



Macro Quoting

A Look Behind the Scenes



2019-11-13 / Tim Lepp / v2.0





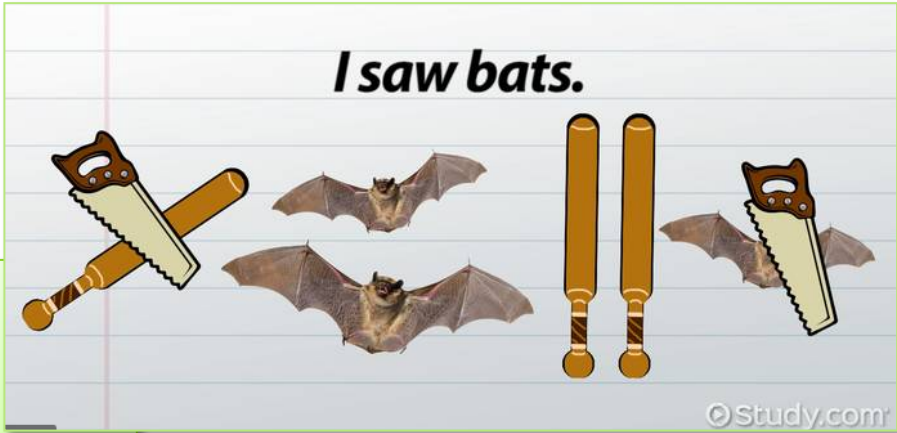
Agenda

- // Macro Quoting – Introduction
- // Macro Quoting Functions – Overview
- // Program Flow – Basics
- // Macro Quoting – Timing
- // Summary

Macro Quoting – Introduction

“I never said she stole my money” means..

- (a) someone else accused her
- (b) the thief was someone else
- (c) it wasn't money she stole, but something else



”H&M, %Percent, g=9.8” means..

- (a) &M is a macro variable, %Percent a macro call
- (b) a parameter list
- (c) fashion label string

blank	)	=	LT
;	(		GE
¬	+	AND	GT
^	—	OR	IN
~	*	NOT	%
, (comma)	/	EQ	&
'	<	NE	#

Masking with non-printable characters from 0x01 to 0x1F

UNQUOTED	& % ' " () + - * / < > = ^ ~ ; ,
NRBQUOTED	[06] [0F] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [1A] [1C] [0B] [1F] [0E] [1E] [08]



Macro Quoting Functions - Overview

□■■■■■

```
DATA _NULL_; CALL SYMPUT('var0','H&M, O'Neill, %Percent, g=9.8'); RUN;
```

```
%LET var1_STR      =      %str(H&M, O%'Neill, %Percent, g=9.8);  
%LET var2_NRSTR    =      %nrstr(H&M, O%'Neill, %Percent, g=9.8);  
%LET var3_BQUOTE   =      %bquote(&var0.);  
%LET var4_NRBQUOTE =      %nrquote(&var0.);  
%LET var5_SUPERQ   =      %superq(var0);
```

name Macro Variable Name	value Macro Variable Value
VAR0	H&M, O'Neill, %Percent, q=9.8
VAR1_STR	[01]H&M[1E] O[11]Neill[1E] %Percent[1E] g[1C]9.8[02]
VAR2_NRSTR	[01]H[0F]M[1E] O[11]Neill[1E] [10]Percent[1E] g[1C]9.8[02]
VAR3_BQUOTE	[04]H&M[1E] O[11]Neill[1E] %Percent[1E] g[1C]9.8[08]
VAR4_NRBQUOTE	[06]H[0F]M[1E] O[11]Neill[1E] [10]Percent[1E] g[1C]9.8[08]
VAR5_SUPERQ	[06]H[0F]M[1E] O[11]Neill[1E] [10]Percent[1E] g[1C]9.8[08]

%STR

- // Masks groups **A** and group **C** only with preceding %
- // Difficulties with unmatched Quotation marks and brackets
- // Compilation phase function
- // Works before resolution of macro triggers

Groups of Special Characters

A + - * / < > = ~ ^ | ~ ; , # blank AND OR NOT EQ NE LE LT GE GT IN

B & %

C ' " ()



Macro Quoting Functions - Overview

■■■■■■■

```
DATA _NULL_; CALL SYMPUT('var0','H&M, O'Neill, %Percent, g=9.8'); RUN;
```

```
%LET var1 STR = %str(H&M, O'Neill, %Percent, g=9.8);
```

```
%LET var2_NRSTR = %nrstr(H&M, O'Neill, %Percent, g=9.8);
```

```
%LET var3_BQUOTE = %bquote(&var0.);
```

```
%LET var4_NRBQUOTE = %nrquote(&var0.);
```

```
%LET var5_SUPERQ = %superq(var0);
```

name Macro Variable Name	value Macro Variable Value
VAR0	H&M, O'Neill, %Percent, g=9.8
VAR1 STR	[01]H&M[1E] O[11]Neill[1E] %Percent[1E] g[1C]9.8[02]
VAR2_NRSTR	[01]H[0F]M[1E] O[11]Neill[1E] [10]Percent[1E] g[1C]9.8[02]
VAR3_BQUOTE	[04]H&M[1E] O[11]Neill[1E] %Percent[1E] g[1C]9.8[08]
VAR4_NRBQUOTE	[06]H[0F]M[1E] O[11]Neill[1E] [10]Percent[1E] g[1C]9.8[08]
VAR5_SUPERQ	[06]H[0F]M[1E] O[11]Neill[1E] [10]Percent[1E] g[1C]9.8[08]

%NRSTR

// Masks groups **A**, **B** and **C** only with preceding %

// Difficulties with unmatched Quotation marks and brackets

// Compilation phase function

// Works before resolution of macro triggers

Groups of Special Characters

A + - * / < > = ~ ^ | ~ ; , # blank AND OR NOT EQ NE LE LT GE GT IN

B & %

C ' " ()



Macro Quoting Functions - Overview

■■■■■

```
DATA _NULL_; CALL SYMPUT('var0','H&M, O'Neill, %Percent, g=9.8'); RUN;
```

```
%LET var1_STR      =      %str(H&M, 0%'Neill, %Percent, g=9.8);  
%LET var2_NRSTR    =      %nrstr(H&M, 0%'Neill, %Percent, g=9.8);  
%LET var3_BQUOTE   =      %bquote(&var0.);  
%LET var4_NRBQUOTE =      %nrbquote(&var0.);  
%LET var5_SUPERQ   =      %superq(var0);
```

name Macro Variable Name	value Macro Variable Value
VAR0	H&M, O'Neill, %Percent, g=9.8
VAR1_STR	[01]H&M[1E] O[11]Neill[1E] %Percent[1E] g[1C]9.8[02]
VAR2_NRSTR	[01]H[0F]M[1E] O[11]Neill[1E] [10]Percent[1E] g[1C]9.8[02]
VAR3_BQUOTE	[04]H&M[1E] O[11]Neill[1E] %Percent[1E] g[1C]9.8[08]
VAR4_NRBQUOTE	[06]H[0F]M[1E] O[11]Neill[1E] [10]Percent[1E] g[1C]9.8[08]
VAR5_SUPERQ	[06]H[0F]M[1E] O[11]Neill[1E] [10]Percent[1E] g[1C]9.8[08]

%BQUOTE

- // Masks groups **A** und **C**
- // Also unmatched Quotation marks and brackets can be processed
- // Execution phase function
- // Works after resolution of macro triggers

Groups of Special Characters

A + - * / < > = ~ ^ | ~ ; , # blank AND OR NOT EQ NE LE LT GE GT IN

B & %

C ' " ()



Macro Quoting Functions - Overview

■■■■■

```
%LET M=ennes&Mauritz;  
%MACRO Percent(); _xxxxxxx %MEND;
```

```
DATA _NULL_; CALL SYMPUT('var0','H&M, O'Neill, %Percent, g=9.8'); RUN;
```

```
%LET var1_STR      =      %str(H&M, 0%'Neill, %Percent, g=9.8);  
%LET var2_NRSTR    =      %nrstr(H&M, 0%'Neill, %Percent, g=9.8);  
%LET var3_BQUOTE   =      %bquote(&var0.);  
%LET var4_NRBQUOTE = %nrbquote(&var0.);  
%LET var5_SUPERQ   =      %superq(var0);
```

name Macro Variable Name	value Macro Variable Value
VAR0	H&M, O'Neill, %Percent, g=9.8
VAR1_STR	[01]H&M[1E] O[11]Neill[1E] %Percent[1E] g[1C]9.8[02]
VAR2_NRSTR	[01]H[0F]M[1E] O[11]Neill[1E] [10]Percent[1E] g[1C]9.8[02]
VAR3_BQUOTE	[04]H&M[1E] O[11]Neill[1E] %Percent[1E] q[1C]9.8[08]
VAR4_NRBQUOTE	[06]H[0F]M[1E] O[11]Neill[1E] [10]Percent[1E] g[1C]9.8[08]
VAR5_SUPERQ	[06]H[0F]M[1E] O[11]Neill[1E] [10]Percent[1E] g[1C]9.8[08]

%NRBQUOTE

- // Masks groups **A**, **B** and **C**
- // Also unmatched Quotation marks and brackets can be processed
- // NR misleading, as one could think it works analogously to %NRSTR
- // Execution phase function
- // Works after resolution of macro triggers

Groups of Special Characters

A + - * / < > = ~ ^ | ~ ; , # blank AND OR NOT EQ NE LE LT GE GT IN

B & %

C ' " ()



