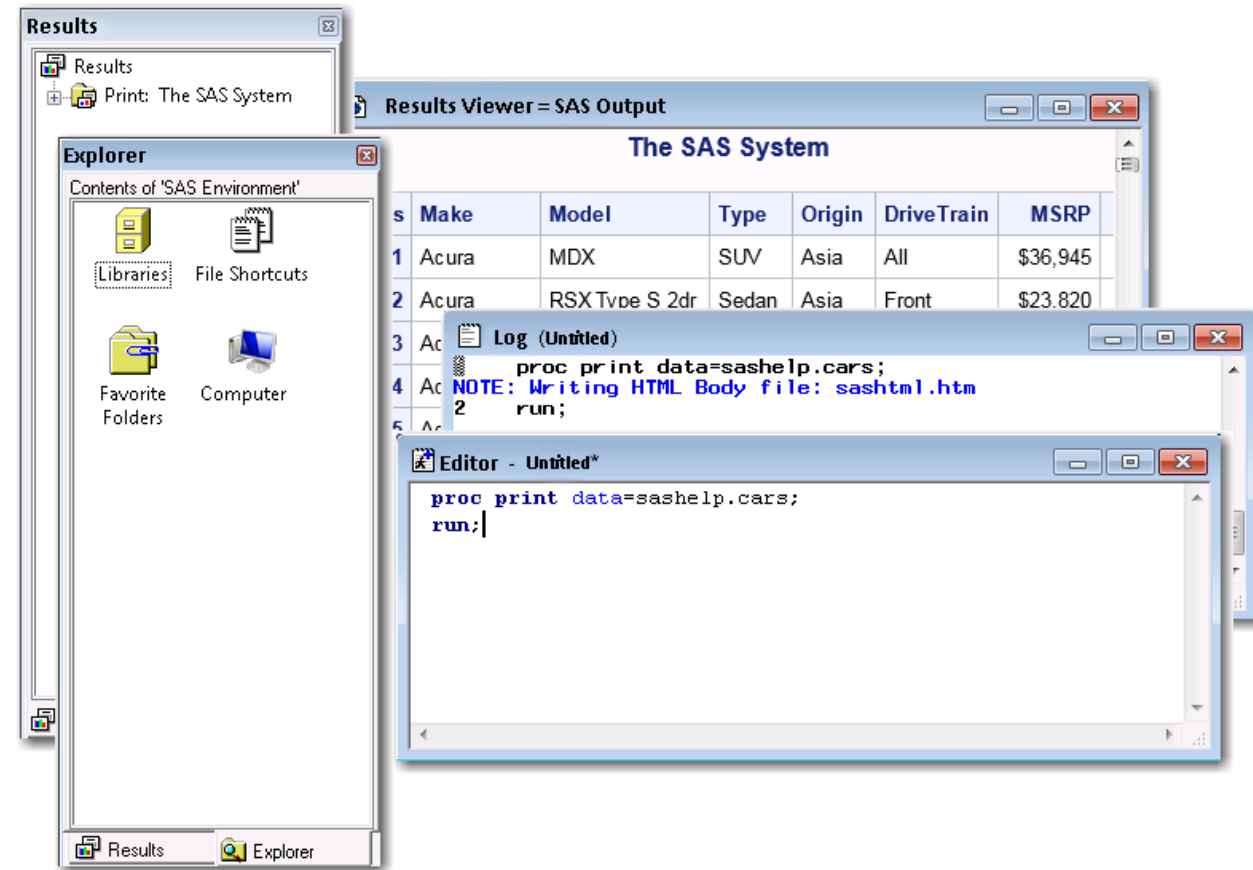
SAS Editor Window (University Edition)

Editor
- ➢ This is a text editor window to type, edit and submit SAS programs

SAS LOG Window

SAS HTML Output

SAS Output Data View

SAS Classic Output – LISTING OUTPUT

Building Blocks of SAS Programming

➢ DATA helps to build a data set. The PROC refers to a Procedure. It processes the data.