Macro Quoting Functions - Overview

■■■■■

```
%LET M=ennes&Mauritz;  
%MACRO Percent(); _Xxxxxxx %MEND;
```

```
DATA _NULL_; CALL SYMPUT('var0','H&M, O'Neill, %Percent, g=9.8'); RUN;
```

```
%LET var1_STR      =      %str(H&M, 0%'Neill, %Percent, g=9.8);  
%LET var2_NRSTR    =      %nrstr(H&M, 0%'Neill, %Percent, g=9.8);  
%LET var3_BQUOTE   =      %bquote(&var0.);  
%LET var4_NRBQUOTE = %nrbquote(&var0.);  
%LET var5_SUPERQ   =      %superq(var0);
```

name Macro Variable Name	value Macro Variable Value
VAR0	H&M, O'Neill, %Percent, g=9.8
VAR1_STR	[01]Hennes&Mauritz[1E] O[11]Neill[1E] _Xxxxxxx[1E] g[1C]9.8[02]
VAR2_NRSTR	[01]H[0F]M[1E] O[11]Neill[1E] [10]Percent[1E] g[1C]9.8[02]
VAR3_BQUOTE	[04]Hennes&Mauritz[1E] O[11]Neill[1E] Xxxxxxx[1E] q[1C]9.8[08]
VAR4_NRBQUOTE	[06]Hennes[0F]Mauritz[1E] O[11]Neill[1E] _Xxxxxxx[1E] g[1C]9.8[08]
VAR5_SUPERQ	[06]H[0F]M[1E] O[11]Neill[1E] [10]Percent[1E] g[1C]9.8[08]

%NRBQUOTE

- // Masks groups **A**, **B** and **C**
- // Also unmatched Quotation marks and brackets can be processed
- // NR misleading, as one could think it works analogously to %NRSTR
- // Execution phase function
- // Works after resolution of macro triggers

Groups of Special Characters

A + - * / < > = ~ ^ | ~ ; , # blank AND OR NOT EQ NE LE LT GE GT IN

B & %

C ' " ()



Macro Quoting Functions - Overview

■■■■■□

```
%LET M=ennes&Mauritz;  
%MACRO Percent(); _Xxxxxxx %MEND;
```

```
DATA _NULL_; CALL SYMPUT('var0','H&M, O'Neill, %Percent, g=9.8'); RUN;
```

```
%LET var1_STR      =      %str(H&M, O%'Neill, %Percent, g=9.8);  
%LET var2_NRSTR    =      %nrstr(H&M, O%'Neill, %Percent, g=9.8);  
%LET var3_BQUOTE   =      %bquote(&var0.);  
%LET var4_NRBOUOTE =      %nrquote(&var0.);  
%LET var5_SUPERQ   =      %superq(var0);
```

name Macro Variable Name	value Macro Variable Value
VAR0	H&M, O'Neill, %Percent, g=9.8
VAR1_STR	[01]Hennes&Mauritz[1E] O[11]Neill[1E] _Xxxxxxx[1E] g[1C]9.8[02]
VAR2_NRSTR	[01]H[0F]M[1E] O[11]Neill[1E] [10]Percent[1E] g[1C]9.8[02]
VAR3_BQUOTE	[04]Hennes&Mauritz[1E] O[11]Neill[1E] _Xxxxxxx[1E] g[1C]9.8[08]
VAR4_NRBOUOTE	[06]Hennes[0F]Mauritz[1E] O[11]Neill[1E] _Xxxxxxx[1E] g[1C]9.8[08]
VAR5_SUPERQ	[06]H[0F]M[1E] O[11]Neill[1E] [10]Percent[1E] g[1C]9.8[08]

%SUPERQ

- // Masks groups **A**, **B** and **C**
- // The argument is expected to be the name of a Macrovariable (without &) whose value is being masked without further resolution
- // Execution phase function
- // Works after resolution of macro triggers

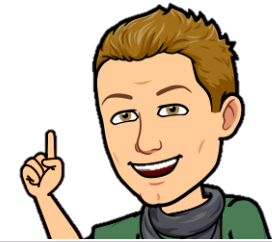
Groups of Special Characters

A + - * / < > = ~ ^ | ~ ; , # blank AND OR NOT EQ NE LE LT GE GT IN

B & %

C ' " ()

Macro Quoting Functions - Overview



Function	Masks	Runtime	Features
%STR	A, C*	Compilation	C will only be masked with preceding %
%NRSTR	A, B, C*	Compilation	C will only be masked with preceding %
%BQUOTE	A, C	Execution	
%NRBQUOTE	A, B, C	Execution	Macro trigger B are resolved before masking
%SUPERQ	A, B, C	Execution	Expects macro variable name as argument
%QUOTE	A, C*	Execution	C will only be masked with preceding %
%NRQUOTE	A, B, C*	Execution	C will only be masked with preceding %
%UNQUOTE	A*, B*, C*	Execution	Removes all masking
%QSCAN, %QSUBSTR, %QUPCASE, %QSYSEFUNC, %QCMPPRES, %QLOWCASE, %QLEFT, %QTRIM	A, B, C	Execution	Same quoting behaviour as %NRBQUOTE

Groups of Special Characters

A + - * / < > = ~ ^ | ~ ; , # blank AND OR NOT EQ NE LE LT GE GT IN

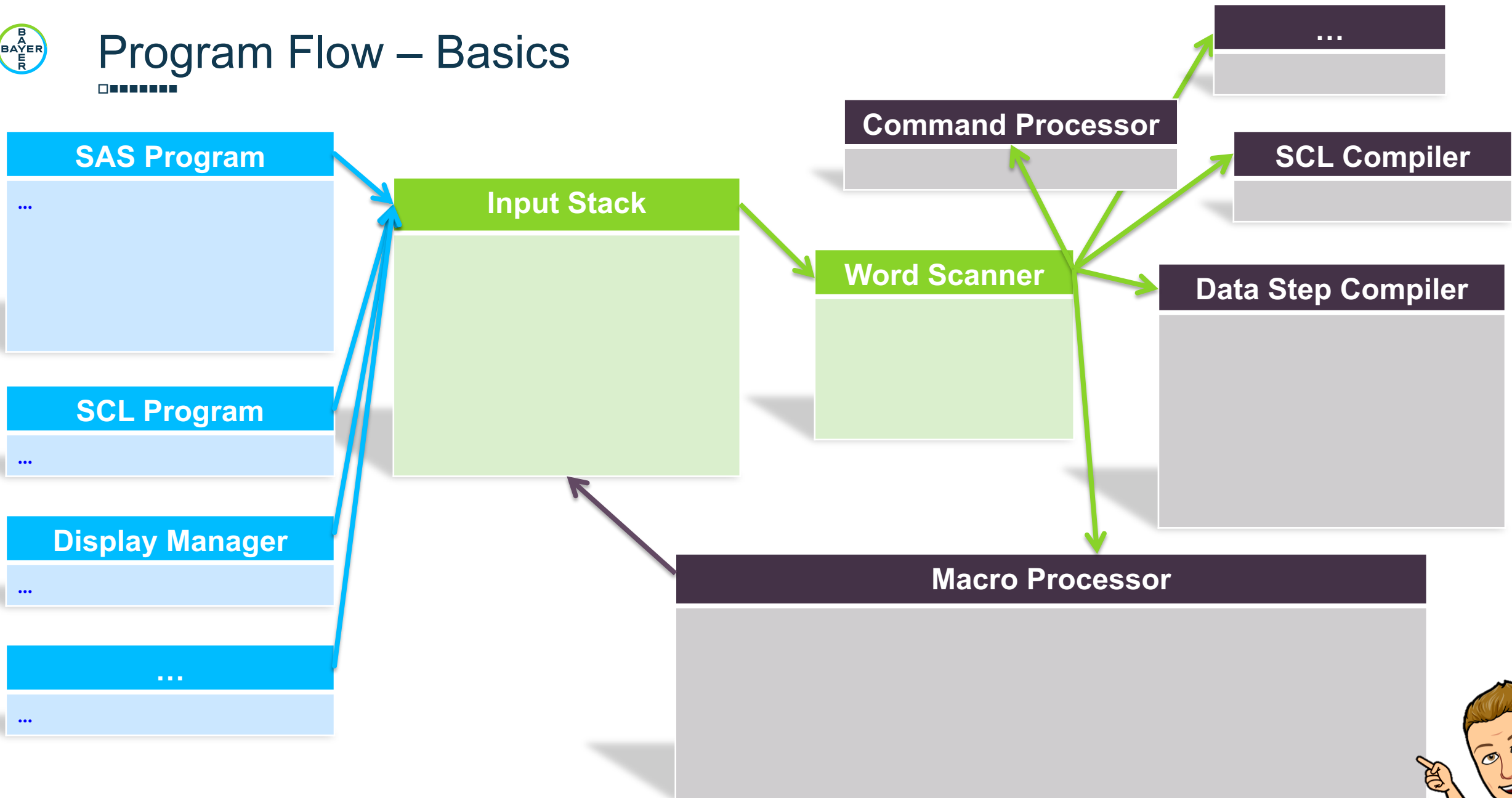
B & %

C ' " ()



Program Flow – Basics

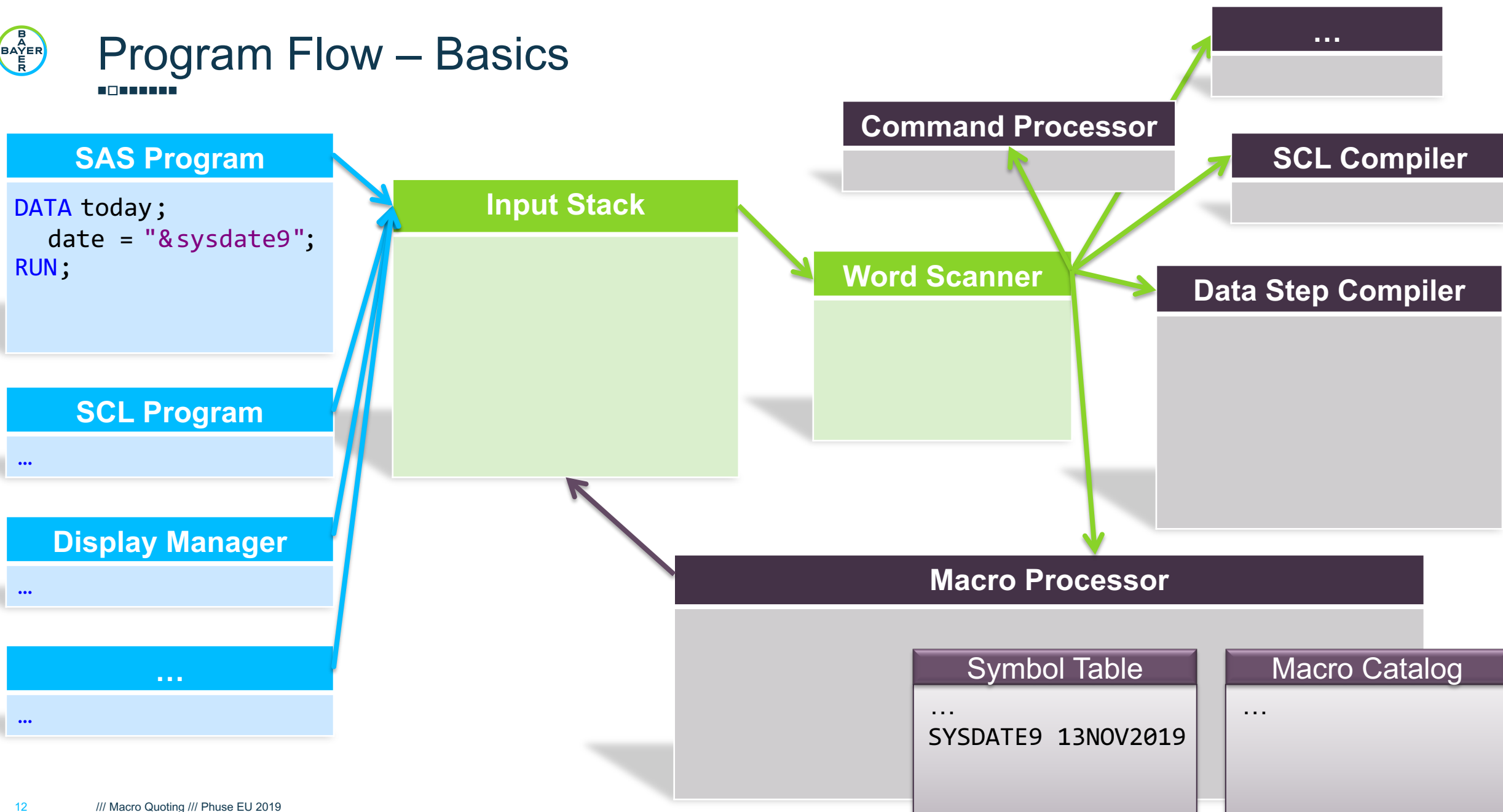
□■■■■■■■





Program Flow – Basics

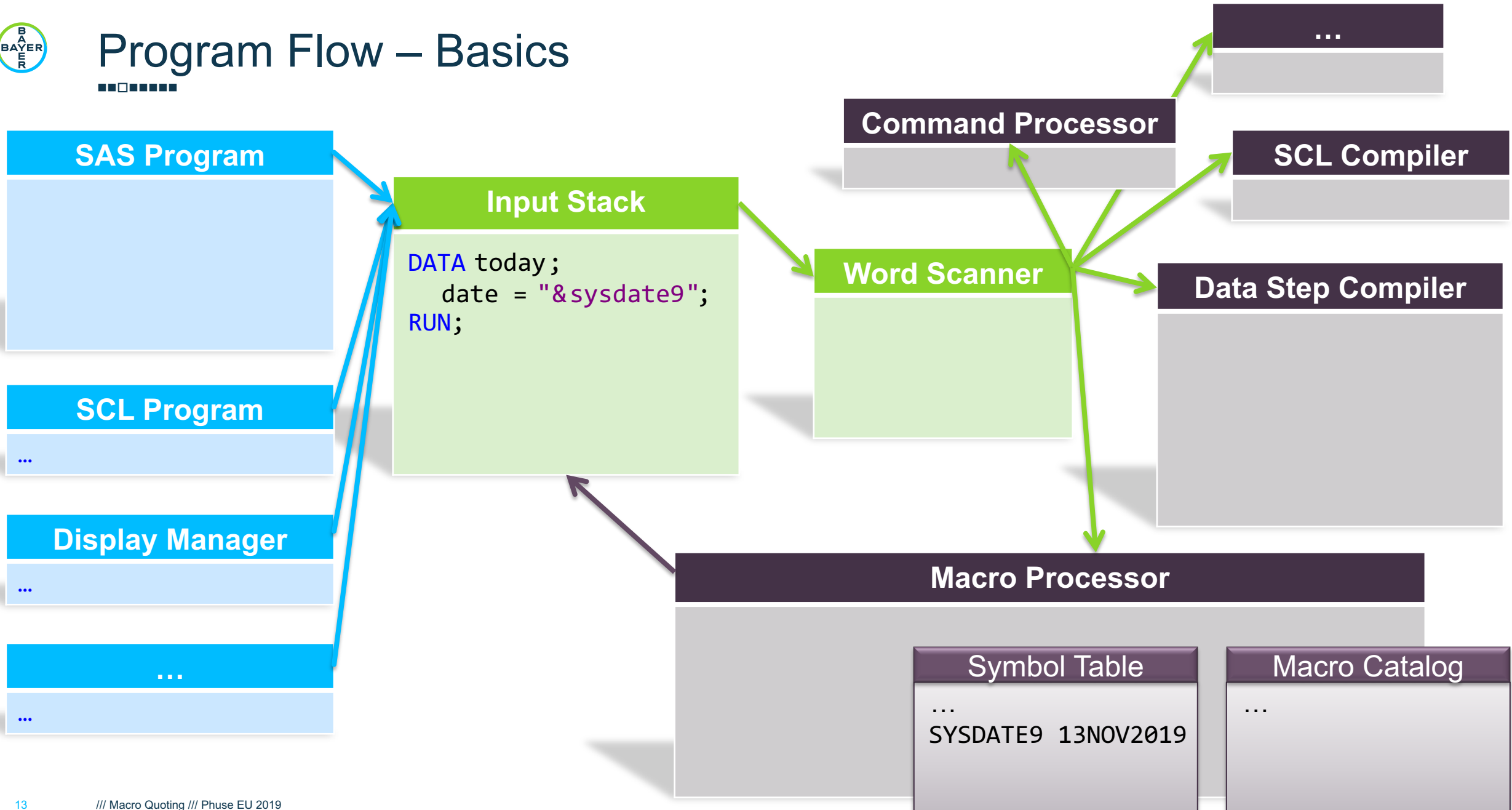
■■■■■■■■





Program Flow – Basics

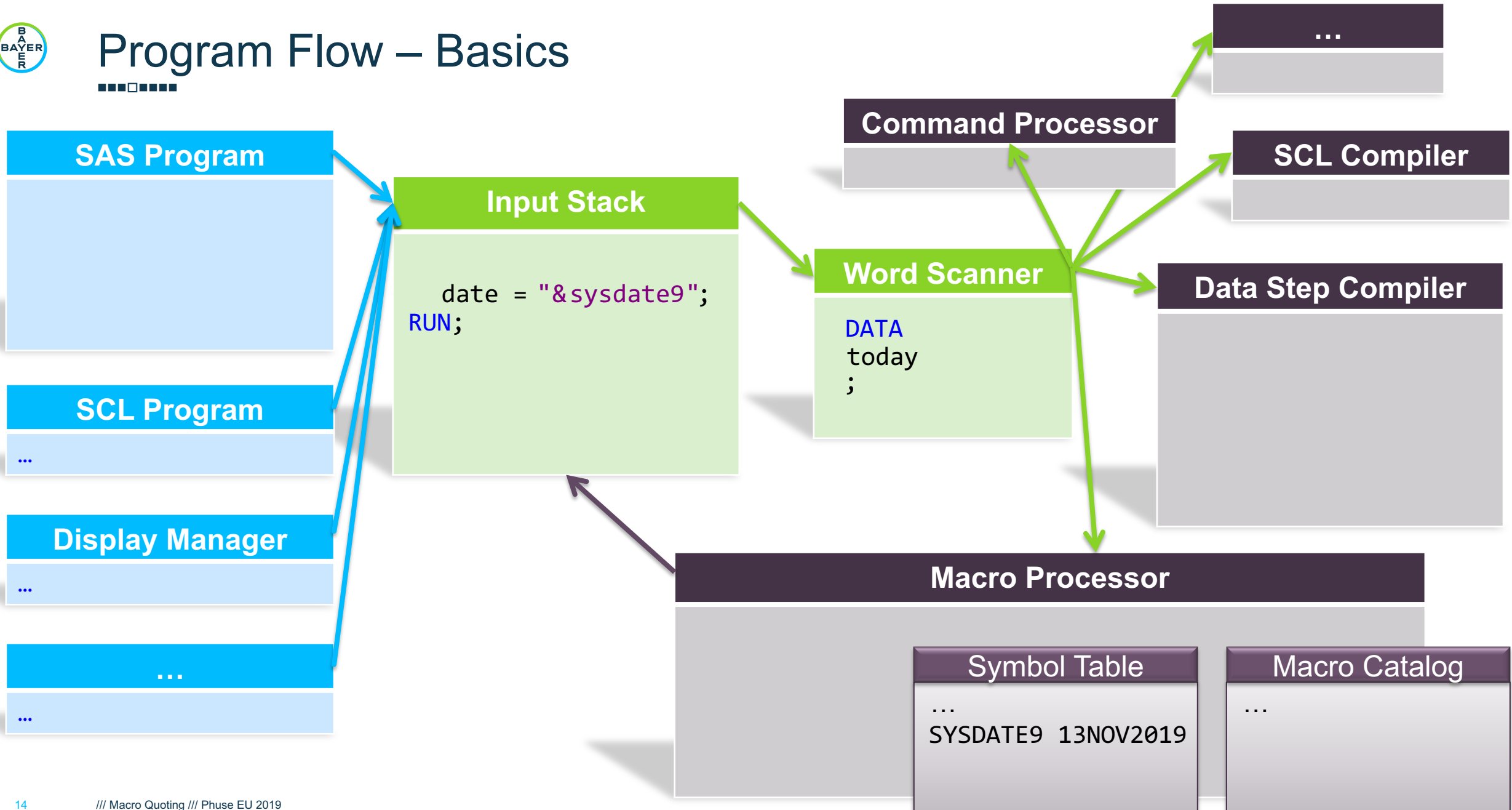
■■■■■■■





Program Flow – Basics

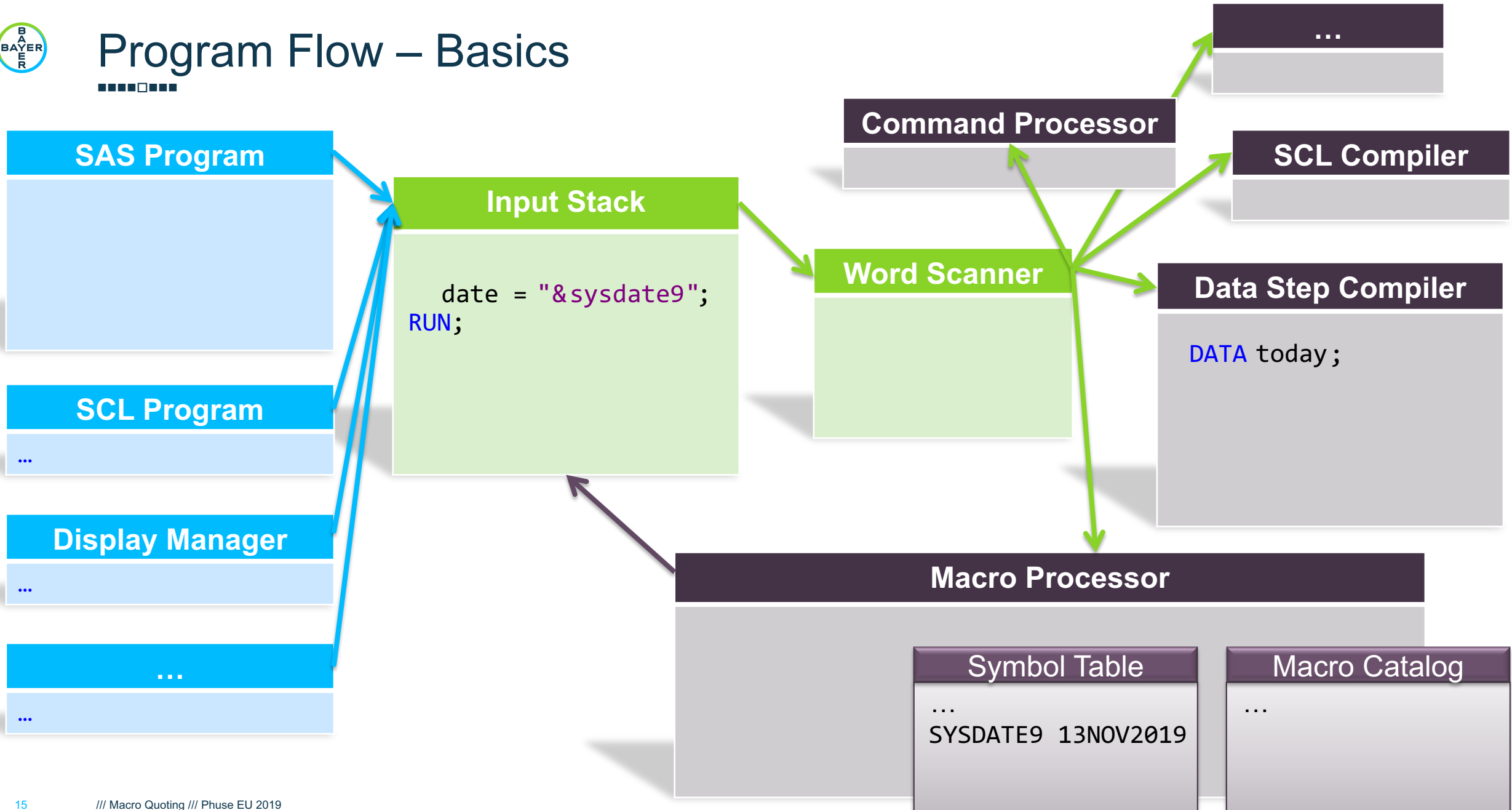
■■■■■





Program Flow – Basics

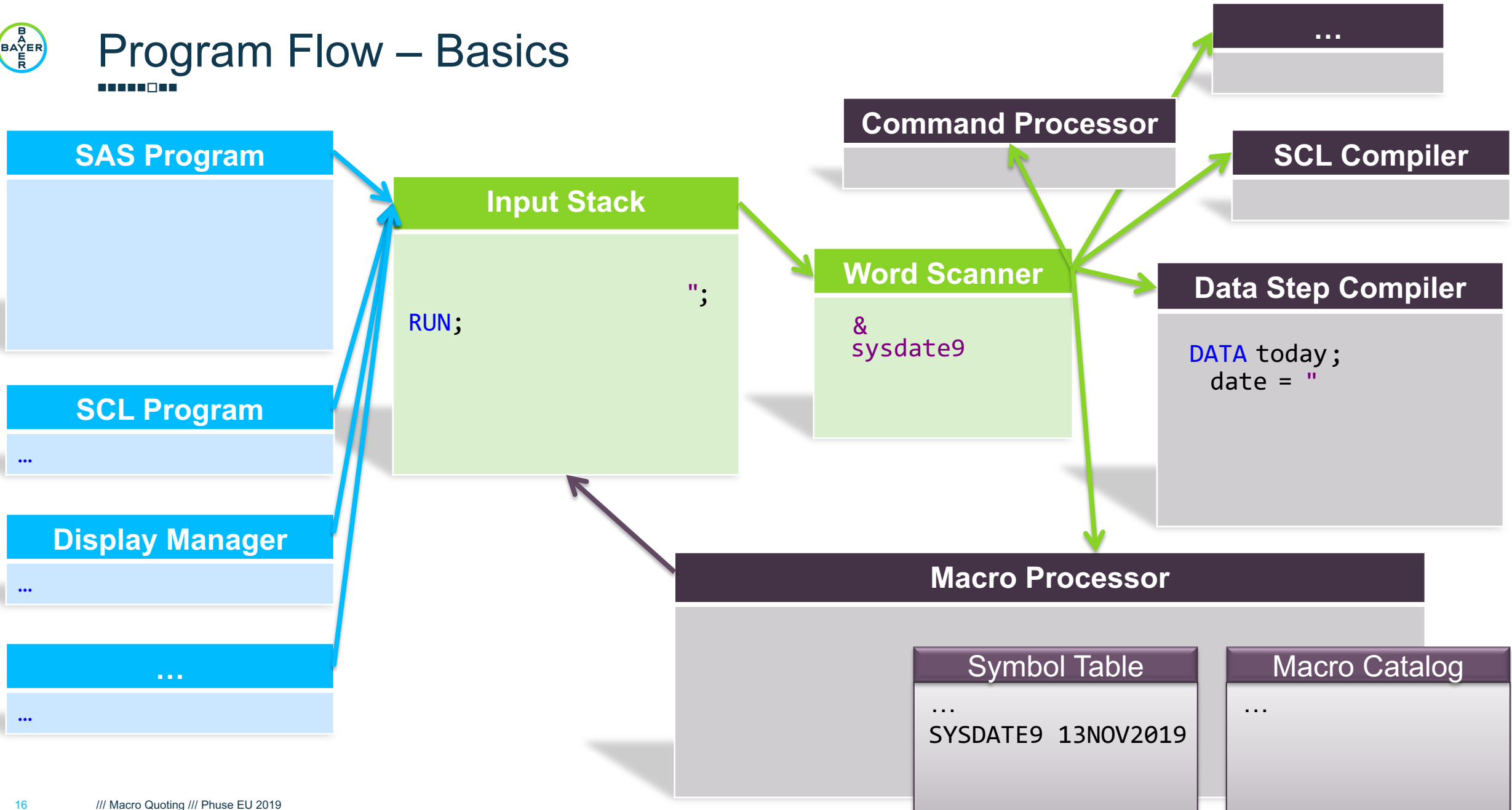
■■■■■





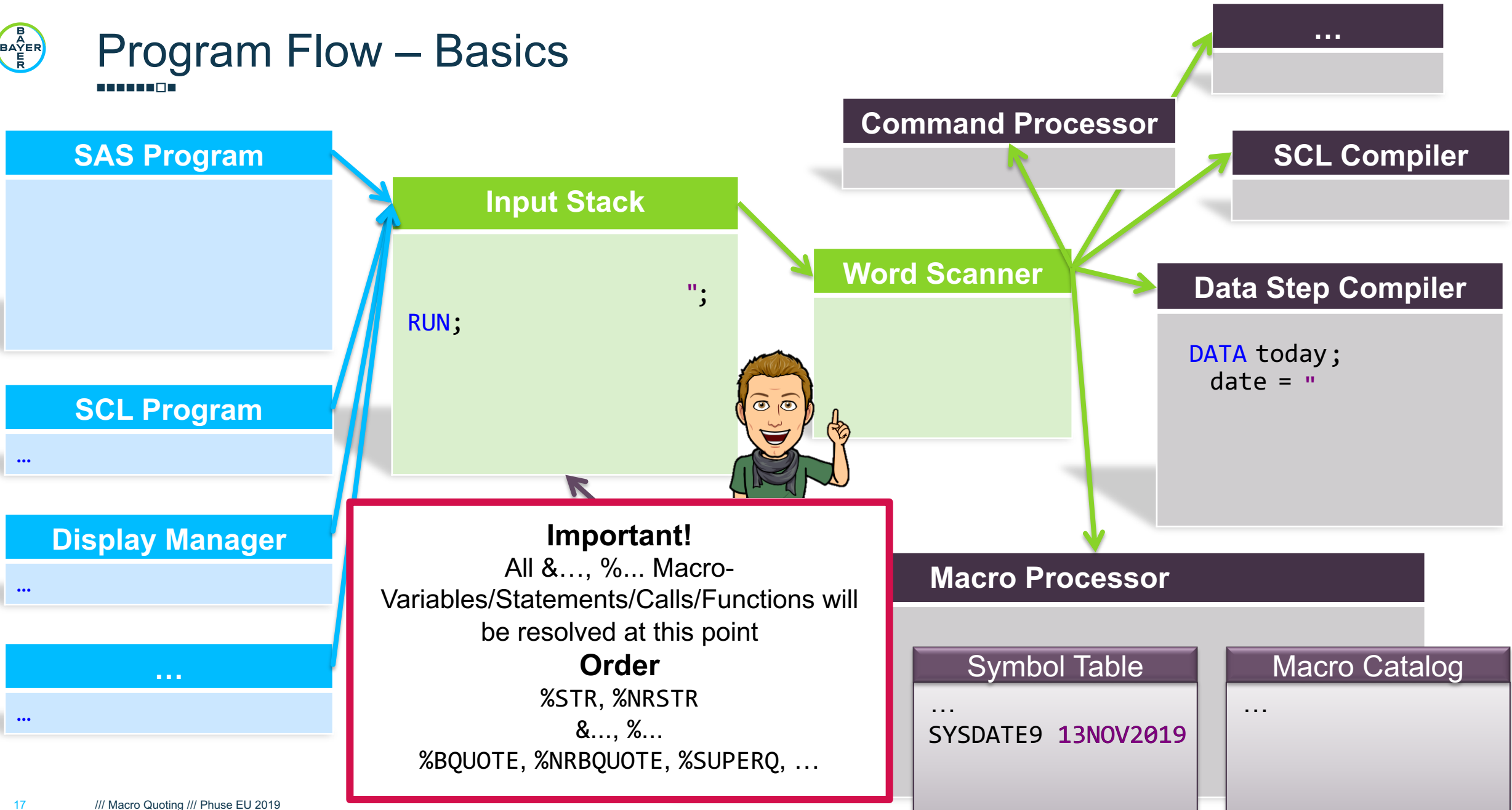
Program Flow – Basics

■■■■■□■■



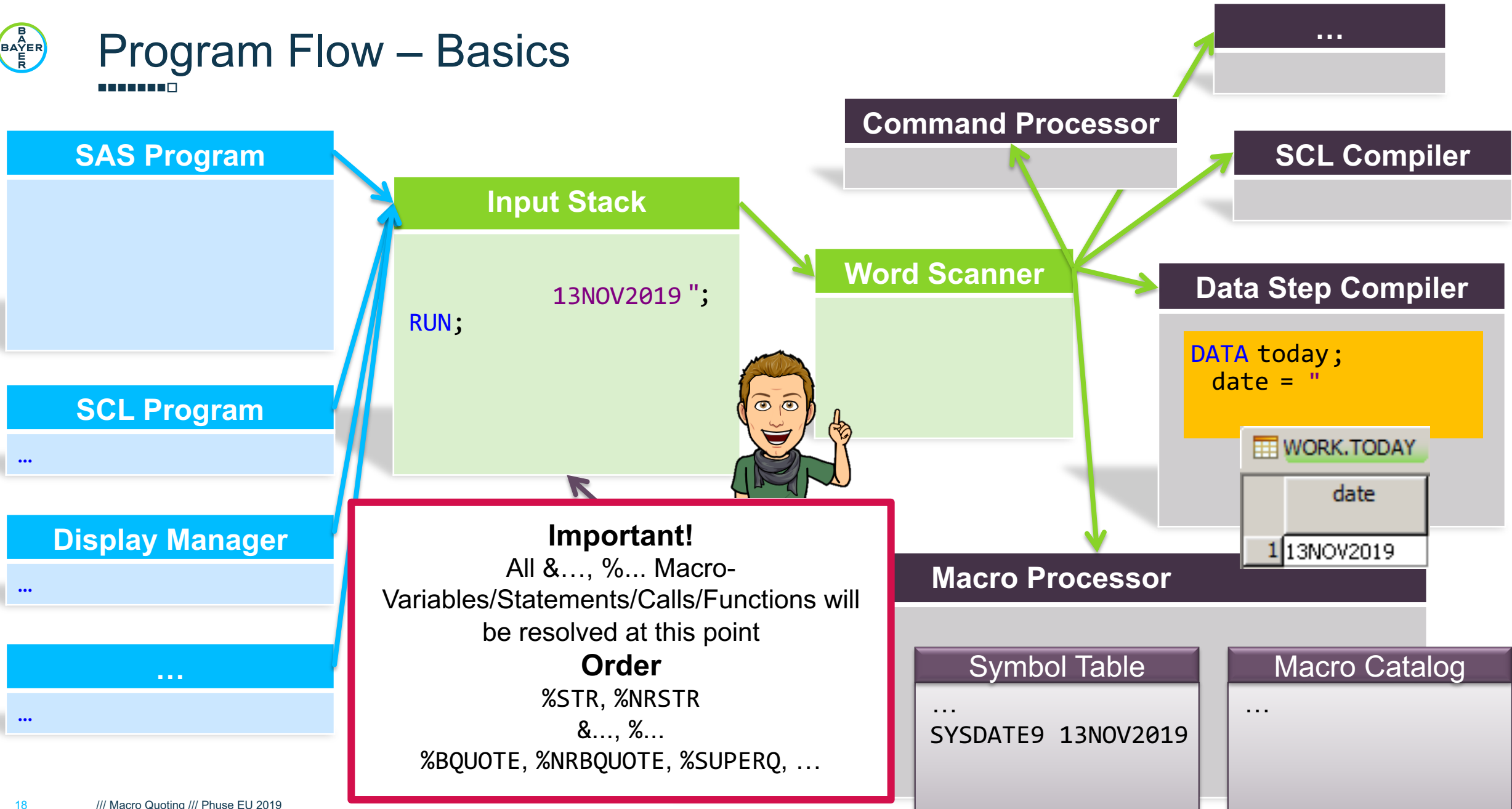
Program Flow – Basics

■■■■■■■



Program Flow – Basics

■■■■■■■□



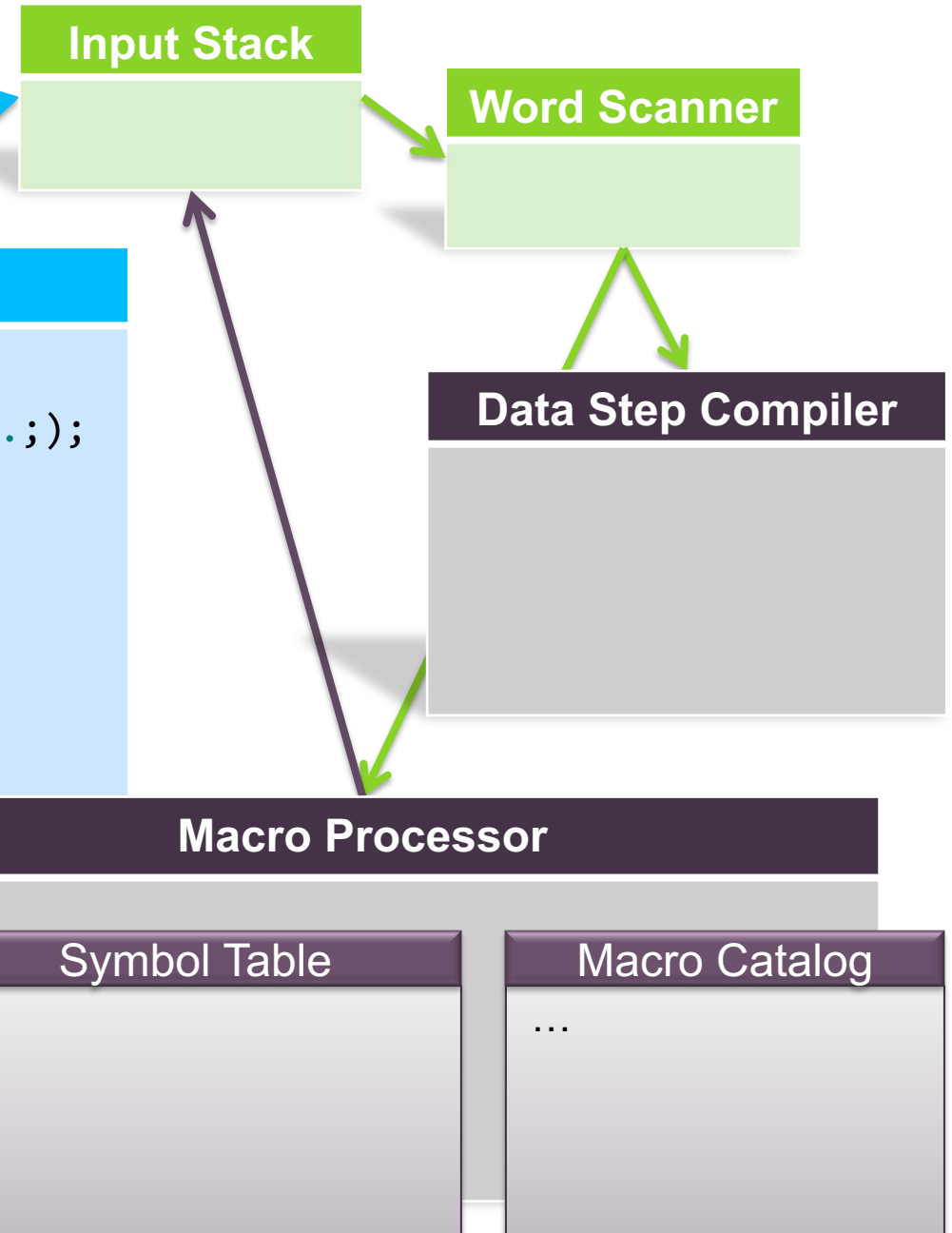


Macro Quoting - Timing

□■■■■■■■