➢ DATA steps are responsible to read data, Data Management Etc, on the other hand, PROC steps are used to perform utility functions, analysis of data, or print reports.
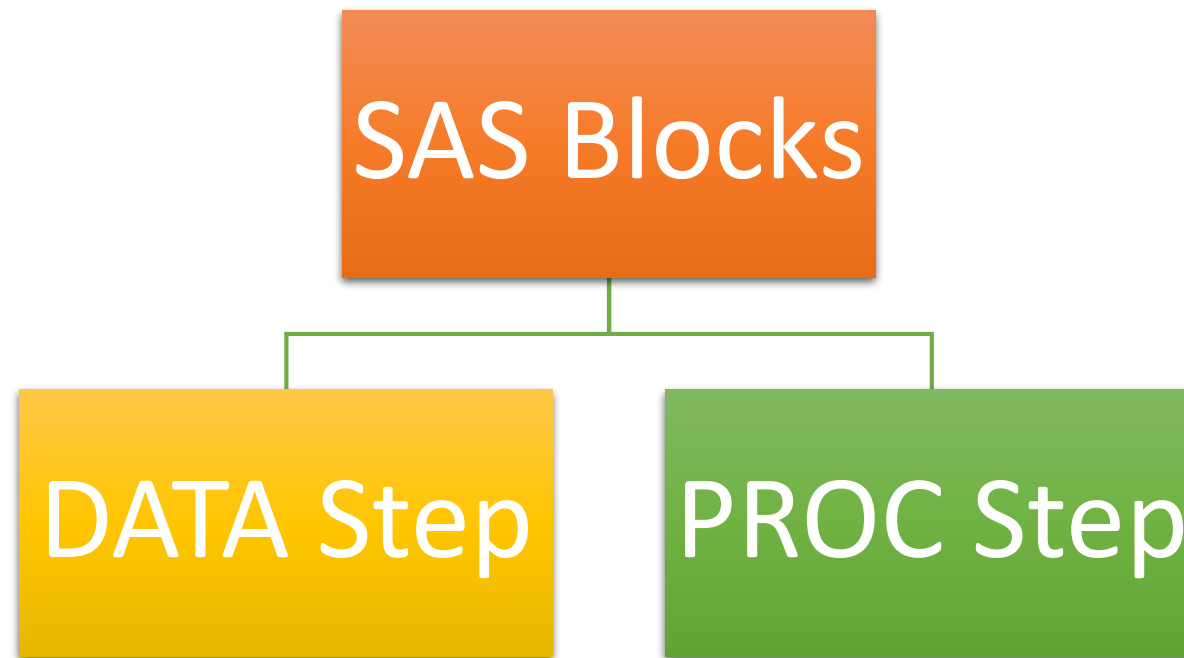
SAS Block Diagram

When you submit a DATA step for execution, it is first compiled and then executed.
1. Compilation Phase
2. Execution Phase

1. **Compilation Phase**
   - SAS checks the syntax of the SAS statements and compiles them, that is, automatically translates the statements into machine code.

During the compile phase, SAS creates the following three items:
1. input buffer
2. program data vector (PDV)
3. descriptor information

Compiler	3. Word Scanner	2. Input Stack	1. Editor
Data sample_run;	Set Sashelp . Class ; Run ;	Data sample_run; Set sashelp.class; Run;	Data sample_run; Set sashelp.class; Run;
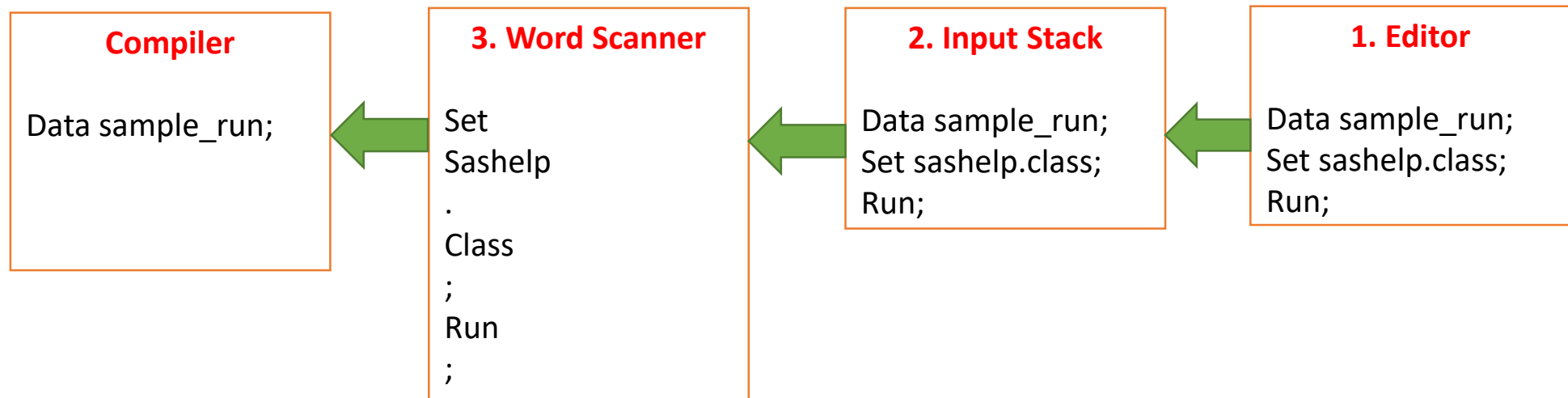
SAS Compilation Phase

Input Stack

- When you submit a program, it goes to an area of memory called the input stack.