SAS Program

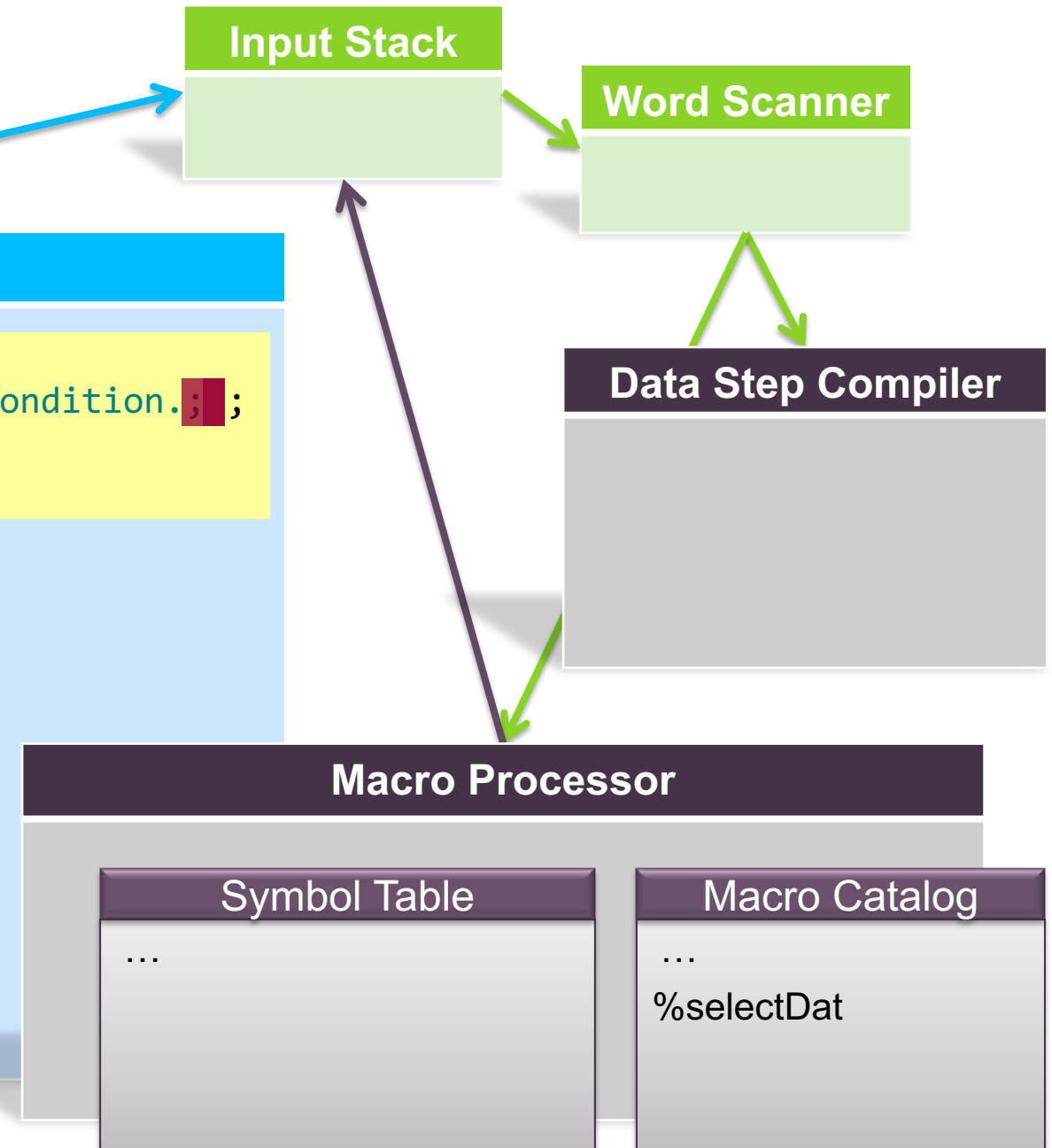
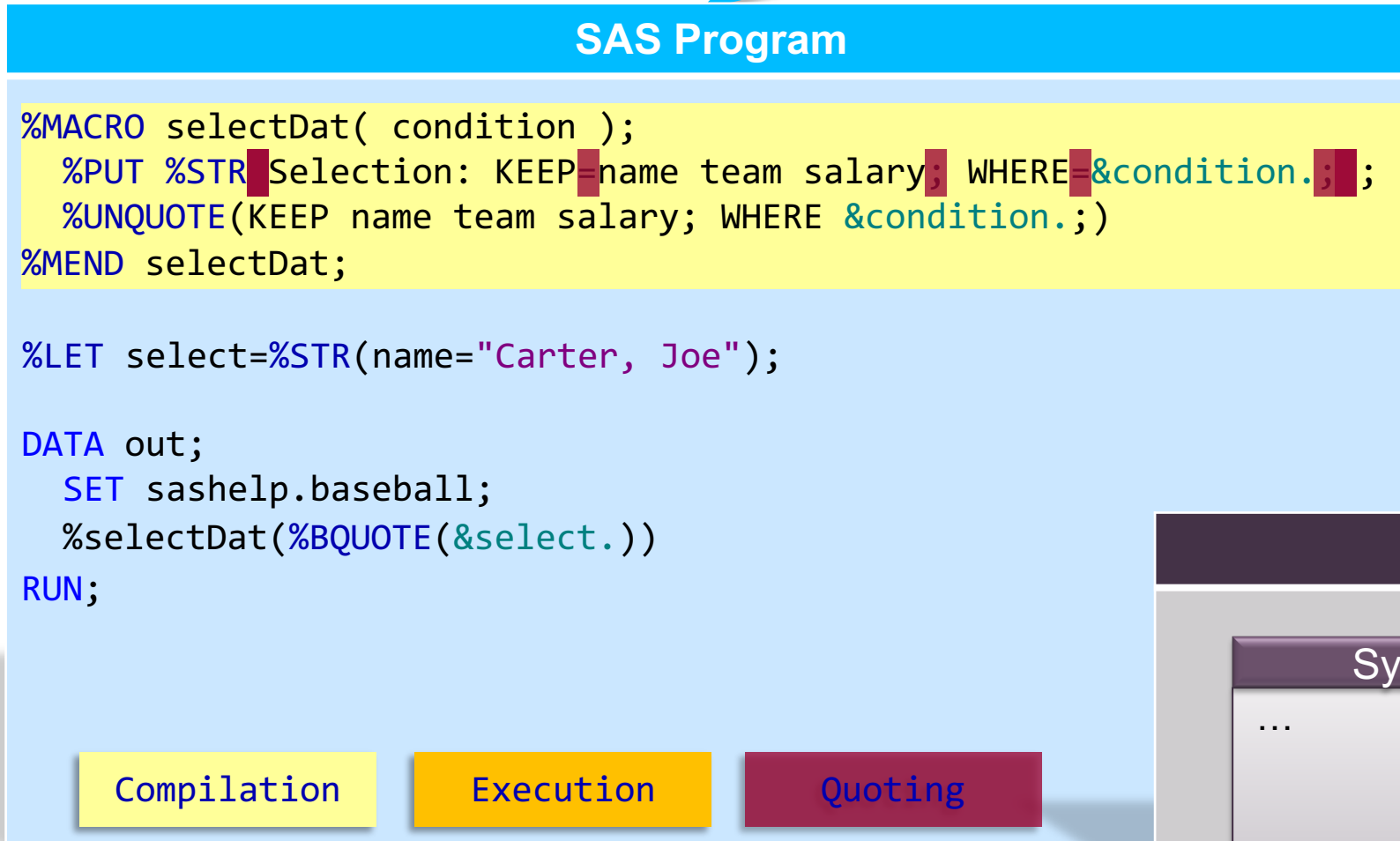
```
%MACRO selectDat( condition );  
  %PUT %STR(Selection: KEEP=name team salary; WHERE=&condition.);  
  %UNQUOTE(KEEP name team salary; WHERE &condition.);  
%MEND selectDat;  
  
%LET select=%STR(name="Carter, Joe");  
  
DATA out;  
  SET sashelp.baseball;  
  %selectDat(%BQUOTE(&select.))  
RUN;
```





Macro Quoting - Timing

■■■■■■■■





Macro Quoting - Timing

■■■■■■■

SAS Program

```
%MACRO selectDat( condition );  
  %PUT %STR Selection: KEEP name team salary; WHERE &condition.;  
  %UNQUOTE(KEEP name team salary; WHERE &condition.);  