1.reads the text in the input stack (left-to-right, top-to-bottom)

2.routes text to the appropriate compiler upon demand

3. suspends this activity when a step boundary such as a RUN statement is reached

4. executes the compiled code if there are no compilation errors

5. repeats this process for any subsequent steps.
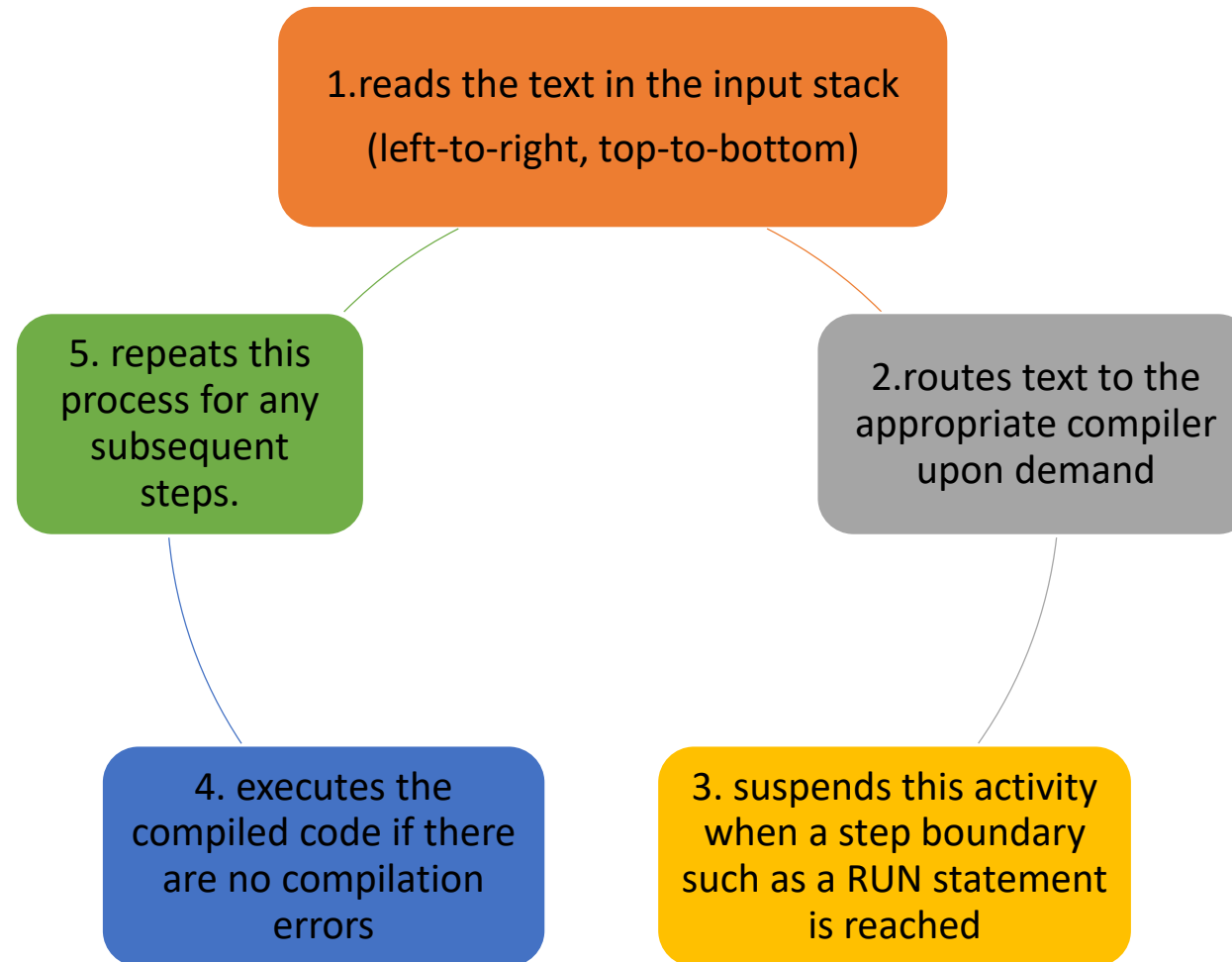
SAS Compilation Phase

- Divides program text into fundamental units called tokens.