%MEND selectDat;  
  
%LET select=%STR(name="Carter, Joe");  
  
DATA out;  
  SET sashelp.baseball;  
  %selectDat(%BQUOTE(&select.))  
RUN;
```

Compilation

Execution

Quoting

Input Stack

Word Scanner

Data Step Compiler

Macro Processor

Symbol Table

Macro Catalog

...

...

%selectDat



Macro Quoting - Timing

■■■■■■■

SAS Program

```
%MACRO selectDat( condition );  
  %PUT %STR Selection: KEEP name team salary; WHERE &condition.;  
  %UNQUOTE(KEEP name team salary; WHERE &condition.);  
%MEND selectDat;  
  
%LET select=%STR name="Carter, Joe";  
  
DATA out;  
  SET sashelp.baseball;  
  %selectDat(%BQUOTE(&select.))  
RUN;
```

Compilation

Execution

Quoting

Input Stack

Word Scanner

Data Step Compiler

Macro Processor

Symbol Table

Macro Catalog

...

...

%selectDat



Macro Quoting - Timing

■■■■■■■

SAS Program

```
%MACRO selectDat( condition );  
  %PUT %STR Selection: KEEP name team salary; WHERE &condition.;  
  %UNQUOTE(KEEP name team salary; WHERE &condition.);  
%MEND selectDat;  
  
%LET select=%STR name="Carter, Joe";  
  
DATA out;  
  SET sashelp.baseball;  
  %selectDat(%BQUOTE(&select.))  
RUN;
```

Compilation

Execution

Quoting

Input Stack

Word Scanner

Data Step Compiler

Macro Processor

Symbol Table

...
[G] select =

name "Carter, Joe"

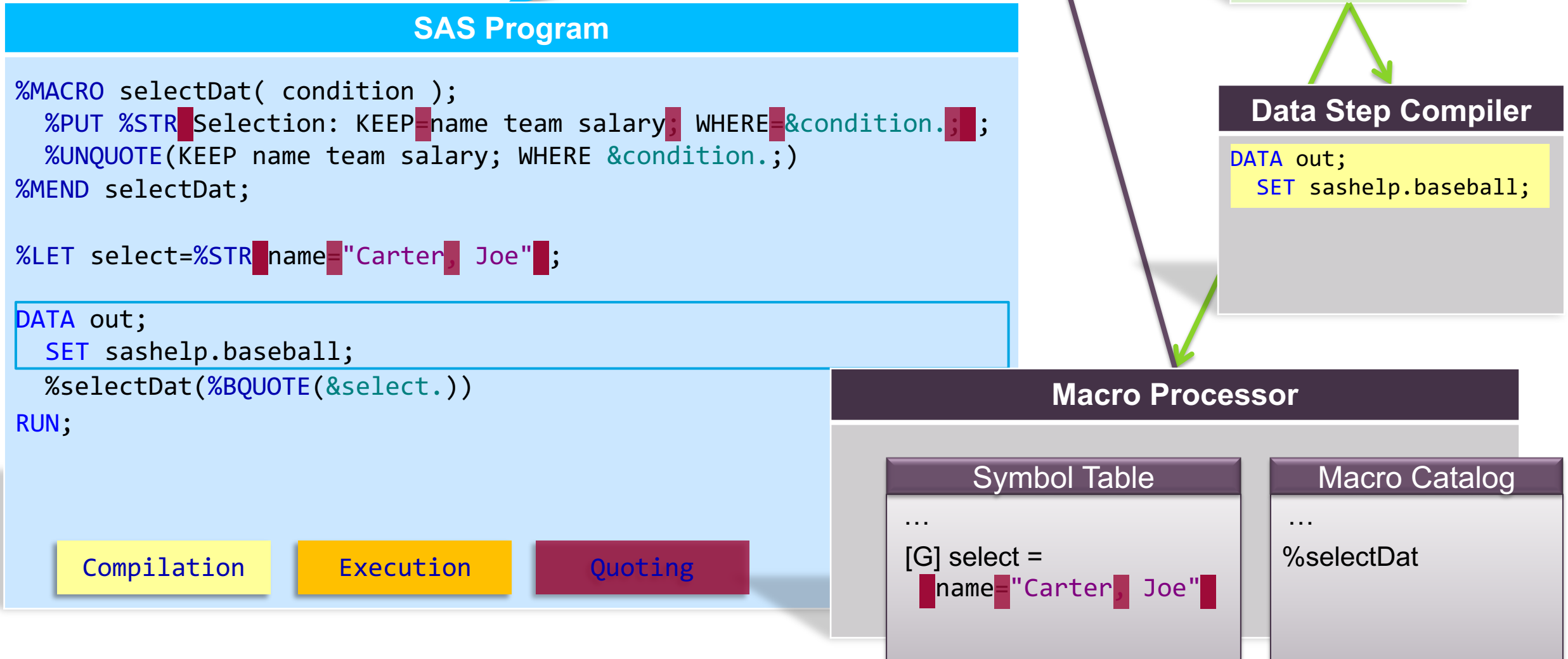
Macro Catalog

...
%selectDat



Macro Quoting - Timing

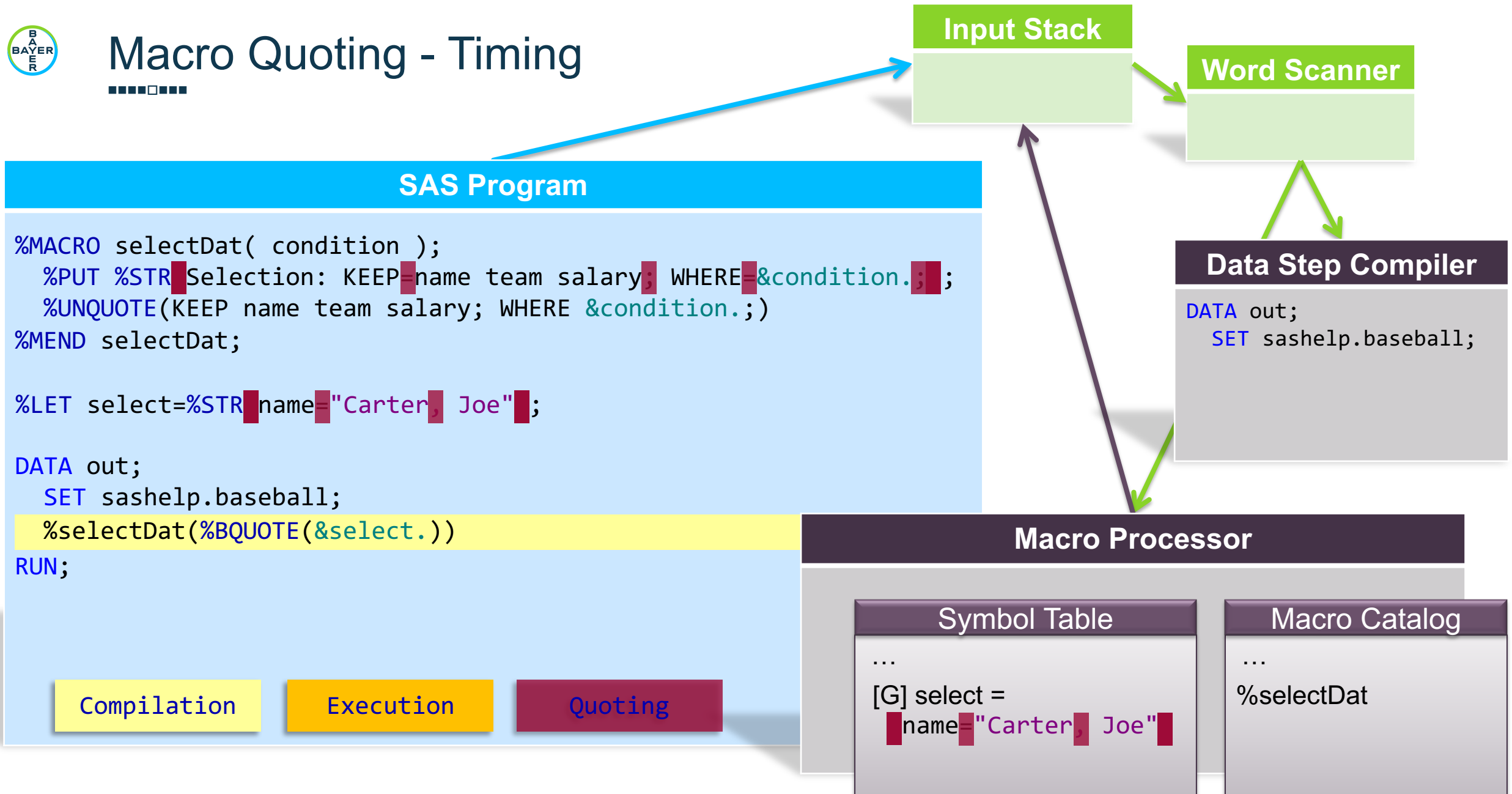
■■■■■





Macro Quoting - Timing

.....





Macro Quoting - Timing

.....

SAS Program

```
%MACRO selectDat( condition );  
  %PUT %STR Selection: KEEP name team salary; WHERE &condition.;  
  %UNQUOTE(KEEP name team salary; WHERE &condition.);  
%MEND selectDat;  
  
%LET select=%STR name="Carter, Joe";  
  
DATA out;  
  SET sashelp.baseball;  
  %selectDat(%BQUOTE(&select.))  
RUN;
```