1. Tokens are passed on demand to the compiler.

WORD SCANNER

3. The compiler performs a syntax check on the statement.

2. The compiler requests tokens until it receives a semicolon.
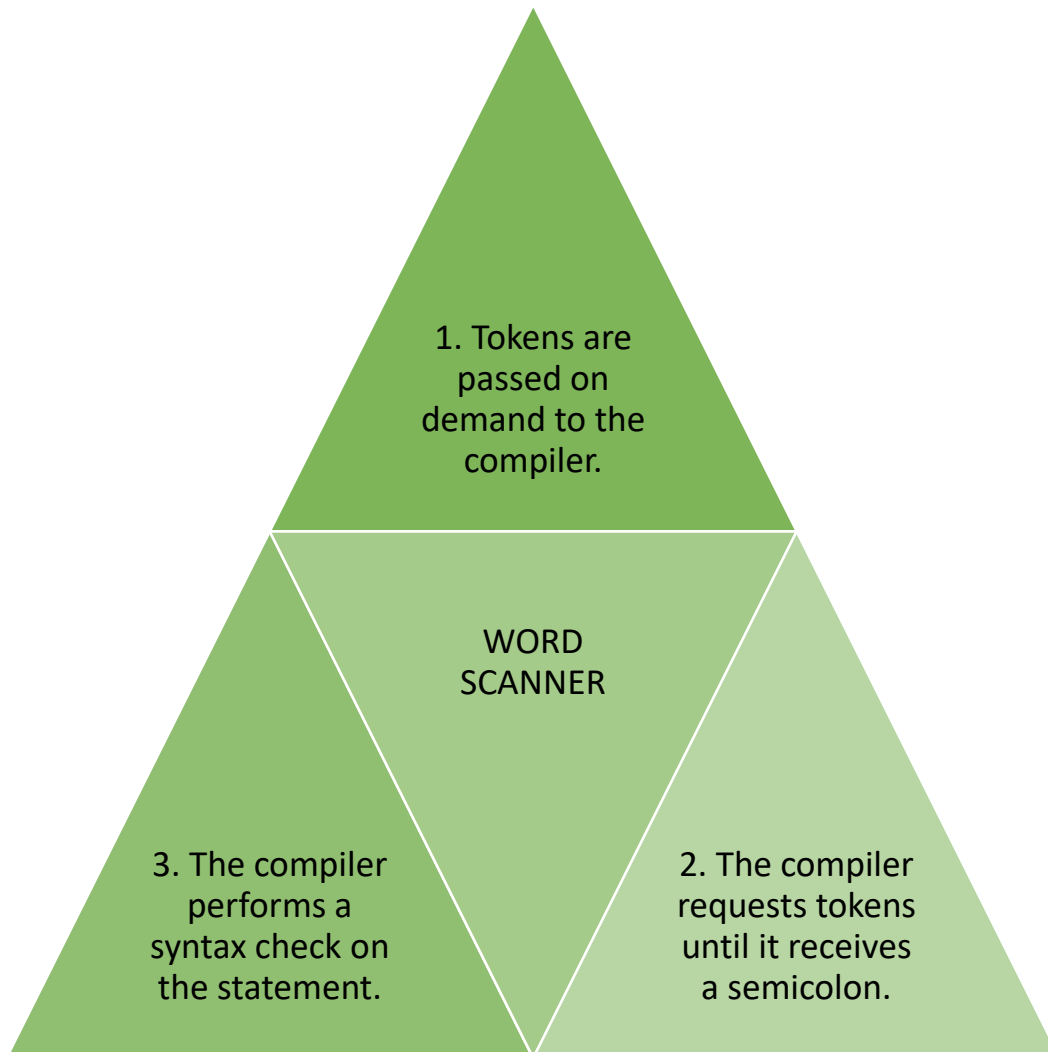
The word scanner recognizes four types of tokens:

1. **Literal token:**
   - Examples: "Any text" 'Any text'.

2. **Number token:**
   - Examples: 23 109 '01jan2002'd 5e8 42.7

3. **Name token:**
   - Examples: infile _n_ item3 univariate dollar10.2

4. **Special token:**
   - Examples: * / + - ** ; $ ( ) . & %

A token ends when the word scanner detects
- the beginning of another token
- a blank after a token.

The maximum length of any token is 32767 characters.

SAS Compilation Phase

Program Data Vector

- The program data vector is the area of memory where SAS builds a data set, one observation at a time. Like the term input buffer, the term program data vector refers to a logical concept.

- The program data vector contains two automatic variables that can be used for processing but
- which are not written to the data set as part of an observation.

    - _N_ counts the number of times that the DATA step begins to execute.

    - _ERROR_ signals the occurrence of an error that is caused by the data during execution.
        - The default value is 0, which means there is no error. When one or more errors occur, the value is set to 1.

_N_	_ERROR_	

SAS Execution Phase

Name	Gender	Salary
Adam	M	100
Eve	F	100
Jack	M	200
Flossy	F	150

NULL Handling:

If a Numeric Data is NULL → Missing, Denoted by Period (.)

If a Character Data is NULL → BLANK

_N_	_ERROR_	Name	Gender	Salary
1	0			.

Descriptor Portion

Compilation Phase	→	Name	Gender	Salary

Execution Phase	→			

Data Portion
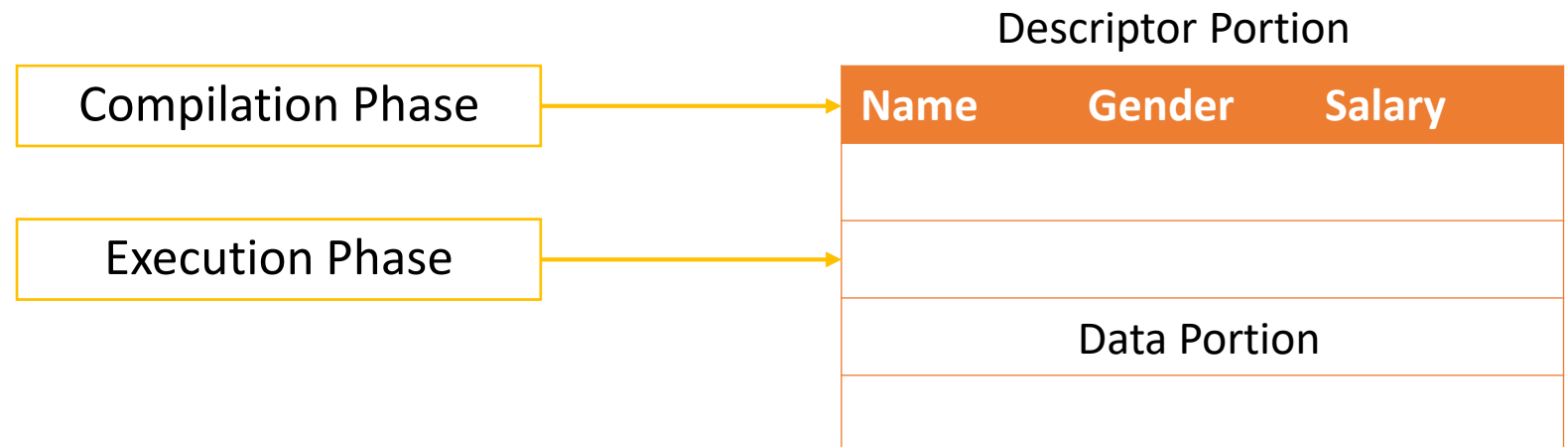
SAS Execution Phase

Step-1

_N_	_ERROR_	Name	Gender	Salary
1	0			.

Name	Gender	Salary
Adam	M	100

Step-2

_N_	_ERROR_	Name	Gender	Salary
1	0	Adam	M	100

Step-3

_N_	_ERROR_	Name	Gender	Salary
2	0			.

Name	Gender	Salary
Adam	M	100
Eve	F	100

Step-4

_N_	_ERROR_	Name	Gender	Salary
2	0	Eve	F	100

Name	Gender	Salary
Adam	M	100
Eve	F	100
Jack	M	200
Flossy	F	150

Repeats till Data step completes

BASIC RULES

SAS Statements

➢ All the SAS statements end in a semicolon. This is a basic rule that differentiates a simple English statement from SAS statement.
➢ Not Case Sensitive
➢ Statements can continue on the next line.
➢ A statement can be on the same line as other statements.

➢ **Errors in SAS Programming**

➢ Errors in SAS occur mainly due to,
  ➢ Missing semi-colon
  ➢ Misspelled words.
  ➢ Invalid variable names
  ➢ Missing or invalid punctuation
  ➢ Invalid options.

➢ We can see these errors in red color in the log window.