Compilation

Execution

Quoting

Input Stack

Word Scanner

Data Step Compiler

```
DATA out;  
  SET sashelp.baseball;
```

Macro Processor

Symbol Table

```
...  
[G] select =  
  name="Carter, Joe"
```

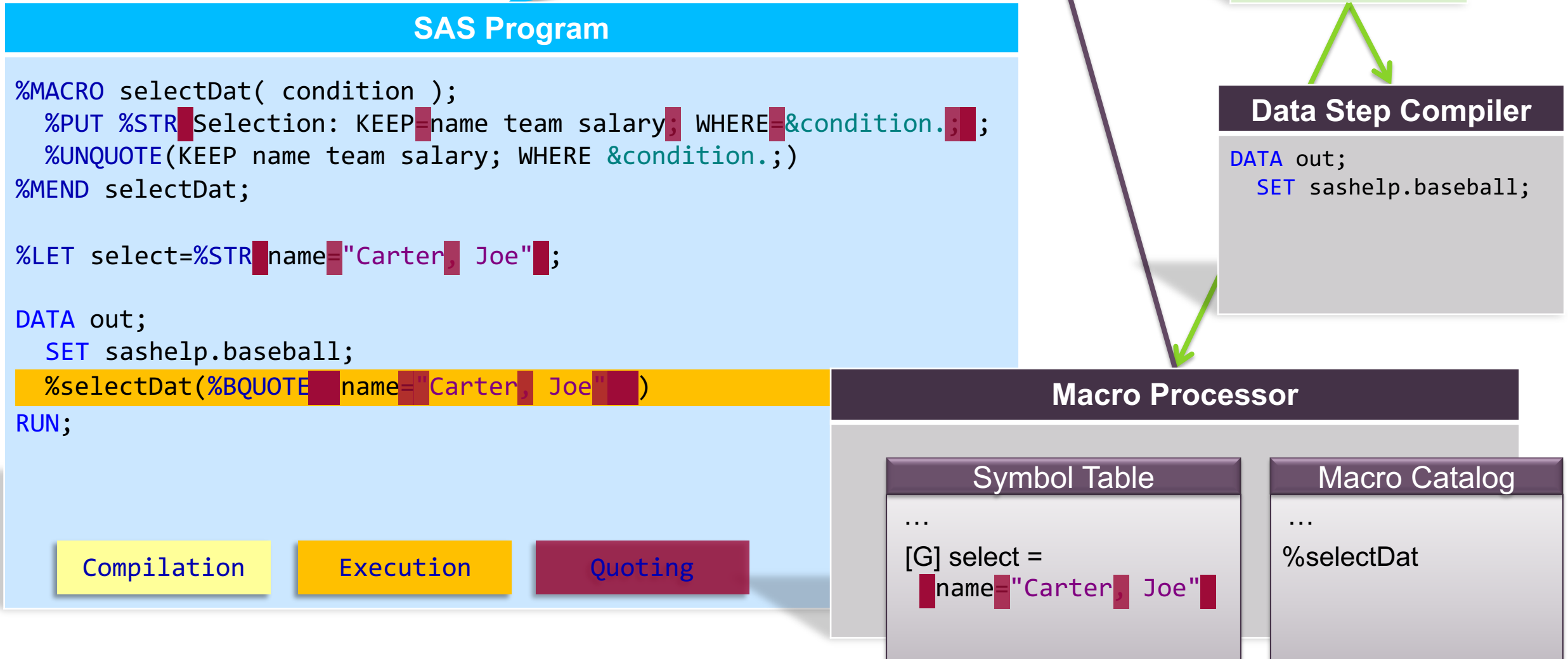
Macro Catalog

```
...  
%selectDat
```



Macro Quoting - Timing

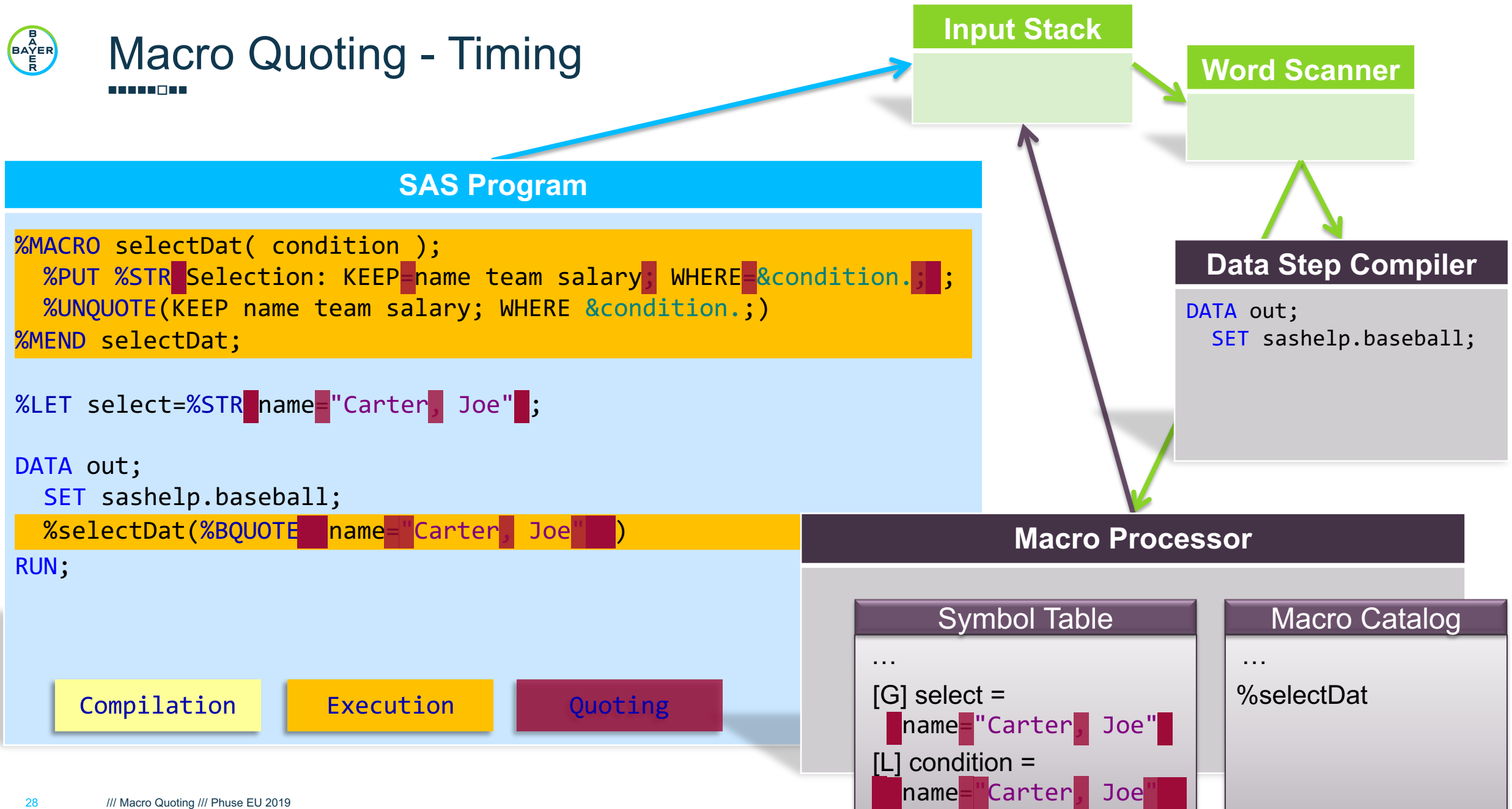
■■■■■□■■





Macro Quoting - Timing

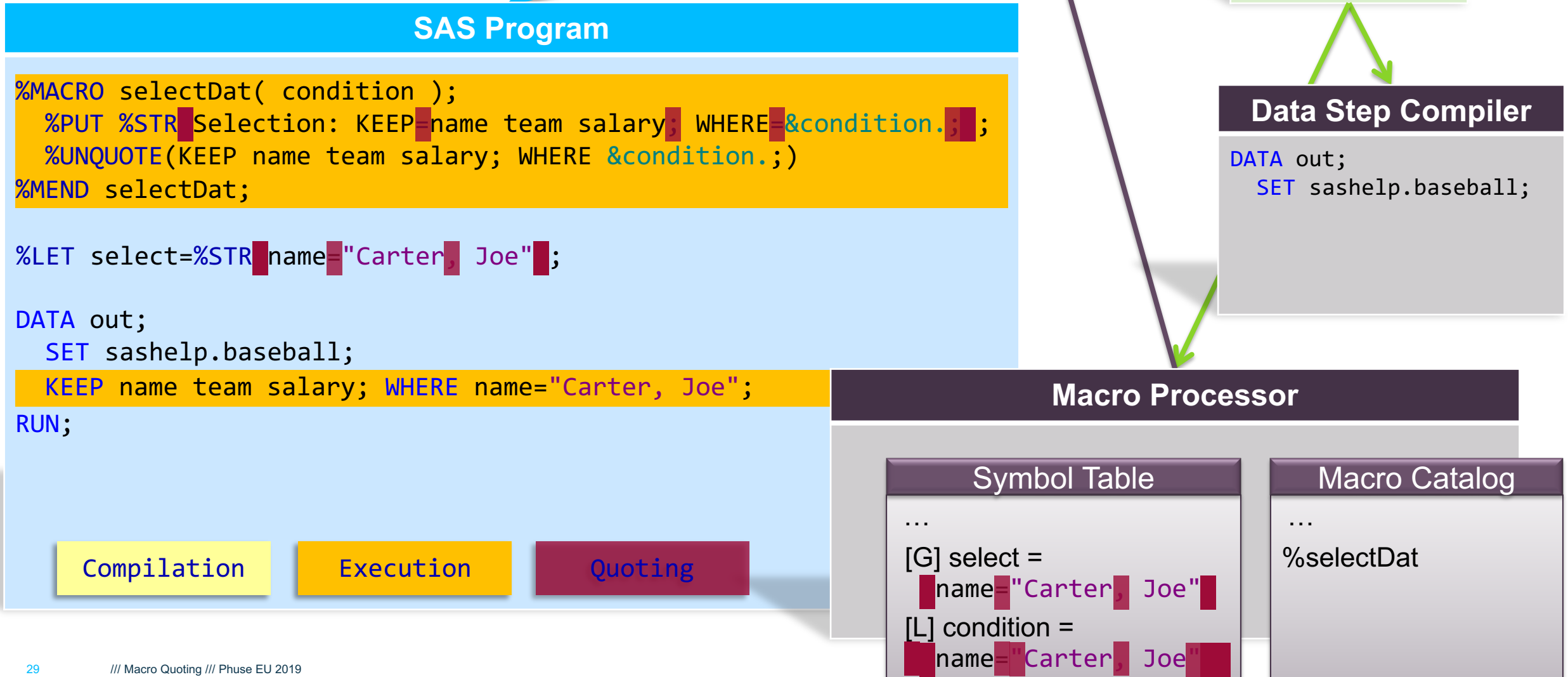
■■■■■□■■





Macro Quoting - Timing

■■■■■■■





Macro Quoting - Timing

■■■■■■■□

SAS Program

```
%MACRO selectDat( condition );  
  %PUT %STR Selection: KEEP name team salary; WHERE &condition.;  
  %UNQUOTE(KEEP name team salary; WHERE &condition.);  
%MEND selectDat;  
  
%LET select=%STR name="Carter, Joe";  
  
DATA out;  
  SET sashelp.baseball;  
  KEEP name team salary; WHERE name="Carter, Joe";  
RUN;
```

Compilation

Execution

Quoting

Input Stack

Word Scanner

Data Step Compiler

```
DATA out;  
  SET sashelp.baseball;  
  KEEP name team salary;  
  WHERE name=  
    "Carter, Joe";  
RUN;
```

Macro Processor

Symbol Table

```
...  
[G] select =  
  name="Carter, Joe"  
[L] condition =  
  name="Carter, Joe"
```

Macro Catalog

```
...  
%selectDat
```

WORK.OUT			
	Name	Team	Salary
1	Carter, Joe	Cleveland	250

Summary



%STR / %NRSTR work **before** resolution of the macro triggers &, %
All other Quoting functions work **after** resolution of &, %

Compilation and Execution phases of the data step have nothing to do with Compilation and Execution Time of the Quoting functions

All masking takes place already in the **compilation phase of the data step**

All quoted characters will be **unquoted automatically** when they leave the Word Scanner

%BQUOTE, %NRBQUOTE and %SUPERQ are used for masking macro **variables/calls/etc.**

%STR and %NRSTR are used for masking **static text**

There is rarely a situation where you need to **use** %BQUOTE instead of %NRBQUOTE

Both are working almost identically and Warnings due to unresolved &.. %.. will also be displayed with **%NRBQUOTE**

%SUPERQ is the actual sister function to %NRSTR

If there is an error in the log, but the log expression can be run without problems – often an **%UNQUOTE** over the whole expression **helps**



Be glad when it's difficult,
the easy things they do all.

-Peter Hohl-

Thanks!



Questions?

Contact: tim.lepp@bayer.com





Backup

```
1      %MACRO selectDat( condition );
2      %PUT %STR(Selection: KEEP=name team salary; WHERE=&condition.);
3      /* %UNQUOTE(KEEP name team salary; WHERE &condition;) */
4      KEEP name team salary; WHERE &condition.;
6      %MEND selectDat;
8      %LET select=%STR(name="Carter, Joe");
11     DATA out;
13     SET sashelp.baseball;
14     %selectDat(%BQUOTE(&select.))
```

Selection: KEEP=name team salary; WHERE=name="Carter, Joe";

NOTE: Line generated by the macro variable "CONDITION".

```
14     name="Carter, Joe"
```

—
22

—
76

ERROR: Syntax error while parsing WHERE clause.

ERROR 22-322: Syntax error, expecting one of the following: a name, a quoted string, a numeric constant, a datetime constant, a missing value, (, *, +, -, :, INPUT, NOT, PUT, ^, ~.

ERROR 76-322: Syntax error, statement will be ignored.

```
15     RUN;
```